İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GÜDÜMLÜ FÜZE MODELİ PARAMETRELERİNİN LİNEER OLMAYAN KESTİRİM YÖNTEMLERİ İLE TAYİNİ

YÜKSEK LİSANS TEZİ

Elçin ASLANOV

Anabilim Dalı : Makina Mühendisliği

Programı : Sistem Dinamiği ve Kontrol

HAZİRAN 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GÜDÜMLÜ FÜZE MODELİ PARAMETRELERİNİN LİNEER OLMAYAN KESTİRİM YÖNTEMLERİ İLE TAYİNİ

YÜKSEK LİSANS TEZİ
Elçin ASLANOV
(503081618)

Tezin Enstitüye Verildiği Tarih : 06 Mayıs 2011
Tezin Savunulduğu Tarih : 07 Haziran 2011

Tez Danışmanı : Prof. Dr. Can ÖZSOY (İTÜ))
Diğer Jüri Üyeleri : Yrd. Doç. Dr. Ayhan KURAL (İTÜ)
Yrd. Doç. Dr. Semih SEZER (YTÜ)

HAZİRAN 2011

Aileme,

ÖNSÖZ

Parametre kestirimi yöntemleri son yıllarda bir sistemin bilinmeyen parametrelerini daha kısa zamanda ve doğru bir şekilde bulunmasında büyük önem taşımaktadır. Özellikle çok parametreli MIMO sistemlerin analizini yapmak için parametre kestirim yöntemlerinin kullanımı günümüzde kaçınılmaz olmuştur. Bu çalışmada güdüm kontrollü bir füze sisteminin bilinmeyen 16 adet parametresi lineer olmayan grey-box ve lineer olmayan arx yöntemleri uygulanarak kestirilmiştir.

Bu çalışmada bilgi ve tecrübeleriyle çalışmama ışık tutan ve değerli yardım ve katkılarıyla beni yönlendiren sayın hocam ve tez danışmanım Prof. Dr. Can ÖZSOY'a teşekkürlerimi sunarım. Sevgili okul arkadaşım Emre Yüzbaşıoğlu'na yardımları için teşekkürlerimi belirtmek isterim.

Son olarak her türlü özveriyi göstererek bugünlere gelmemi sağlayan, benden desteklerini ve sevgilerini hiç eksik etmeyen başta annem, babam, ablam ve abim Opr. Dr. Ayaz ASLAN olmak üzere aileme şükranlarımı sunarım.

Haziran 2011

Elçin ASLANOV

(Makina Mühendisi)

İÇİNDEKİLER

Sayfa

ÖNSÖZ ..... **v**
KISALTMALAR ..... **ix**
ÇİZELGE LİSTESİ ..... **xi**
ŞEKİL LİSTESİ ..... **xiii**
ÖZET ..... **xv**
SUMMARY ..... **xvii**
1. GİRİŞ **1**
1.1 Amaç 1
1.2 Günümüze Kadar Yapılan Çalışmalar 2
1.3 Kapsam 4
2. LİNEER OLMAYAN PARAMETRE KESTİRİMİ **6**
2.1 Lineer Olmayan ARX Model 6
2.1.1 Lineer olmayan ARX modellerin yapısı 7
2.1.2 Lineer olmayan ARX modelleri için lineer olmayan kestirimciler 8
2.1.3 Lineer olmayan kestirimin yapılandırma yolları 11
2.1.4 Lineer olmayan ARX modelin derecesi ve gecikmesi 12
2.1.5 Lineer olmayan ARX modeller için kestirim algoritması 12
2.1.6 Model regresörlerinin yapılandırılması 13
2.2 Grey – Box Model 14
2.2.1 Grey-Box modellerin yapısı 14
2.2.2 Lineer olmayan Grey-Box modellerin kestirimi 15
2.2.3 Lineer olmayan Grey-Box model yapısı 15
2.2.4 IDNLGREY komutunun yapılandırılması 16
2.2.5 IDNLGREY modellerinin kestirimi için pem kullanımı 16
3. OPTİMİZASYON METOTLARI **19**
3.1 Trust-Region Metodu 19
3.1.1 Trust-Region algoritması 21
3.2 Lineer Olmayan En Küçük Kareler Problemlerinin Çözümü 23
3.3 Gauss-Newton Metodu 25
3.4 Levenberg-Marquardt Metodu 28
3.4.1 Levenberg-Marquardt metodunun yakınsaması 30
3.4.2 Levenberg-Marquardt metodunun uygulaması 31
4. FÜZE BODY DENKLEMLERİ **38**
4.1 Füzenin Yere Göre Davranışı 42
5. GÜDÜMLÜ FÜZE SİSTEMİNİN PARAMETRE KESTİRİMİ **47**
5.1 Grey-Box Modeli Kestirimi 48
5.1.1 Trust-Region metodu 56
5.1.2 Gauss-Newton metodu 60
5.1.3 Levenberg-Marquardt metodu 64
5.2 Lineer olmayan ARX modeli kestirimi 71
6. SONUÇLAR VE DEĞERLENDİRME **76**
KAYNAKLAR **80**
EKLER **82**

KISALTMALAR

NLARX **:** Nonlinear ARX
IDNLGREY **:** Identification Nonlinear Grey-Box
TR **:** Trust-Region
GN **:** Gauss-Newton
LM **:** Levenberg-Marquart

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1: Lineer olmayan ARX modelleri için kestirimciler 7
Çizelge 5.1: Trust-Region iterasyon sonuçları..... 60
Çizelge 5.2: Trust-Region kestirim sonuçları. 63
Çizelge 5.3: Gauss-Newton iterasyon sonuçları. 64
Çizelge 5.4: Gauss-Newton kestirim sonuçları..... 67
Çizelge 5.5: Levenberg-Marquardt iterasyon sonuçları..... 68
Çizelge 5.6: Levenberg-Marquardt kestirim sonuçları. 71
Çizelge 5.7: Her bir optimizasyon metodunun karşılaştırması 75

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1: Sistem Tanılama Süreci. 2
Şekil 1.2: Amaçlanan NLARX model 3
Şekil 2.1: Lineer olmayan ARX modelinin yapısı 6
Şekil 3.1: Trust-region eğrisi için parça-parça lineer yaklaşım 15
Şekil 4.1: Füze sisteminin genel şeması 28
Şekil 5.1: Giriş ve çıkışları belirtilmiş güdümlü füze sistemi 36
Şekil 5.2: Sistemin ölçülmüş girişlerinin zamana göre değişimi 38
Şekil 5.3: Sistemin ölçülmüş çıkışlarının zaman göre değişimi 39
Şekil 5.4: İlk parametrelere göre ölçülmüş model ile kestirim karşılaştırması 42
Şekil 5.5: İlk güdümlü füze modeli hatalarının tahmini 43
Şekil 5.6: Ölçülmüş ve Trust –Region simülasyon çıkışlarının karşılaştırılması 46
Şekil 5.7: Trust-Region metodu ile kestirilmiş güdümlü füzenin tahmin hataları 47
Şekil 5.8: Ölçülmüş ve Gauss-Newton simülasyon çıkışlarının karşılaştırılması 50
Şekil 5.9: Gauss-Newton metodu ile kestirilmiş güdümlü füzenin tahmin hataları 51
Şekil 5.10: Ölçülmüş ve Levenberg-Marquardt simülasyon çıkışları 54
Şekil 5.11: Levenberg-Marquardt metodu ile kestirilmiş sistemin tahmin hataları 55
Şekil 5.12: x ekseni etrafındaki açısal hızın kestirimi 56
Şekil 5.13: y ekseni etrafındaki açısal hızın kestirimi 56
Şekil 5.14: z ekseni etrafındaki açısal hızın kestirimi 57
Şekil 5.15: y ekseni yönündeki ivme 57
Şekil 5.16: z ekseni yönündeki ivme 58
Şekil 5.17: Wavenet fonksiyonu ile lineer olmayan arx parametre kestirimi 60
Şekil 5.18: Sigmoident fonksiyonu ile lineer olmayan arx parametre kestirimi 62
Şekil 6.1: x-ekseni etrafındaki açısal hız 63
Şekil 6.2: y-ekseni etrafındaki açısal hız 64
Şekil 6.3: z-ekseni etrafındaki açısal hız 64
Şekil 6.4: y-ekseni yönündeki ivme 65
Şekil 6.5: z-ekseni yönündeki ivme 65

GÜDÜMLÜ FÜZE MODELİ PARAMETRELERİNİN LİNEER OLMAYAN KESTİRİM YÖNTEMLERİ İLE TAYİNİ

ÖZET

Son yıllarda bilim adamları ve mühendisler lineer ve lineer olmayan sistemlerin analizinde modelin bilinmeyen parametrelerini elde etmek için parametre kestirimi metotlarını yaygın olarak kullanmaktadır. Parametre kestirimi bir sistem modelinin bilinmeyen parametrelerini kestirmek için uygulanan en yaygın yöntemdir. Özellikle deneysel olarak bazı parametrelerin ölçülemediği zamanlarda parametre kestirimi metotlarını uygulayarak sistem analizi gerçekleştirilir.

Lineer olmayan sistemlerin analizinde en yaygın kullanılan parametre kestirimi metotları lineer olmayan Grey-Box ve lineer olmayan ARX modelleridir. Gery-box model için lineer olmayan ayrık zamanlı ve sürekli zamanlı sistemlere zaman serisi verisi sağlayarak modelleyebiliriz. Grey-box modeller statik veya dinamik olabilir ve sistem davranışını bilinmeyen parametrelerle lineer olmayan diferansiyel denklemler olarak tanımlar. Lineer olmayan ARX modelde lineer olmayan yapılar, lineer olmayan ve lineer blokların paralel olarak birleştirilmesi ile kullanılırlar. Lineer ve lineer olmayan fonksiyonlar regressörler olarak bilinen, değişkenlerin terimleri olarak tanımlanmaktadır.

Bu çalışmada güdümlü füze sisteminin bilinmeyen parametrelerinin lineer olmayan grey-box ve lineer olmayan arx model kullanılarak parametre kestirimi yapılacaktır. Her bir model için ayrı ayrı Gauss-Newton, Levenberg-Marquardt ve Trust-Region optimizasyon metotları kullanılmıştır.

DETERMINATION OF GUIDED MISSILE PARAMETERS USING NONLINEAR PARAMETER ESTIMATION METHODS

SUMMARY

Nowadays scientists and engineers are using parameter estimation methods widely to obtain unknown parameters in linear and nonlinear system analyzing. Parameter estimation is the most common method to estimate system unknown parameters. Especially when we do not obtain some parameters experimentally we use parameter estimation methods for analyzing.

Nonlinear Grey-Box and Nonlinear ARX models are most widely methods using to analyzing nonlinear system. Grey-box models ODEs specify the mathematical structure of the model explicitly, including couplings between parameters and known parameter values. Grey-box modeling is useful when you know the relationships between variables, constraints on model behavior, or explicit equations representing system dynamics. We can estimate nonlinear discrete-time and continuous-time grey-box models for arbitrary nonlinear ordinary differential equations using single-output and multiple-output time-domain data, or output-only time-series data. Grey-box models can be static or dynamic. Grey-box models describe the system behavior as a set of nonlinear ordinary differential or difference equations (ODEs) with unknown parameters.We can estimate discrete-time nonlinear ARX models from time-domain input-output data or time-series data and single-output or multiple-output data. Nonlinear ARX models describe nonlinear structures using a parallel combination of nonlinear and linear blocks. The nonlinear and linear functions are expressed in terms of variables called regressors.

During this thesis, guided missile system unknown parameters determined using nonlinear Grey-Box and nonlinear ARX models. Gauss-Newton, Levenberg-Marquardt and Trust-Region optimization methods applied for each parameter estimation methods.

1. GİRİŞ

1.1 Amaç

21.yy' da lineer olmayan sistemler üzerinde yapılan araştırmalar ve uygulamalar sonucu lineer olmayan sistemlerin analizi çok gelişmiştir. Lineer olmayan sistemlerin analizi amaca göre değişmektedir. Herzaman ilk olarak Sistem Tanımlama ile analize başlanmaktadır. Sistem tanımlama iteratif bir süreçtir, ilk olarak verilerden farklı yapılardaki modeller belirlenmektedir ve model performansı karşılaştırılmaktadır. İlk olarak basit model yapılarından parametrelerin kestirimine başlanmaktadır. Eğer model performansı iyi değil ise model yapısının karmaşıklığı adım-adım arttırılmaktadır. En sonunda en basit modeli seçilir ki bu model sistemin dinamiğini tanımlayan en iyi model olacaktır. Model yapısını belirlemede diğer önemli bir husus yüksek dereceden modellerin daima doğru olmamasıdır. Modelin karmaşıklığını arttırmak demek parametre kestiriminde belirsizliğin artması demektir ve genellikle daha fazla veri gerektirmektedir. Modelin yapıları, model yapılarının doğruluğunu belirlemede tek faktör değildir. Eğer model iyi seçilmemişse verileri gözden geçirmek gerekebilir, gürültünün filtrelenmesini örnek olarak verebilir.

Model yapısı belirlendikten sonra parametre kestiriminde uygun kestirim metodu seçilmelidir. Son yıllarda parametre kestirimi metotlarının algoritmaları çok geliştirilmiştir. Bazı özel durumlar dışında genel olarak Grey-Box modeli ve Black-Box modeli kullanılan en yaygın modellerdendir. Bu kestirim modelleri günümüzde sanayi araştırma merkezlerinde çok yaygın bir şekilde kullanılmaktadır. Belirlenen modelleri zamana bağlı verilerle analiz edilmektedir. Sürekli zaman sistemi direk olarak kestirebilmek için modelin giriş-çıkış polinom modeli, düşük dereceli transfer fonksiyonu modeli veya durum uzayı modeli olması gerekir. Sistem tanımlama verileri ile sürekli zaman transfer fonksiyonların dışında bütün lineer ve lineer olmayan modeller ayrık zamanlı sistem olarak kestirilebilmektedir. Grey-box modellerde, sürekli ve ayrık zamanlı sistemlerin her ikisini lineer ve lineer olmayan diferansiyel denklemler için zamana bağlı verilerden kestirilebilmektedir. Lineer olmayan modellerde, ayrık zamanlı Hammerstein-Wiener ve lineer olmayan ARX

modelleri zamana bağlı verilerden kestirebilir. Ayrıca lineer olmayan grey-box modelleri zamana bağlı verilerden kestirilebilmektedir.

Model yapısını kestirimden bağımsız bir şekilde kurduğumuzda model yapıları bir model nesnesini, bütün gerekli model parametrelerini açık bir şekilde tanımlamak için komut satırlarında kullanılır. Bir model simülasyonu, model analizi veya kestirmeden önce model parametre değerleri belirlemek için ilk tahmin belirleme gibi durumlarda model nesnesi kestirilmeden bağımsız olarak modellenebilmektedir.

Birçok durumda yapıyı ve modeli kestirmek için model yapısından bağımsız olarak kestirim komutu kullanılmaktadır. Örnek olarak, pem kestirim komutu bilinmeyen parametreli model yapısını ve kestirim algoritmasını belirtmek için kullanılmaktadır.

Grey-box modellerinde ilk olarak model belirlenmelidir ve sonra diferansiyel denklemlerden parametreler kestirilmelidir.

Yeni bir model kestirildiğinde model özellikleri kestirimci yapı içerisinde direk olarak belirtilebilmektedir. Komut kullandığında model hem kurulabilmektedir hem de kestirilebilmektedir ve bütün üst seviye model özelliklerini kestirimci komut içerinde belirtebiliriz. Ayrıca algoritmanın yapısı özellik değerlerini kullanarak kestirimci içinde belirtilebilmektedir.

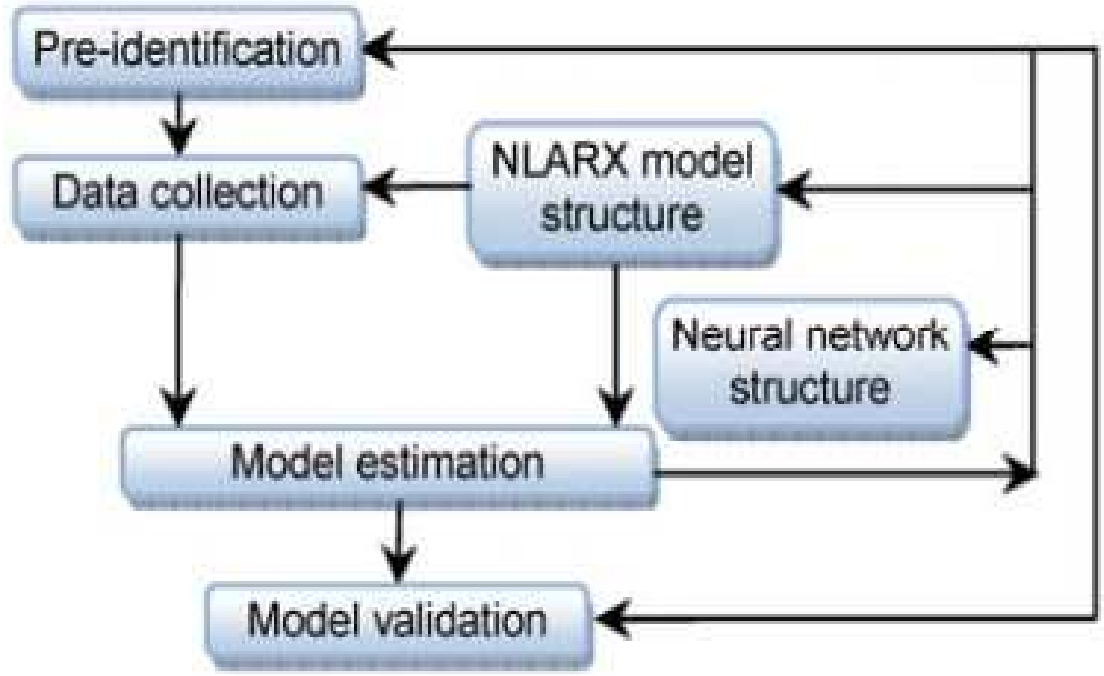
Bu çalışmada lineer olmayan grey-box ve lineer olmayan ARX parametre kestirim modellerini kullanarak 10 giriş ve 5 çıkışlı güdümlü füze sisteminin 16 adet parametresini başarılı bir şekilde kestirmek için uygulama yapılmıştır. Çalışmada MATLAB/Simulink paket programı yardımı ile kestirim için ilk olarak lineer olmayan grey-box modeli uygulanmış ikinci olarak lineer olmayan ARX modeli uygulanmıştır. Her bir model için Gauss-Newton, Levenberg-Marquardt ve Trust-Region optimizasyon metotları uygulanmış ve performanslarının gözlemlenmesi amaçlanmıştır. Son olarak yapılan analizlerin sonucuna göre yorumlar yapılmıştır.

1.2 Günümüze kadar yapılan çalışmalar

Lineer olmayan parametre kestirimi metotları lineer olmayan sistemlerin ölçülemeyen parametrelerini belirlemek için en elverişli yöntemler olduklarından günümüzde gittikçe artan bir sıklıkla uygulanmaktadır. Aşağıda lineer olmayan kestirim metotlarının uygulandığı bazı çalışmalar anlatılmaktadır.

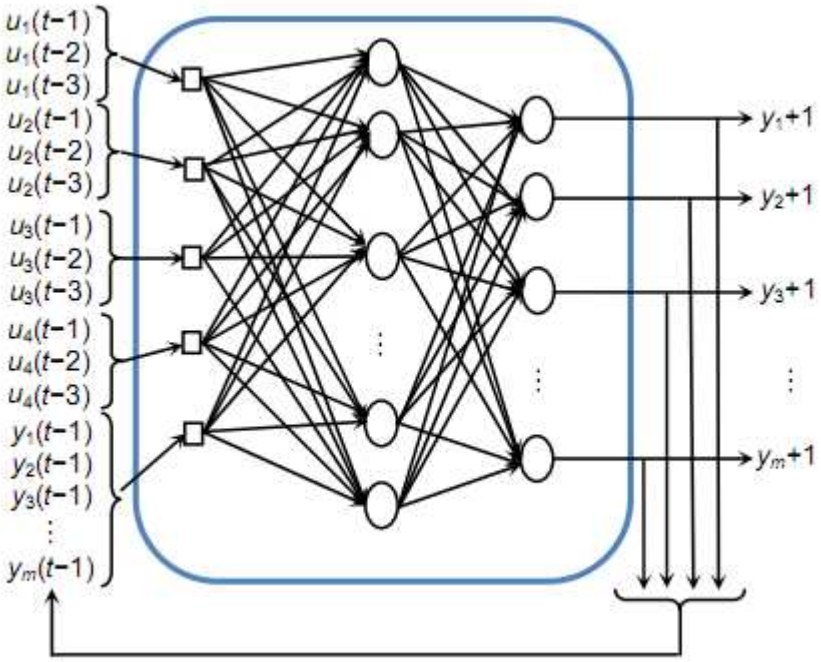
Gazi Üniversitesinde 2004 yılında yapılan bir çalışma ile lineer olmayan regresyonda parametre kestirimi için genetik algoritma yöntemi uygulanmıştır. Bu çalışmada doğrusal olmayan regresyonda parametre tahmini için alternatif bir yöntem önerilmiştir. Bu yöntem son yıllarda oldukça geniş bir kullanım alanına sahip olan genetik algoritmalar yöntemidir. Genetik algoritmanın kullanımı diğer parametre tahmin yöntemleri gibi yardımcı bilgiler gerektirmez ve bu nedenle pratik uygulamalar için kullanışlıdır. Bu çalışmada genetik algoritmaların diğer yöntemler kadar etkin çözümlere ulaşabileceği S-biçimli büyüme modellerinde parametre tahminleri yapılarak gösterilmiştir. Sonuçlar genetik algoritmaların yardımcı bilgiler gerektirmeksizin doğrusal olmayan regresyonda parametre tahmini için kullanılabilir olduğunu göstermektedir.

Malaya Üniversitesinde 2010 yılında yapılan çalışma ile küçük ölçekli helikopterin kontrolu ve tanılaması yapılmıştır. Bu çalışmada matematiksel yapıyı tanımlamak ve parametre kestirimi yapılması için lineer olmayan ARX model yapısı kullanılmıştır. Bu kestirim sürecinde gradyant arama metodu kullanılmıştır. Aşağıda sistemin genel simülasyon diyagramı verilmiştir.



Şekil 1.1: Sistem tanılama süreci

Genel olarak uygulanacak NLARX modelin yapısı aşağıdaki gibi olmaktadır.



Şekil 1.2: Amaçlanan NLARX model

Linköping Üniversitesinde 2005 yılında endüstriyel robotların sistem tanılaması için lineer olmayan grey-box model uygulaması yapılmıştır. İlk iki adımda robotun dinamik denklemleri, sürtünme ve esnekliği tanımlanmıştır. Bu ilk iki adımda zamana bağlı ölçülmüş veriler kullanılarak ilk parametreler kestirilmiştir. Çalışmada kullanılan veriler gerçek endüstriyel robottan elde edilmiş verilerdir. Robotun modeli çıkartıldıktan sonra lineer olmayan grey-box model yapısına göre kestirim yapılmıştır.

Lineer olmayan parametre kestirimi metotlarını kullanarak daha birçok lineer olmayan sistemin analizi yapılmıştır ve bu metotların geliştirilmesi için çalışmalar yapılmaktadır.

1.3 Kapsam

Bu tez çalışmasında yukarıda bahsedilen çalışmalara paralel olarak lineer olmayan güdümlü füze sistemi için lineer olmayan parametre kestirimi metotlarını kullanarak belirlenemeyen parametrelerin kestirimi uygulaması yapılmıştır. Birinci bölümde lineer olmayan parametre kestirimi metotlarına bir giriş yapılmıştır. Ayrıca

çalışmanın amacı ve günümüze kadar yapılan endüstriyel ve akademik çalışmalardan örnekler verilmiştir. İkinci bölümde lineer

olmayan parametre kestirimi metotları üzerinde durularak çalışmada kullanılacak grey-box ve lineer olmayan ARX modellerinin yapısı anlatılmıştır. Üçüncü bölümde sınırlandırılmamış optimizasyon metotları anlatılmıştır. Çalışmada kullanılacak olan Gauss-Newton, Levenberg-Marquardt ve Trust-Region optimizasyon metotlarından ayrıntılı olarak bahsedilmiştir. Dördüncü bölümde füzenin dinamik denklemlerinin elde edilmesi ve durum uzay modelinin çıkarılışı anlatılmıştır. Beşinci bölümde tezde kullanılan lineer olmayan güdümlü füze sistemi tanıtılarak yukarıda bahsedilen lineer olmayan parametre kestirimi metotları kullanımı ile parametre kestirimi anlatılmıştır. Çalışma lineer olmayan parametre kestirimi metotlarının performanslarının karşılaştırılması ile son bulmaktadır.

2. Lineer Olmayan Parametre Kestirimi

2.1 Lineer Olmayan ARX Model

Lineer olmayan ARX model lineer olmayan durumlar için lineer ARX modelin genişletilmiş halidir ve aşağıdaki yapıya sahiptir:

$$y(t) = f\left(y(t-1),...,y(t-na),u(t-nk),...,u(t-nk-nb+1)\right) \quad \textbf{(2.1)}$$

burada f fonksiyonu sonlu sayıda u girişlerine ve y çıkışlarına bağlıdır. na, y çıkışlarını tahmin etmek için bir önceki çıkış terimlerinin sayısını ifade etmektedir. nb ise u girişlerini tahmin etmek için bir önceki giriş terimlerini, nk girişden çıkışa olan gecikmedir ve örneklerin sayısını tanımlar.

Genellikle lineer olmayan ARX modelleri black-box yapıları olarak kullanılır. Lineer olmayan ARX modelin lineer olmayan fonksiyonu esnek lineer olmayan kestiricidir. Uygulamalarda lineer olmayan ARX modeller idnlarx nesnesi kullanılarak ifade edilmektedir.

Bir lineer SISO ARX modelin yapısı aşağıdaki gibidir:

$$\begin{aligned} &y(t) + a_1 y(t-1) + a_2 y(t-2) + ... + a_{na} y(t-na) = \\ &= b_1 u(t) + b_2 u(t-1) + ... + b_{nb} u(t-nb-1) + e(t) \end{aligned} \quad \textbf{(2.2)}$$

burada giriş gecikmesi nk sıfır alınmıştır.

Bu yapıda belirtilen $y(t)$ çıkışı bir önceki çıkış ve giriş değerlerini kullanarak kestirim yapmaktadır. Denklemi aşağıdaki gibi tekrar yazabiliriz:

$$\begin{aligned} &y_p(t) = \left[-a_1, -a_2, ..., -a_{na}, b_1, b_2, ..., b_{nb}\right] * \\ &*\left[y(t-1), y(t-2), ..., y(t-na), u(t), u(t-1), ..., u(t-nb-1)\right]^T \end{aligned} \quad \textbf{(2.3)}$$

burada $y(t-1), y(t-2), ..., y(t-na), u(t), u(t-1), ..., u(t-nb-1)$ ifadesi giriş ve çıkışın gecikmesini ifade etmektedir. Lineer ARX model y_p çıkışını regresörlerin toplamı olarak tahmin etmektedir.

Lineer olmayan ARX model esnek lineer olmayan birçok fonksiyona içermektedir:

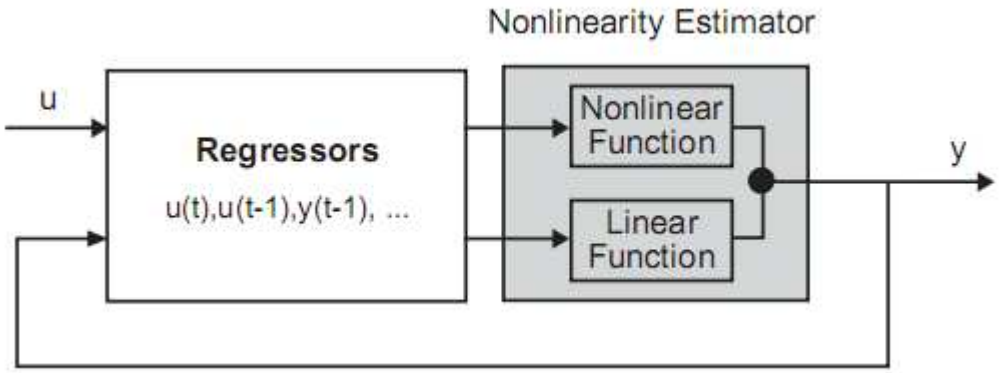
$$y_p(t) = f\left(y(t-1), y(t-2), y(t-3), ..., u(t), u(t-1), u(t-2), ...\right) \quad \textbf{(2.4)}$$

Burada f fonksiyonu lineer olmayan fonksiyondur. f fonksiyonu için girişler model regresörleridir. Lineer olmayan ARX model yapısını belirtirken ileride bahsedilecek olan birkaç lineer olmayan fonksiyondan bir tanesi seçilmektedir.

Lineer olmayan ARX regresörleri giriş çıkış değişkenlerinin gecikmesi, giriş çıkış değişkenlerinin gecikmesinin lineer olmayan ifadeleri olabilir. Lineer olmayan regresörlere örnek olarak $y(t-1)^2, u(t-1)*y(t-2), \tan\left(u(t-1)\right)$, ve $u(t-1)*y(t-3)$ verilebilir.

2.1.1 Lineer olmayan ARX modellerin yapısı

Aşağıdaki blok diyagramı lineer olmayan ARX modelin yapısını tanımlamaktadır:



Şekil 2.1: Lineer olmayan ARX modelinin yapısı

Bu lineer olmayan ARX model y çıkışını iki aşamada hesaplamaktadır. İlk olarak regresörleri hesaplanan ve bir önceki giriş çıkış verilerinden hesaplamaktadır. Bu basit durumda regröseler girişlerin ve çıkışların gecikmeleridir, örnek olarak $u(t-1)$ veya $u(t-1)*y(t-3)$ gösterilebilir, bunlar standart regresörler olarak tanımlanır. Ayrıca özel regresörlerde belirtilebilir ve bunlar giriş çıkış gecikmelerinin lineer olmayan fonksiyonlarıdır. Örnek olarak, $\tan\left(u(t-1)\right)$ veya $u(t-1)*y(t-3)$.Varsayılan bütün regröserler lineer olmayan kestirimcilerin lineer ve lineer olmayan blokları için girişlerdir. Lineer olmayan fonksiyon bloğunun girişleri olarak bir regrösör altkümesi seçilebilmektedir.

İkinci olarak lineer olmayan kestirimci bloğu regresörleri lineer ve lineer olmayan fonksiyonların birleşenlerini kullanarak model çıkışını eşleştirir.Tree- partition ağları, wavelet ağları ve çok girişli ağlar olarak bilinen lineer olmayan kestirimcilerden birisini seçebiliriz. Ayrıca lineer olmayan kestirimciden lineer veya lineer olmayan fonksiyonlardan bir tanesini hariç tutabiliriz.

Lineer olmayan kestirimci blok lineer ve lineer olmayan blokları paralel bir şekilde içerebilir.

Örnek olarak:

$$F(x) = L^T(x-r) + d + g(Q(x-r)) \quad \textbf{(2.5)}$$

Burada x regresörlerin vektörüdür, $L^T(x)+d$ lineer fonksiyon bloğunun çıkışıdır ve $d \neq 0$ olduğunda geçerlidir. d bir skaler uzaklıktır. $g(Q(x-r))$ lineer olmayan fonksiyon bloğunun çıkışını ifade emektedir. r, x regresörlerinin ortalamasıdır. Q bir öngörü matrisidir. $F(x)$ fonksiyonunun tam hali lineer olmayan kestirimcinin seçimine bağlıdır.

Lineer olmayan ARX modelin kestirimi model parametre değerlerini hesaplar, örnek olarak L,r,d,Q ve g 'yi belirten diğer parametreleri gösterebiliriz. Sonuçta oluşan modeller idnlarx nesnelerinin içerdiği bütün model verileri, model regresörleri ve lineer olmayan kestirimcinin parametreleridir.

2.1.2 Lineer olmayan ARX modelleri için lineer olmayan kestirimciler

Bu çalışmada lineer olmayan ARX modelleri için birkaç lineer olmayan F(x) kestirimcileri anlatılacaktır.

Lineer olmayan kestirimcilerin çoğu lineer olmayan birimlerin toplamı olarak lineer olmayan fonksiyonları ifade etmektedir. Buna örnek olarak wavelet ağları veya sigmoid fonksiyonlarını gösterebiliriz. Kestirim için n adet lineer olmayan birim tanımlayabiliriz. Her bir kestirimcinin detaylı bir şekilde tanımlanması için aşağıdaki ifadeleri inceleyebiliriz.

Çizelge 2.1: Lineer olmayan ARX modelleri için kestirimciler

Lineer Olmama Durumu	Sınıf	Yapı	Açıklama
Wavelet Network	wavenet	$g(x)=\sum_{k=1}^{n}\alpha_k k(\beta_k(x-\gamma_k))$ Burada k(s) wavelet fonksiyonudur	Kestirim algoritması n adet birimi otomatik olarak hesaplar. Birim sayısı n=10 dur.
Sigmoid Network	sigmoidnet	$g(x)=\sum_{k=1}^{n}\alpha_k k(\beta_k(x-\gamma_k))$ Burada k(s) sigmoid fonksiyonudur.	
Tree Partition	treepartition	Parça parça lineer fonksiyon iki kısımdan meydana gelen regresör alanı tarafından tanımlanmaktadır.	Birimleri kestirim algoritması otomatik olarak hesaplar.
Lineer F Fonksiyonu	linear	Bu kestirimci lineer ARX model'e benzer olarak bir model üretmektedir ama özel regresörleri belirtmek için ek esneklik gerekmektedir.	Lineer olmayan kestirimci olarak özel regresörleri belirtmek için kullanılmaktadır.

Wavenet fonksiyonu tanımlanmış lineer olmayan kestirimci wavelet ağlarını, lineer olmayan ARX modeli için sınıflandırmaktadır. Wavenet lineer olmayan ARX modelini kestirmek için lineer olmayan kestirimci wavelet ağlarını bir arada tutmaktadır.

Aşağıda wavenet fonksiyonunun lineer olmayan $y=F(x)$ fonksiyonunu tanımlamak için kullanılması gösterilmektedir. Burada y skalerdir ve x parametresi m boyutunda satır vektörüdür. Wavenet fonksiyonu aşağıdaki fonksiyonun açılımına bağlıdır:

$$\begin{aligned} F(x) = (x-r)PL &+ a_{s1} f\left(\left(b_{s1}(x-r)\right)Q - c_{s1}\right) + \ldots \\ &+ a_{sn} f\left(\left(b_{sn} s(x-r)\right)Q - c_{sn}\right) \\ &+ a_{w1} g\left(\left(b_{w1}(x-r)\right)Q - c_{w1}\right) + \ldots \\ &+ a_{wn} w g\left(\left(b_{wn} w(x-r)\right)Q - c_{wn}\right) + d \end{aligned} \qquad \textbf{(2.6)}$$

Burada f ölçek fonksiyonu ve g wavelet fonksiyonudur. P ve Q sırasıyla $m \times p$ ve $m \times q$ tahmin matrisleridir ve bu matrisler kestirim verilerinin analizinden elde

edilmektedir. Genellikle $p = m$ dir. Eğer kestirim verilerindeki x in bileşenleri lineer olarak bir birine bağlıysa $p < m$ olacaktır. Q matrisinin sütun sayısı olan q ölçek ve wavenet fonksiyonlarında kullanılan x bileşenlerinin sayısına göre uygun bir şekilde ayarlanmaktadır. r kestirim verilerinden hesaplanan regresör vektörünün ortalama değerini göstermektedir ve $1 \times m$ vektörü olarak tanımlanmaktadır. Fonksiyondaki d, a_s, b_s, a_w ve b_w parametreleri skalerdir. s tabanlı parametreler ölçek fonksiyonunun parametrelerdir ve w tabanlı parametreler ise wavenet fonksiyonunun parametreleridir. L parametresi $p \times 1$ vektörüdür. c_s ve c_w parametreleri $1 \times q$ vektörüdür. Aşağıda verilen f ölçek fonksiyonu ve g wavelet fonksiyonları her ikisi dairesel fonksiyonlardır:

$$\begin{aligned} f(x) &= e^{-0.5x'x} \\ g(x) &= \left(\dim(x) - x'x\right) e^{-0.5x'x} \end{aligned} \qquad \textbf{(2.7)}$$

Sigmoidnet fonksiyonu tanımlanmış lineer olmayan kestirimci sigmoid ağlarını lineer olmayan ARX modeli için sınıflandırmaktadır. Sigmoidnet lineer olmayan ARX modelini kestirmek için lineer olmayan kestirimci sigmoid ağlarını bir arada tutmaktadır.

Aşağıda sigmoidnet fonksiyonunun lineer olmayan $y = F(x)$ fonksiyonunu tanımlamak için kullanılması gösterilmektedir. Burada y skalerdir ve x parametresi m boyutunda satır vektörüdür. Sigmoidnet fonksiyonu aşağıdaki fonksiyonun açılımına bağlıdır:

$$\begin{aligned} F(x) = (x-r)PL + a_1 f\left((x-r)Qb_1 - c_1\right) + \ldots \\ + a_n f\left((x-r)Qb_n - c_n\right) + d \end{aligned} \qquad \textbf{(2.8)}$$

Burada f sigmoid fonksiyonudur ve aşağıdaki gibi elde edilmiştir:

$$f(z) = \frac{1}{e^{-z} + 1} \qquad \textbf{(2.9)}$$

Yukarıdaki fonksiyonda P ve Q sırasıyla $m \times p$ ve $m \times q$ tahmin matrisleridir ve bu matrisler kestirim verilerinin analizinden elde edilmektedir. Genellikle $p = m$ dir. Eğer kestirim verilerindeki x in bileşenleri lineer olarak bir birine bağlıysa $p < m$ olacaktır. Q matrisinin sütun sayısı olan q ölçek ve sigmoidnet fonksiyonlarında

kullanılan x bileşenlerinin sayısına göre uygun bir şekilde ayarlanmaktadır. r kestirim verilerinden hesaplanan regresör vektörünün ortalama değerini göstermektedir ve $1\times m$ vektörü olarak tanımlanmaktadır. Fonksiyondaki d, a_k ve c_k parametreleri skalerdir. L parametresi $p\times 1$ ve b_k parametresi $q\times 1$ vektörüdür.

2.1.3 Lineer olmayan ARX kestirimin yapılandırma yolları

Lineer olmayan ARX model yapısına ve verilere farklı modeller tasarlamak için çeşitli elementler ayarlanabilir.

Model yapıları yapılandırıldığında ilk olarak model derecesi ve gecikmesi belirtilir, bu şekilde standart regresörler yaratılmış olur. İkinci olarak ise özel regresörler yaratılır. Özel regresörler en son giriş çıkışların rastgele fonksiyonlarıdır örnek olarak ürünler, güçler ve giriş ve çıkışların diğer Matlab ifadeleridir.

Veri modellerken standart regresörlerin yerine daha esnek özel regresörler belirtmek mümkündür. Lineer olmayan kestirimci bloğun lineer olmayan fonksiyonunda regresörlerin bir alt kümesi vardır. Bir lineer ARX modeli kullanmaya başlarken modelin tüm yapısı bitirilmiş olmalıdır.

Bu operasyon ancak komut yazarak uygulanabilmektedir. Komut ile başlangıç standart regresörleri kullanarak lineer olmayan modeli yapılandırılır ve lineer modelin derecesini ve gecikmesini kullanarak hesaplama yapılır.

Lineer olmayan kestirimci blok aşağıdaki şekilde yapılandırılırken ilk olarak birim sayılarını içeren lineer olmayan fonksiyonları belirtmek ve yapılandırmak gerekmektedir. Daha sonra $F(x)=L^T(x)+d$ gibi lineer olmayan kestirimciden lineer olmayan fonksiyonu çıkartmak ve $F(x)=g(Q(x-r))$ gibi lineer olmayan kestirimciden lineer fonksiyonu çıkartmak gerekmektedir.

Varsayılan yapılandırma ile bir lineer olmayan ARX model kestiriminde ilk olarak sadece model derecesi ve giriş gecikmesi belirtilir. Derecenin belirtilmesi standart regresörleri otomatik olarak yaratır. İkinci olarak lineer ARX model belirtilir. Lineer model, model derecelerini ve lineer olmayan modelin lineer fonksiyonunu belirler. Bu işlem komut yazılarak uygulanmaktadır. Üçüncü olarak lineer olmayan kestirimci wavelet network olduğunda bu model yeterli sonuçlar sağlar ve hızlı kestirim metodu

kullanır. Bütün standart regresörler wavelet ağlarının lineer olmayan fonksiyonlarının girişleridir.

2.1.4 Lineer olmayan ARX modelin derecesi ve gecikmesi

Lineer olmayan ARX modellerin derecesi ve gecikmesi pozitif tamsayıdır. Burada na – çıkışı kestirmek için bir önceki çıkış terimlerin sayısıdır, nb – çıkışı kestirmek için bir önceki giriş terimlerin sayısıdır, nk – örnekleme terimlerindeki girişten çıkışa gecikmeyi belirtmektedir.

na, nb, nk terimlerinin anlamı lineer ARX model parametreleri içinde benzerdir. Dereceler SISO verisi için skalerdir ve MIMO verileri için matrislerden oluşmaktadır. Derece ve gecikmenin nasıl belirtileceğine karar veremediğimizde lineer ARX modellere dayanarak kestirim yapılır ve ilk öngörü sağlanmış olur, bu öngörü ile lineer olmayan ARX model için derece ve gecikme belirtilebilir.

Örnek olarak, eğer SISO model için giriş u ve çıkış y'nin derece ve gecikmesi

na = 2, nb = 3, ve nk = 5 olarak elde edilirse, $y(t-2), y(t-1), u(t-5), u(t-6),$ ve $u(t-7)$ standart regresörleri hesaplanır.

2.1.5 Lineer olmayan ARX modeller için kestirim algoritması

Kestirim algoritması lineer olmayan kestirimcinin ve nlarx sınıfının diğer özelliklerinin seçimine bağlıdır. Ayrıca algoritma özellikleri GUI ve komut yolu ile ayarlanabilir. Varsayılan lineer olmayan ARX modellerin kestirimi kestirim hatalarını minimize eder, bunun anlamı hedef değerin kestirim değerine karşılık gelmesidir. Eğer simülasyon davranışını tekrarlayan optimize bir model isteniyorsa simülasyon için hedef değerleri ayarlamak gerekmektedir. Bu durumda treepartition ve neuralnet kullanılmamaktadır çünkü bu lineer olmayan kestirimciler farklı değildir. Simülasyon hatalarının minimize etmek için farklı lineer olmayan fonksiyonlar gerekir. Simülasyon hatasını minimize etmek bir adım önceki kestirim hatasını minimize etmekten daha çok zaman alır.

nlarx sınıfın genel algoritma özelliklerini açıklarsak MaxIter – maksimum iterasyon sayısını, SearchMethod – Kestirim veya simülasyon hatalarını minimize etmek için arama metodunu göstermektedir, örnek olarak Gauss-Newton ve Levenberg Marquardt çizgi arama metodu ve Trust-Region reflektif Newton yaklaşımı

verilebilir. Tolerance – Belirtilen bir değerden az olarak parametre değerlerinde beklenen gelişme elde edildiğinde iterasyona son verme durumunu belirtir. Display - Matlab komut penceresinde minimizasyon iterasyonunun ilerlemesini gösterir.

nlarx komutu kullanarak lineer olmayan bir ARX modeli hem kurulabilir hem de kestirilebilir. Her bir kestirimden sonra diğer modellerle karşılaştırarak modelin doğruluğuna bakılmalıdır ve model cevabı kestirilmelidir.

En basit kestirime $m = nlarx\left(data,\left[na\ nb\ nk\right]\right)$ kullanarak başlanır.

Örnek olarak: $m = nlarx\left(data,\left[2\ 2\ 1\right]\right) \quad \% \ na = nb = 2 \ ve \ nk = 1$

Varsayılan lineer olmayan kestirimci wavelet networkdür, bu kestirimci lineer ve lineer olmayan fonksiyonlar için bütün standart regresörleri girişler olarak alacaktır. m bir nlarx nesnesidir. MIMO sistemler için nb,nf ve nk matrislerdir.

2.1.6 Model regresörlerinin yapılandırılması

Standart regresörlerde farklı model regresörleri ile model yapısı yaratmak için model derecesi değiştirilmelidir, burada gecikmiş giriş ve çıkış değişkenleri lineer olmayan kestirimcinin girişleridir. Model giriş çıkış değişkenleri cinsinden özel regresörleri kurmak için polyreg veya customreg kullanılmalıdır. Var olan bir model için Customregressors özelliklerini kullanılarak özel regresörleri belirtebiliriz.

Bütün model regresörler lineer olmayan kestirimcinin lineer ve lineer olmayan fonksiyon blokları için girişleri olarak girmektedir. Modelin karmaşıklığını azaltmak için ve kestirimi iyi durumda tutmak için lineer olmayan kestirimci bloğunun lineer olmayan fonksiyonu için regresörlerin alt kümeleri kullanılmaktadır.

Lineer olmayan ARX grafikleri lineer olmayan modelin bir veya iki regresörlerinin fonksiyonu olarak karakteristiklerini göstermektedir. Lineer olmayan ARX modelin grafiklerini inceleyerek hangi regresörün model çıkışı üzerinde daha çok etkili olduğunu görebiliriz. Çıkış regresörlerinin önemli ilişkilerini anlayarak hangi regresörün lineer olmayan fonksiyon içerdiğine karar verebiliriz. Ayrıca, farklı lineer olmayan kestirimciler kullanarak aynı veri için lineer olmayan birkaç model modellenebilir. Örnek olarak wavelet network ve treepartition kullanılabilir ve daha sonra bu modellerin lineer olmayan yüzlerinin karşılaştırması yapılır. Lineer

olmayan yüzeyler arasında uygunluk arttığında bu lineer olmayan modellerin sistemin dinamiğini doğru bir şekilde yakaladığını gösterir.

2.2 Grey – Box Model

2.2.1 Grey-Box modellerin yapısı

Sistemin fiziği iyi bir şekilde anlaşıldığında ve sistemin bilinmeyen parametreli diferansiyel denklemlerini kullanarak ifade edildiğinde parametre kesitirimi için lineer veya lineer olmayan grey-box model yapısı uygulanabilmektedir. Grey-box model, bilinen parametre değerleri ve parametreler arasındaki bağlantıyı içererek modelin yapısını açık bir şekilde belirtmektedir. Değişkenler arasında ilişkileri, model davranışındaki sınırlamaları veya sistem dinamiğini tanımlayan denklemleri bildiğimizde grey-box modelleme en uygunudur. Bu model sürekli zaman ve ayrık zaman modellerinin her ikisini desteklemektedir. Fiziksel kanunların bir çoğu sürekli zaman göre açıklandığına göre sürekli zaman için modeli yapılandırmak ayrık zaman için yapılandırmaktan daha kolaydır.

Dinamik giriş çıkışlı modeller için ayrıca zaman serisi modelleri yaratılabilir, bu modeller girişler olmayabilir ve statik modellerde durum denklemleri olamayabilir. Bilinen fiziksel kanunları kullanarak sistemi tanımlamak zor olduğunda lineer olmayan ARX model uygulanmaktadır.

Sürekli zaman ve ayrık zamanlı grey-box modelleri, girişleri olmayan zaman serisi verileri içeren zaman tabanlı veya frekans tabanlı veriler ve tek çıkış veya çok çıkışlı veriler için kestirilebilir. Ayrıca lineer olmayan grey-box modeller sadece zaman tabanlı verileri destekler.

Sistemi modellemek için ilk olarak verilerin Matlab çalışma alanına alınması gerekmektedir. Komut çalışması tercih edildiği zaman verileri iddata veya idfrd nesneleri ile temsil edilmesi gerekir. Grey-box modeller, ODE model yapısının bir dosya içinde belirtilmesini gerektirir. Bu dosya idgrey veya idnlgrey model nesnesi yaratmak için kullanılmaktadır. Lineer sistemlerin modelleri için idgrey ve idnlgrey nesnelerinin her ikisi kullanılmaktadır. Ama lineer olmayan dinamikler için sadece idnlgrey model nesnesi kullanılmaktadır. idgrey nesnesi lineer dinamikleri tanımlayan fonksiyonun durum uzay formu halinde yazılmasını gerektirir. Bu durumda durum uzay matrisleri sistem parametrelerinin fonksiyonuna dönüşmektedir.

idnlgrey nesnesi sistem dinamiklerini tanımlamak için birinci dereceden diferansiyel denklemleri grup olarak fonksiyon dosyasında yazılmasını gerektirir. Bu dosya çıkış ve durum türevlerini zamanın, girişin, durumun ve parametrelerin fonksiyonuna dönüştürmektedir.

2.2.2 Lineer olmayan Grey-Box modellerin kestirimi

Lineer olmayan ayrık zamanlı ve sürekli zamanlı grey-box modeller rastgele lineer olmayan diferansiyel denklemler için tek çıkış ve çok çıkış zaman tabanlı verilerini veya sadece zaman serisi verilerini kullanarak kestirilebilir. Grey-box modeller statik veya dinamik olabilir.

Grey-box modeller, sistem davranışını bilinmeyen parametreli lineer olmayan diferansiyel denklemler grubu olarak tanımlamaktadır.

2.2.3 Lineer olmayan Grey-Box model yapısı

Sistemi aşağıdaki gibi birinci dereceden lineer olmayan diferansiyel denklemler gurubu olarak ifade etmek zorundayız:

$$x^{\dagger} = F\left(t, x(t), u(t), par1, par2, ..., parN\right)$$
$$y(t) = H\left(t, x(t), u(t), par1, par2, ..., parN\right) + e(t)$$
$$x(0) = x_0$$

Burada $x^{\dagger}(t) = {dx(t)}/{dt}$ ifadesi sürekli zaman için geçerlidir, $x^{\dagger}(t) = x(t + T_s)$ ifadesi ise ayrık zaman için geçerlidir, T_s örnekleme aralığını ifade etmektedir. F ve H terimleri sırasıyla N_x ve N_y bileşenlerinin rastgele lineer veya lineer olmayan fonksiyonlarıdır. N_x durum sayısını ve N_y çıkış sayısını göstermektedir.

Sistemin denklemlerini elde ettikten sonra Matlab'da fonksiyon dosyası yazılmaktadır. Model dosyasının amacı aşağıdaki gibi durum türevlerini ve model çıkışlarını zamanın, durumların, girişlerin ve model parametrelerinin fonksiyonuna dönüştürmektir.

$$[\mathrm{dx}, \mathrm{y}] = \mathrm{MODFILENAME}(\mathrm{t}, \mathrm{x}, \mathrm{u}, \mathrm{p1}, \mathrm{p2}, ..., \mathrm{pN}, \mathrm{FileArgument})$$

Çıkış değişkenlerindeki dx durum denklemlerinin sağ tarafını temsil etmektedir. Sütun vektörü olarak N_x ile ifade edilmektedir. Statik modeller için $dx = [\ ]$ ile gösterilir. Ayrık zamanlı modeller için dx bir sonraki adım olan $x(t+Ts)$ de durum denklemlerinin değeridir. Sürekli zaman için dx , t veya $\frac{dx}{dt}$ zamanında durumun türevleridir. y çıkış denklemlerinin sağ tarafını ifade etmektedir. Sütun vektörü N_y olarak ifade edilmektedir.

Dosya girişlerindeki parametrelerden t zamanı, x parametresi t zamanındaki durum vektörünü ifade etmektedir. Statik modeller için $[\ ]$ eşittir. Daha sonra u parametresi t zamanındaki giriş vektörünü ifade etmektedir. Zaman serisi modeller için $[\ ]$ eşittir. $p_1, p_2, ..., p_N$ terimleri parametreler, gerçek skalerler, sütun vektörleri veya iki boyutlu matrisler olabilir. N parametre sayısını göstermektedir. Skaler parametreler için N parametre elemanlarının toplam sayısıdır. FileArgument yardımcı değişkenler içermektedir, bu değişkenler durum denklemlerindeki sabitleri yenilemek için gerekmektedir.

2.2.4 IDNLGREY komutunun yapılandırılması

Model yapısı ile fonksiyon dosyasını yarattıktan sonra idnlgrey komutunu tanımlamak gerekir. Bu komut lineer idgrey model komutunun birçok özelliğini paylaşmaktadır. idnlgrey komutunu tanımlamak için aşağıdaki format kullanılmaktadır:

m = idnlgrey('filename',Order,Parameters,InitialStates)

idnlgrey komutunun argümanlarından 'filename' – model yapısını içeren fonksiyon dosyasının ismidir. Model kestirimi veya simülasyonu için modeli kullandığımızda bu dosya Matlab dosyaları içinde olmalıdır. Order terimi $[Ny\ Nu\ Nx]$ vektörünü ifade etmektedir, Ny model çıkışlarının sayısını, Nu model girişlerinin sayısını ve Nx model durumlarının sayısını belirtmektedir. Parameters ifadesi parametreler, yapı dizilerini, hücre dizilerini veya çift dizileri belirtmektedir. InitialStates ifadesi parametreler olarak aynı yolu belirtmektedir. idngrey yapısı için dördüncü giriş olmalıdır. Grey-box modelin kestirimi için pem kullanılmaktadır.

2.2.5 Lineer olmayan Grey-Box modellerin kestirimi için pem kullanımı

Bilinmeyen idnlgrey model parametrelerini ve ölçülmüş verileri kullanarak ilk durumları kestirmek için pem komutu kullanılmaktadır. idnlgrey model için verinin giriş-çıkış boyutları ile giriş ve çıkış sayıları uyumlu olmalıdır.

Kestirim için aşağıdaki genelleştirilmiş komutlar kullanılmaktadır:

m = pem(data,m)

burada veriler kestirim verileridir ve m yapılandırılmış idnlgrey model komutudur.

Modelin veya algoritma kestiriminin özelliklerini belirtmek için pem komutuna ek özellik-değer çiftlerini tanımlamak gerekmeyebilir. Yüklenebilir özellikler get(idnlgrey) komutu tarafından tek-tek dönüştürülebilir ve algoritma özellikleri get(idnlgrey, 'Algorithm') tarafından dönüştürülebilir. Örnek olarak, MaxIter ve Tolerance verilebilir.

Kestirim algoritmasını belirlerken modelin algoritma özellikleri kestirim hatasını azaltmak için çeşitli parametreler deneyerek modelin birkaç defa simülasyonunu yapmaktadır.

Simülasyon Metodu, Arama Metodu, Gradient Seçenekleri gibi algoritma özellikleri sonuçların kalitesini etkileyebilmektedir.

Model Algoritma özelliğinin SimulationOptions (struct) alanları kullanarak simülasyon metodu belirtilebilmektedir. Sistem tanımlama komutları idnlgrey modellerinin simülasyonu için birkaç değişken adım ve sabit adım çözücüleri sağlamaktadır. Çözücülerin ve onların özelliklerinin listesini görmek için aşağıdaki komut çalıştırılabilir:

idprops idnlgrey algorithm.simulationoptions

Ayrık zamanlı sistemler için temel çözücü 'FixedStepDiscrete' dür. Sürekli zamanlı sistemler için temel çözücü 'ode45' dir. Varsayılan SimulationOptions.Solver çözücüsü 'Auto' için ayarlanmıştır, kestirim ve simülasyon esnasında sistemin ayrık zamanlı veya sürekli zamanlı olmasına karşılık otomatik olarak 'ode45' veya 'FixedStepDiscrete' den birisini seçmektedir.

Algoritma özelliklerinin SearchMethod komutunu kullanarak model parametrelerinin kestirimi için arama metodu belirlenmektedir. Lineer olmayan grey-box modelleme

için metotlar iki kategoride değerlendirilmektedir. Metotların birinci kategorisi çizgi arama metoduna dayanarak minimizasyonu içermektedir, bu minimize metotları Gauss-Newton metotları, steepest-descent metotları ve Levenberg-Marquardt metotlarını içermektedir. Trust-Region Reflective Newton metodu ölçülen ve simülasyon yapılmış çıkışlar arasındaki hataların karelerinin toplamını ifade eden lineer olmayan en küçük kareler (lsqnonlin) yöntemini kullanmaktadır. Parametre sınırları varsayılandan farklı olduğu zaman bu arama metodu çizgi arama metoduna göre daha iyi sonuçlar vermektedir. Buna karşılık çizgi aramaya dayalı metotların aksine lsqnonlin sadece Criterion='Trace' ile çalışmaktadır. Normalde SearchMethod, Auto için ayarlanmıştır ve otomatik olarak elde olan minimize edicilerden bir metot seçmektedir. Eğer optimizasyon verilerimiz yüklü ise SearchMethod metot olarak 'lsqnonlin' seçecektir.

Algoritma özelliklerinden GradientOptions kullanarak gradyantların hesaplanması için metotlar belirlenmelidir. Gradyantlar bilinmeyen parametreler ve ilk durumlar için hataların türevleri olarak adlandırılır. Gradyantlar bozucu bilinmeyen büyüklükleri nümerik olarak hesaplar ve simülasyon hatası üzerindeki etkilerini ölçer. Gradyant hesabı için seçim taslağı bilinmeyen büyüklüklerin minimum bozucu boyutu ve acaba gradyant aynı anda mı hesaplandı yoksa bireysel olarak mı hesaplandı gibi farklı seçimler içermektedir.

Model algoritması kestirim komutu içerisinde direk olarak belirtilebilmektedir. Örnek olarak, pem komutunun parçası olarak algoritma özelliklerini aşağıdaki gibi belirtilebilmektedir.

```
m = pem(data,init_model,'Search','gn',...
                    'MaxIter',5,...
                    'Display','On')
```

Lineer ve lineer olmayan grey-box modellerin kestiriminden sonra sim komutunu kullanarak model çıkışlarının simülasyonunu yapabiliriz.

3. Optimizasyon Metotları

3.1 Trust-Region Metodu

Trust region metodu garanti yakınsama için bir taslak oluşturmaktadır. Bu metot lineer olmayan en küçük kareler problemlerini çözmek için kullanmaktadır, son yıllarda daha çok genel optimizasyon problemlerinin çözümünde karşılaşmaktayız.

Trust-region metodu hedeflenen fonksiyonun modeli için belirgin referans oluşturmaktadır. Newton metotları için bu model kuadratik bir modeldir ve x_k noktası etrafında Taylor serisinden elde edilmiştir:

$$q_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T \nabla^2 f(x_k) p \qquad \textbf{(3.1)}$$

Bu metot sadece modeli sınırlı x_k noktaları etrafında analiz etmektedir ve sınırlar aşağıda tanımlanmıştır:

$\|p\| \leq \Delta_k$

Bu ifade x_k dan x_{k+1} e kadar olan adım boyutu içinde modeli çalıştırmaktadır. Δ_k nın değeri model $q_k(p)$ ve belirtilen fonksiyon $f(x_k + p)$ arasındaki uyuşmaya göre ayarlanmaktadır. Eğer uyum iyi ise model güvenilirdir ve Δ_k arttırılmaktadır. Eğer uyum iyi değil ise Δ_k azaltılmaktadır. Metodun ilk iterasyonlarında x_k değeri x_* dan uzak olabilir, bu durumda Δ_k değeri küçük olabilir ve Newton adımının tam bir şekilde almasını engelleyebilir. Aksi durumda iterasyonlarda x_k değeri x_* değerine çok yakınsa model için metodun iyi çalıştığı düşünülmektedir. Δ_k değeri yeteri kadar büyük olabilir, bu Newton metodunu engellemez ve kuadratik yakınsama oranı elde edilebilir.

Trust-region metodunun k iterasyonunda adım boyu için aşağıdaki problem çözülmelidir:

$$q_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T \nabla^2 f(x_k) p \quad ,$$

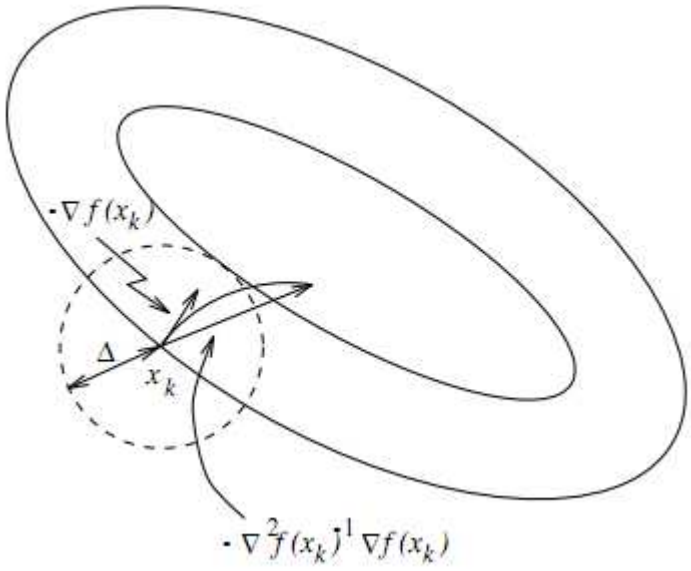
$\|p\| \le \Delta_k$ tanımlı aralığında p'nin minimize edilmesi gerekmektedir

Bu bir sınırlandırılmış optimizasyon problemidir. Bu problem için optimal durumlar gösteriyor ki p_k lineer sistemin çözümü olacaktır.

$$\left(\nabla^2 f(x_k) + \lambda I\right) p_k = -\nabla f(x_k) \qquad \textbf{(3.2)}$$

Burada $\lambda \ge 0$ skalerdir (sınırlar için Lagrange çarpanı olarak adlandırılır), $\left(\nabla^2 f(x_k) + \lambda I\right)$ pozitif tam değerdir, ve $\lambda\left(\Delta_k - \|p_k\|\right) = 0$

Burada bu sonuç türetilmeyecek,



Şekil 3.1: Trust-region eğrisi için parça-parça lineer yaklaşım

Eğer $\nabla^2 f(x_k)$ pozitif tam değer ise ve Δ_k yeterli kadar büyük ise problemin çözümü aşağıdaki gibidir.

$$\nabla^2 f(x_k) p = -\nabla f(x_k) \qquad \textbf{(3.3)}$$

Bu varsayımlar olmazsa metot aşağıdaki gibi olur:

$$\Delta_k \geq \|p_k\| = \left\| \left(\nabla^2 f(x_k) + \lambda I \right)^{-1} \nabla f(x_k) \right\| \qquad \textbf{(3.4)}$$

Ve bu durumda $\Delta_k \to 0$, sonra $\lambda \to \infty$ olmaktadır.

Bu durumda $p_k \approx -\frac{1}{\lambda} \nabla f(x_k)$ elde edilir.

Bunun sonucu olarak p_k, λ nın fonksiyonudur ve Δ_k nın endirek fonksiyonudur. λ değeri 0 ve ∞ arasında değiştiğinde bu $p_k = p_k(\lambda)$ nin Newton yönü ve $-\nabla f(x_k)$ çarpımı arasında sürekli bir halde değiştiğini söyleyebiliriz. Buna örnek yukarıdaki şekilde verilmiştir. Bu şekildeki eğri $p_k(\lambda)$ nin değerlerini göstermektedir. $\lambda \to \infty$ gibi $p_k(\lambda)$ noktaları negatif gradyant yönündedir. $\lambda = 0$ için $p_k(0)$ Newton yönündedir.

Bu yaklaşım çizgi arama metodunun tam zıttıdır, burada arama yönü seçilmişti (Newton denklemlerini kullanarak) ama adım uzunluğu hesaplandıktan sonra bu yön sabitlenmiştir. Trust-region metodunda Δ_k sınırının seçimi adım uzunluğunu ve p_k adım yönünü etkilemektedir.(adım uzunluğu her zaman için bir veya sıfırdır.)

Newton'un metoduna dayanarak bir trust region algoritmasının adımları aşağıdaki gibi belirtilmektedir.

3.1.1 Trust-Region algoritması

1. x_0 ın çözümünün bazı ilk tahmini değerleri belirtilmektedir. $\Delta_0 > 0$ olarak ilk trust-region sınırı seçilmektedir. $0 < \mu < \eta < 1$ sabitleri belirtilmektedir. (belki $\mu = \frac{1}{4}$ ve $\eta = \frac{3}{4}$ olarak belirtilir.)

2. $k = 0,1,...$ için

(i) Eğer x_k optimal ise, dur.

(ii) Çözüm, p_k deneme adımı için

$$\min_{p} imize \quad q_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T \nabla^2 f(x_k) p$$

$$koşul \quad \|p\| \le \Delta_k$$

(iii) Hesapla

$$p_k = \frac{f(x_k) - f(x_k + p_k)}{f(x_k) - q_k(p_k)} = \frac{gerçek\ azalma}{kestirilmiş\ azlam}$$

(iv) Eğer $p_k \le \mu$ ise $x_{k+1} = x_k$ dır (başarısız adım), diğer adım $x_{k+1} = x_k + p_k$ dır (başarılı adım).

(v) Δ_k yenilenmektedir:

$$p_k \le \mu \Rightarrow \Delta_{k+1} = \frac{1}{2}\Delta_k$$
$$\mu < p_k < \eta \Rightarrow \Delta_{k+1} = \Delta_k$$
$$p_k \ge \eta \Rightarrow \Delta_{k+1} = 2\Delta_k$$

p_k değerleri, modelin fonksiyon değerindeki azalmayı ne kadar iyi kestirebildiğini göstermektedir. Eğer p_k değeri küçük ise ($p_k \le \mu$ demektir) fonksiyon değerindeki gerçek azalma $q_k(p_k)$ nin kestiriminden daha küçüktür, bu durum modelin Δ_k kadar geniş aralıkta güvenilir olmayacağını göstermektedir; bu durumda p_k adımı iptal edilecektir ve Δ_k değeri azaltılacaktır. Eğer p_k değeri büyük ise ($p_k \ge \eta$ demektir) model fonksiyon değerindeki azalma uygun bir şekilde kestirilecektir, model geniş bir aralıkta güvenilir olabilir. Bu durumda sınır Δ_k arttırılacaktır.

Aşağıdaki trust-region probleminin çözümü zordur:

$$\min_p imize \quad q_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T \nabla^2 f(x_k) p \qquad \textbf{(3.5)}$$

$$koşul \quad \|p\| \le \Delta_k$$

Örnek olarak, eğer Newton yönü sınır için yeterli değil ise λ değerini bulmak gerekmektedir.

Bu durumda $\|p_k\| = \left\| \left(\nabla^2 f(x_k) + \lambda I \right)^{-1} \nabla f(x_k) \right\| = \Delta_k$ olacaktır.

Bu denklemde lineer değildir. Buna ek olarak λ değerinin $\nabla^2 f(x_k)+\lambda I$ pozitif tamsayı olarak seçilmesi gerekmektedir, buda çözümü zorlaştırmaktadır.

Pratik olarak trust-region metotları trust-region alt problemi için sadece yaklaşık çözüm bulmaktadır. Birinci yaklaşım lineer olmayan denklemler için yaklaşık çözümün bulunmasıdır. Newtonun metodu da uygulanabilir ancak çok etkili özel metotlar türetilmelidir. Hatta bu metotlar sayısal olarak çok pahalıya mahal olabilir. Her bir λ'nın yeniden kestirimi için $\nabla^2 f(x_k)+\lambda I$ matrisi çözülmelidir.

Problemi çözmek için ikinci yaklaşım, yaklaşık olarak $p_k(\lambda)$'dır, adımların eğrisi λ nın 0 ve ∞ arasındaki değişimi olarak tanımlanmıştır. Lineer eğrinin kullanımı genel bir yaklaşımdır, parça-parça çizgilerin bir biri arkasına gelmesidir. $\lambda \approx 0$ için parça çizgi Newton adımı ile çakışmalıdır ve $\lambda \to \infty$ için parça çizgi $-\nabla f(x_k)$ ile çakışmalıdır. Burada art arda gelen parça çizgiler üzerindeki hangi noktanın problemi çözeceğine karar vermek kolaydır. Art arda gelen parça çizgiler bazen "keskin açı" olarak da tanımlanmaktadır.

Line search ve trust-region metotlarının her ikisi optimizasyon için temel hesaplama yöntemlerindendir. Deneyler göstermiştir ki bireysel problemlerde performanslar farklı olmasına rağmen, karşılaştırma olarak her iki metot ortalama etkiye sahiptir. Bazen birinci yaklaşımın diğerine göre tercih edilmesinde ince sonuçlar olabilmektedir.

3.2 Lineer olmayan en küçük kareler problemlerinin çözümü

Bu bölümde aşağıdaki lineer olmayan en küçük kareler problemlerinin çözümü anlatılmıştır:

$$\min_{x \in R^n} f(x) = \frac{1}{2} r(x)^T r(x) = \frac{1}{2} \sum_{i=1}^{m} \left[r_i(x) \right]^2 , m \geq n \qquad \textbf{(3.6)}$$

burada $r : R^n \to R^m$ ifadesi x'in lineer olmayan fonksiyonudur. Eğer $r(x)$ lineer fonksiyon ise problem lineer en küçük kareler problemi olacaktır.
Lineer olmayan en küçük kareler problemi özel yapısı ile sınırlandırılmamış minimizasyon için özel durum gibi dikkate alınmalıdır. Bu problemi bir sistemin m adet lineer olmayan denkleminin çözümü olarak da ifade edebiliriz.

$r_i(x) = 0, i = 1, 2, ..., m$

burada, $r_i(x)$ rastgele fonksiyon olarak tanımlanmaktadır. $m > n$ olduğunda sistem aşırı kararlıdır diye tanımlanır ve $m = n$ olduğunda sistem normal kararlıdır diye tanımlanmaktadır. Lineer olmayan en küçük kareler problemleri parametre kestirimi, fonksiyon tahmini ve başka gibi geniş uygulamalarda kullanılmaktadır.

Örnek olarak, elimizde $(t_1, y_1), (t_2, y_2), ..., (t_m, y_m)$ verisinin olduğunu düşünelim ve bu veriler için bir $\phi(t, x)$ lineer olmayan fonksiyonu bulmaya çalışalım. İlk olarak öyle bir x seçeriz ki $\phi(t, x)$ fonksiyonu veriler için mümkün olduğu kadar rastgele karelerin toplamını minimize etsin.

$$\min \sum_{i=1}^{m} \left[r_i(x) \right]^2 \qquad \textbf{(3.7)}$$

burada $r_i(x) = \phi(t_i, x) - y_i, \quad i = 1, ..., m$ rastgeledir. Genellikle $m \square n$ olmaktadır.Bu yüzden yukarıdaki lineer olmayan en küçük kareler denklemini elde etmekteyiz. Lineer olmayan en küçük kareler problemini çözmek için genellikle Newton metodu kullanılmaktadır. Ancak bu pahalı olmaktadır ve sistem denklemleri çok kolay bir şekilde karmaşık hali almaktadır. Ayrıca yukarıda belirtilen lineer olmayan en küçük kareler problemi özel yapıya sahiptir ve bunun için özel metotlar gerekmektedir. Bu bölümde lineer olmayan en küçük kareler problemini çözmek için bazı efektif ve özel metotlar verilecektir.

$r(x)$'in Jacobian'ı olan $J(x)$ aşağıdaki gibidir:

$$J(x) = \begin{bmatrix} \frac{\partial r_1}{\partial x_1}(x) & \frac{\partial r_1}{\partial x_2}(x) & \dots & \frac{\partial r_1}{\partial x_n}(x) \\ \frac{\partial r_2}{\partial x_1}(x) & \frac{\partial r_2}{\partial x_2}(x) & \dots & \frac{\partial r_2}{\partial x_n}(x) \\ \dots & \dots & \dots & \dots \\ \frac{\partial r_m}{\partial x_1}(x) & \frac{\partial r_m}{\partial x_2}(x) & \dots & \frac{\partial r_m}{\partial x_n}(x) \end{bmatrix} \qquad \textbf{(3.8)}$$

$f(x)$ in gradyantı aşağıdaki gibidir:

$$g(x) = \sum_{i=1}^{m} r_i(x) \nabla r_i(x) = J(x)^T r(x) \qquad \textbf{(3.9)}$$

ve Hessian :

$$G(x)=\sum_{i=1}^{m}\left(\nabla r_i(x)\nabla r_i(x)^T+r_i(x)\nabla^2 r_i(x)\right)=J(x)^T J(x)+S(x) \quad \textbf{(3.10)}$$

Burada

$$S(x)=\sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x) \quad \textbf{(3.11)}$$

Bu durumda $f(x)$ fonksiyonunun kuadratik modeli aşağıdaki gibi olacaktır:

$$\begin{aligned} q^{(k)}(x) &= f(x_k)+g(x_k)^T(x-x_k)+\frac{1}{2}(x-x_k)^T G(x_k)(x-x_k)= \\ &=\frac{1}{2}r(x_k)^T r(x_k)+\left(J(x_k)^T r(x_k)\right)^T(x-x_k)+\frac{1}{2}(x-x_k)^T\times \\ &\left(J(x_k)^T J(x_k)+S(x_k)\right)(x-x_k) \end{aligned} \quad \textbf{(3.12)}$$

$\min_{x\in R^n} f(x)=\frac{1}{2}r(x)^T r(x)=\frac{1}{2}\sum_{i=1}^{m}\left[r_i(x)\right]^2, m\geq n$ denklemi için Newton metodunu kullanalım:

$$x_{k+1}=x_k-\left(J(x_k)^T J(x_k)+S(x_k)\right)^{-1} J(x_k)^T r(x_k) \quad \textbf{(3.13)}$$

Bu iterasyon kuadratik olarak yakınsaktır. Ancak yukarıdaki Newton metodunun ana dezavantajı $G(x)$ Hessian'in ikinci dereceden $S(x)$ teriminin hesaplanması zor ve pahalıdır. Bu ayrıca bütün $G(x)$ in kestiriminde kullanmak için uygun değildir çünkü $J(x)$ ve $G(x)$ denklemindeki birinci dereceden terim olan $J(x)^T J(x)$, gradyantı hesaplarken kullanılabilir durumda olmaktadır. Ancak hesaplamayı azaltmak için $S(x)$ i ihmal etmek veya $S(x)$ i kestirmek için birinci dereceden türev kullanmak daha uygun ve efektif olabilir. $r_i(x)$ sıfıra yaklaştığında veya lineer fonksiyona yaklaştığında $\nabla^2 r_i(x)$ terimi de sıfıra yaklaşır. $S(x)$ terimi küçüktür ve ihmal edilebilir. Bu durum küçük artık problem, aksi halde büyük artık problem olarak tanımlanmaktadır.

3.3 Gauss-Newton Metodu

Bu bölümde Gauss-Newton metodu anlatılacaktır. Bu metot kuadratik modeldeki gradyant $G(x)$in terimi olan ikinci dereceden $S(x)$ in ihmal edilmesi ile elde edilmiştir. Bu durumda aşağıdaki kuadratik form elde edilmiştir:

$$q^{(k)}(x)=\frac{1}{2}r(x_k)^T r(x_k)+\left(J(x_k)^T r(x_k)\right)^T (x-x_k)+\frac{1}{2}(x-x_k)^T \times \\ \times\left(J(x_k)^T J(x_k)\right)(x-x_k) \qquad \textbf{(3.14)}$$

Bu durumda iterasyon aşağıdaki gibi olur:

$$x_{k+1}=x_k+s_k=x_k-\left(J(x_k)^T J(x_k)\right)^{-1} J(x_k)^T r(x_k) \qquad \textbf{(3.15)}$$

İterasyonu başarılı bir şekilde yapmak için $J(x_k)$ Jacobian matrisinin rankı full olmalıdır. Aşağıda Gauss-Newton algoritması anlatılmıştır:

Adım 0 : $x_0,\ \varepsilon>0, k:=0$ verilmiştir.

Adım 1 : Eğer $\|g_k\|\le\varepsilon$ ise dur.

Adım 2 : s_k için $J(x_k)^T J(x_k)s=-J(x_k)^T r(x_k)$ denklemini çöz.

Adım 3 : $x_{k+1}=x_k+s_k,\ k:=k+1$ ayarla. Adım 1'e git.

Açık bir şekilde $J(x_k)$ matrisinin rankı full olduğunda ve gradyant $g(x_k)$ sıfır olmadığında arama yönü s_k, f fonksiyonu için azalma yönündedir çünkü

$$s_k{}^T\nabla f(x_k)=s_k{}^T J(x_k)^T r(x_k)=-s_k{}^T J(x_k)^T J(x_k)s_k\le 0$$

En son eşitsizlik $J(x_k)^T s_k=0$ olmazsa değişmezdir, bu da $J(x_k)^T r(x_k)=g(x_k)=0$ eşitliğini ifade etmektedir.

$J(x_k)^T J(x_k)s=-J(x_k)^T r(x_k)$ denklemi Gauss-Newton denklemidir.

$J(x_k)^T J(x_k)s=-J(x_k)^T r(x_k)$ ve

$x_{k+1}=x_k-\left(J(x_k)^T J(x_k)+S(x_k)\right)^{-1} J(x_k)^T r(x_k)$ denklemleri karşılaştırıldığında Gauss-Newton metodu ile Newton metodu arasındaki fark görülmektedir.

Denklemdeki birinci dereceli $J(x_k)^T J(x_k)$ terimi ile Hessian $G(x_k)$ terimi yer değiştirmiştir.

Yukarıdaki algoritmada adım 2 lineer en küçük karelerin normal denklemlerine benzemektedir. Bunun yanında modelin x_k yakınında $r(x)$ modeli ile örtüştüğü düşünülmektedir.

$$\bar{M} = r(x_k) + J(x_k)(x - x_k) \qquad \textbf{(3.16)}$$

Ve lineer en küçük kareler problemi çözüldüğünde

$$\min \frac{1}{2} \left\| \bar{M}_k(x) \right\|^2 \qquad \textbf{(3.17)}$$

Bu iki incelemeler Gauss-Newton metodunun lineer olmayan en küçük kareler problemi için lineerleştirme olduğunu ortaya çıkarmaktadır.

$x_{k+1} = x_k + s_k = x_k - \left(J(x_k)^T J(x_k) \right)^{-1} J(x_k)^T r(x_k)$ bu denklemden anlaşıldığı gibi Gauss-Newton metodunun bazı avantajları vardır, $r(x)$ fonksiyonunun birinci dereceli türevinin ele alınması ve $J(x)^T J(x)$ pozitif ve tam sayı olması gerekmektedir.

Newton metodunda standart varsayımlar kuadratik olarak yakınsamaktadır, Gauss-Newton metodunun başarılı olması $G(x)$ deki ikinci dereceden $S(x)$ teriminin ihmal edilmesinin önemine bağlıdır. Bir çok teorem ispatları aşağıdaki sonuçları sağlamaktadır:

1. Eğer $S(x^*) = 0$ ise Gauss-Newton metodu kuadratik olarak yakınsaktır.
2. Eğer $J(x^*)^T J(x^*)$ e göre $S(x^*)$ küçük ise Gauss-Newton metodu lineer olarak yakınsaktır.
3. Eğer $S(x^*)$ çok büyükse Gauss-Newton metodu yakınsak olmayacaktır.

Pratikte genellikle Gauss-Newton metodu çizgi arama ile kullanılmaktadır:

$x_{k+1} = x_k - \alpha_k \left(J(x_k)^T J(x_k) \right)^{-1} J(x_k)^T r(x_k)$ bu denklem sönümlü Gauss-Newton metodu olarak adlandırılmaktadır, burada α_k adım genişliğidir. Görüldüğü gibi bu metot her adımda fonksiyonun adım boyunun azalacağı ve bu yüzden yakınsayacağını garanti etmektedir.

Bu bölümün sonucu olarak Gauss-Newton metodu için bazı nümerik bakış açıları incelenecektir. Gauss-Newton denklemlerini çözmek için bu öneriler gerekmektedir.

$$J(x_k)^T J(x_k) s = -J(x_k)^T r(x_k) \quad \textbf{(3.18)}$$

Genellikle yukarıdaki denklemi çözmek yerine matris çarpanlarına ayırma yöntemi kullanılmaktadır. Sonra çözüm için bulunanlar yerine koyma tekniği kullanılmaktadır. Bu şekilde nümerik olarak kesin sonuç büyük ölçüde geliştirilmiş olacaktır. Bunu görmek için hata analizi aşağıdaki gibi yapılmaktadır:

$$\frac{\|\delta s\|}{\|s\|} \le k\left(J(x_k)^T J(x_k)\right) \frac{\|E\|}{\left\|J(x_k)^T J(x_k)\right\|} \quad \textbf{(3.19)}$$

Burada

$$k\left(J(x_k)^T J(x_k)\right) = \frac{\sigma_1^{\,2}}{\sigma_n^{\,2}} \quad \textbf{(3.20)}$$

δs ve E sırasıyla s ve $J(x_k)^T J(x_k)$ nın hatalarını vermektedir ve σ_1 ve σ_n değerleri $J(x_k)$ nın sırasıyla en büyük ve en küçük tekil değerleridir.

Eğer arttırılmış matris için QR çarpanlarına ayırırsak $\left[J(x_k) \quad r_k \right] = Q \left[R \quad Q^T r_k \right]$ elde edilir. Burada Q ortagonal matristir,

$$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

Ve R_1 tekil olmayan üst üçgen matristir. Daha sonra aşağıdaki ifade elde edilmektedir;

$$J(x_k)^T J(x_k) = R^T R = R_1^{\,T} R_1 \quad \textbf{(3.21)}$$

$J(x_k)^T J(x_k) s = -J(x_k)^T r(x_k)$ denkleminin çözümü aşağıdaki elde edilmektedir:

$$R_1 s = -\left(Q^T r_k\right)_n$$

Burada $(.)_n$ bölmenin ilk n elementini göstermektedir.

$k(R_1) = \frac{\sigma_1}{\sigma_n}$ olarak da gösterilebilmektedir, bu yüzden hatalar ortagonal çarpanlarına ayırma ile ortaya çıktığı gösterilmiştir.

3.4 Levenberg-Marquardt metodu

Genellikle Gauss-Newton metodu pratikte çizgi arama metodu ile çalışmaktadır. Ancak $J(x)$ matrisinin rankı full olmadığı zaman Gauss-Newton metodu iyi bir şekilde çalışamaz veya algoritma alakasız bir noktaya yakınsar.

Bu zorluğun üstesinden gelmek için trust-region metodunun çalıştırılması düşünülmüştür. Gerçekte Gauss-Newton metodunun $\bar{M} = r(x_k) + J(x_k)(x - x_k)$ lineerleştirilmiş modeli kullanarak lineer olmayan $r(x)$ fonksiyonunu değiştirmiştir ve lineer en küçük kareler problemi elde edilmiştir. Maalesef bu lineerleştirme bütün $(x - x_k)$ için efektif değildir. Bu yüzden trust-region sınırı koyulmuştur ve aşağıdaki trust-region metodu düşünülmüştür:

$$\begin{aligned} &\min \frac{1}{2}\left\|J(x_k)(x - x_k) + r(x_k)\right\|_2^2 \\ &\left\|x - x_k\right\|_2 \le \Delta_k \end{aligned} \qquad \textbf{(3.22)}$$

Bu denklem sınırlandırılmış lineer en küçük kareler problemidir. Bu modeller aşağıdaki gibi yazılabilir:

$$\begin{aligned} &\min q_k(x) = \frac{1}{2}\left\|r_k\right\|^2 + r_k{}^T J(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T J(x_k)^T (x - x_k) \\ &\left\|x - x_k\right\|_2 \le \Delta_k \end{aligned} \qquad \textbf{(3.23)}$$

Yukarıdaki denklemlerin çözümü sistemin çözülmesi ile karakterize edilmektedir.

$$\left(J(x_k)^T J(x_k) + \mu_k I\right) s = -J(x_k)^T r(x_k) \qquad \textbf{(3.24)}$$

Bunun sonucu olarak aşağıdaki ifade elde edilmektedir:

$$x_{k+1} = x_k - \left(J(x_k)^T J(x_k) + \mu_k I \right)^{-1} J(x_k)^T r(x_k) \qquad \textbf{(3.25)}$$

$\left\| \left(J(x_k)^T J(x_k) \right)^{-1} J(x_k)^T r(x_k) \right\| \le \Delta_k$ olduğunda $\mu_k = 0$ elde edilir ve s_k ile problem çözülmüştür. Aksi takdirde $\mu_k > 0$ olacaktır ki bu durumda s_k nın çözümü için $\|s_k\| = \Delta_k$ yeterlidir ve $\left(J(x_k)^T J(x_k) + \mu_k I \right) s_k = -J(x_k)^T r(x_k)$ elde edilecektir.

$\left(J(x_k)^T J(x_k) + \mu_k I \right)$ pozitif ve tam sayıdır, s in yönü $\left(J(x_k)^T J(x_k) + \mu_k I \right) s = -J(x_k)^T r(x_k)$ ifadesinin azalma yönünde olacaktır. Bu metot Levenberg-Marquardt metodu olarak adlandırılmaktadır ve Levenberg ve Marquardt tarafından bulunmuştur. Yukarıdaki incelemelerden sonra Levenberg-Marquardt metodunun bir Gauss-Newton metodu olduğunu sadece çizgi aramanın trust-region stratejisi ile yer değiştirilmiş hali olduğu ortaya çıkmaktadır.

Levenberg-Marquardt metodu için diğer bir bakış açısı aşağıda anlatılacaktır. Bu metot ayrıca Gauss-Newton metodu ile steepest descent metodu arasındaki anahtar kuralıdır. Buradan bu metodun arama yönü için iki yönden herhangi birini seçmesine izin verdiği anlaşılmaktadır. $\mu_k = 0$ olduğunda Gauss-Newton yönü azalmaktadır. μ_k çok büyük olduğunda $\left(J(x_k)^T J(x_k) + \mu_k I \right) s = -J(x_k)^T r(x_k)$ denklemi $\mu_k I s = -J(x_k)^T r(x_k)$ için yaklaşmaktadır. Bu ifade yönün steepest descent yönüne yakındır. Ayrıca, I matrisinin yerine bazı pozitif ve diyagonal D_k matrisi de kullanılır. Bu durumda denklem aşağıdaki gibi olacaktır:

$$\left(J(x_k)^T J(x_k) + \mu_k D_k \right) s = -J(x_k)^T r(x_k) \qquad \textbf{(3.26)}$$

Bu durumda, bulunan yön metrik D_k matrisi ile Gauss-Newton yönü ve steepest descent yönünün birleşmesidir.

İleride Levenberg-Marquardt metodunun bazı özellikleri tanımlanacaktır.

$s = s(\mu)$ sonucu $\left(J^T J + \mu I \right) s = -J^T r$ çözümüdür.

Burada yakınsama için $J = J(x)$, $r = r(x)$, ve $g = g(x) = J^T r$ ifadeleri geçerlidir.

3.4.1 Levenberg-Marquardt metodunun yakınsaması

Bu bölümde Levenberg-Marquardt metodunun yakınsaması anlatılacaktır. Yukarıda anlatıldığı gibi Levenberg-Marquardt metodu, trust-region metodunun çerçevesinde tanımlanmaktadır ve analiz edilmektedir. Bu durumda birçok teoreminde desteklediği aşağıdaki algoritmayı tanımlayabiliriz.

Algoritma : (Trust-Region tipi Levenberg-Marquardt Algoritması)

Adım 1: İlk nokta verilmiştir $x_0, \bar{\Delta}, \Delta_0 \in \left(0, \bar{\Delta}\right), \varepsilon \geq 0, 0 < \eta_1 < \eta_2 < 1$ ve $0 < \gamma_1 < 1 < \gamma_2, k := 0$

Adım 2: Eğer $\|g_k\| = \|J_k{}^T r_k\| \leq \varepsilon,$ dur.

Adım 3: Yaklaşık olarak $\min \frac{1}{2}\|J(x_k)(x - x_k) + r(x_k)\|_2^2$ $\|x - x_k\|_2 \leq \Delta_k$ denklemlerini s_k için çöz.

Adım 4: Hesapla $\Pr ed_k = f(x_k) - q_k(s_k)$

$Ared_k = f(x_k) - f(x_k + s_k)$

$r_k = \frac{Ared_k}{\Pr ed_k}$

Adım 5: Eğer $r_k < \eta_1$, $\Delta_k = \gamma_1 \Delta_k$ ayarla ve Adım 3'e git.

Adım 6: $x_{k+1} = x_k + s_k$ ayarla. $\Delta_{k+1} = \begin{cases} \min\left\{\gamma_2 \Delta_k, \bar{\Delta}\right\}, & eğer\, r_k \geq \eta_2 \; ve \|s_k\| = \Delta_k, \\ \Delta_k, & aksi\; takdirde. \end{cases}$

Adım 7: $k := k + 1$ ayarla, Adım 2'ye git.

Algoritmadaki Adım 3 de bahsedilen s_k terimi $\min \frac{1}{2}\|J(x_k)(x - x_k) + r(x_k)\|_2^2$ $\|x - x_k\|_2 \leq \Delta_k$ probleminin yaklaşık çözümünü göstermektedir.

3.4.2 Levenberg-Marquardt metodunun uygulaması

Levenberg-Marquardt metodunun çeşitli uygulamaları vardır. Levenberg-Marquardt metodunu kullanarak aşağıdaki denklemlerin çözümü ile s i bulmak düşünülmüştür.

$$s(\mu) = -\left(J_k{}^T J_k + \mu_k D_k{}^T D_k\right)^{-1} J_k{}^T r_k \qquad \textbf{(3.27)}$$

Bu denklem trust-region problemine benzemektedir. (veya sınırlandırılmış lineer en küçük kareler problemine)

$$\min \|r_k + J_k s\| \; ve \|D_k s\| \le \Delta_k \qquad \textbf{(3.28)}$$

Burada $\Delta_k > 0$ trust-region yarıçapıdır. Eğer J_k tekil ve $\mu_k = 0$ ise $\min \|r_k + J_k s\| \; ve \|D_k s\| \le \Delta_k$ denkleminin çözümü aşağıdaki gibi tanımlanabilir:

$$D_k s(0) = \lim_{\mu_k \to 0^+} D_k s(\mu_k) = -\left(J_k D_k{}^{-1}\right)^+ r_k \qquad \textbf{(3.29)}$$

İki olasılık vardır: $\mu_k = 0$ ve $\|D_k s(0)\| \le \Delta_k$, bu durumda $s(0)$, $\min \|r_k + J_k s\| \; ve \|D_k s\| \le \Delta_k$ denkleminin çözümüdür; veya $\mu_k > 0$ ve $\|D_k s(\mu_k)\| = \Delta_k$, bu durumda $s(\mu_k)$, $\min \|r_k + J_k s\| \; ve \|D_k s\| \le \Delta_k$ denkleminin tek çözümüdür. Bunun sonucu olarak aşağıdaki algoritmayı tanımlayabiliriz.

Algoritma (Levenberg-Marquardt Algoritması):

(a) $\Delta_k > 0$ verilmektedir, $\mu_k \ge 0$ bulunur

$$\left(J_k{}^T J_k + \mu_k D_k\right) s_k = -J_k{}^T r_k$$

$\mu_k = 0$ ve $\|D_k s_k\| \le \Delta_k$ veya $\mu_k > 0$ ve $\|D_k s_k\| = \Delta_k$

(b) Eğer $\|r(x_k + s_k)\| \le \|r(x_k)\|$, $x_{k+1} = x_k + s_k$ ayarla ve J_{k+1} 'i hesapla; aksi durumda $x_{k+1} = x_k$ ve $J_{k+1} = J_k$ hesapla.

(c) Δ_{k+1} ve D_{k+1} 'i hesapla.

Bu algoritmanın nasıl efektif ve güvenilebilir uygulanabilmesi aşağıda verilemektedir.

(1) Trust-region problemi nasıl çözülmektedir:

$\left(J_k{}^T J_k + \mu_k D_k{}^T D_k\right) s = -J_k{}^T r_k$ denklemleri için en kolay yol Cholesky çarpanlarına

ayırma yöntemini kullanmaktır. Ancak $\left(J_k^T J_k + \mu_k D_k^T D_k\right)s = -J_k^T r_k$ denklemindeki katsayıların özel yapısından dolayı QR çarpanlarına ayırmayı kullanmak kolaydır.

$\left(J_k^T J_k + \mu_k D_k^T D_k\right)s = -J_k^T r_k$ denklemi lineer en küçük kareler problemi için sadece normal denklemlerdir

$$\begin{bmatrix} J_k \\ \mu_k^{1/2} D_k \end{bmatrix} s \cong -\begin{bmatrix} r \\ 0 \end{bmatrix} \qquad \textbf{(3.30)}$$

Bu matrisin yapısı için $J_k^T J_k$ ve $D_k^T D_k$ ve onların Cholesky çarpanlarının yerine sütun QR çarpanlarına ayırmayı kullanabiliriz.

Burada lineer en küçük kareler probleminin çözümünü bulmak için QR çarpanlarına ayırmanın iki adımını tanımlayabiliriz.

Birinci Adım: J_k nın QR çarpanlarına ayrılması ve $QJ_k\pi = \begin{bmatrix} T & W \\ 0 & 0 \end{bmatrix}$ elde edilmesi.

Burada Q ortagonaldir, T tekil olmayan $rank\left(T\right) = rank\left(J_k\right)$ ile üst üçgen matristir ve π değişim matrisidir. Eğer $\mu_k = 0$ ise $\begin{bmatrix} J_k \\ \mu_k^{1/2} D_k \end{bmatrix} s \cong -\begin{bmatrix} r \\ 0 \end{bmatrix}$ 'nin çözümü aşağıdaki gibi olacaktır:

$$s = \pi \begin{bmatrix} T^{-1} & 0 \\ 0 & 0 \end{bmatrix} Q r_k \equiv J_k^{-} r_k \qquad \textbf{(3.31)}$$

Burada J_k^{-}, $\{1,3\}$ olarak $J_k J_k^{-} J_k = J_k J_k J_k^{-} = \left(J_k J_k^{-}\right)^T$.

Eğer $\mu_k > 0$ ise $QJ_k\pi = \begin{bmatrix} T & W \\ 0 & 0 \end{bmatrix}$ matrisi aşağıdaki gibi olur:

$\begin{bmatrix} Q & 0 \\ 0 & \pi^T \end{bmatrix} \begin{bmatrix} J_k \\ \mu_k^{1/2} D_k \end{bmatrix} \pi = \begin{bmatrix} R \\ 0 \\ D_\mu \end{bmatrix}$, burada $D_\mu = \mu_k^{1/2} \pi^T D_k \pi$ ve R üst trapozoid matristir,

$\begin{bmatrix} J_k \\ \mu_k^{1/2} D_k \end{bmatrix} s \cong -\begin{bmatrix} r \\ 0 \end{bmatrix}$ ile birleştirilirse $\begin{bmatrix} R \\ 0 \\ D_\mu \end{bmatrix} \pi^T s = -\begin{bmatrix} Q_r \\ 0 \end{bmatrix}$ elde edilmektedir.

İkinci Adım: $\begin{bmatrix} R \\ 0 \\ D_\mu \end{bmatrix} \pi^T s = -\begin{bmatrix} Q_r \\ 0 \end{bmatrix}$ ifadesinde $\frac{n(n+1)}{2}$ dizisini kullanarak D_μ'yü elemek kolaydır.

$W \begin{bmatrix} R \\ 0 \\ D_\mu \end{bmatrix} = \begin{bmatrix} R_\mu \\ 0 \\ 0 \end{bmatrix}$ elde edilmektedir, burada R_μ tekil olmayan üst üçgen matristir ve W bir dönme dizilerinin çarpımıdır.

$\begin{bmatrix} R \\ 0 \\ D_\mu \end{bmatrix} \pi^T s = -\begin{bmatrix} Q_r \\ 0 \end{bmatrix}$ ifadesi $\begin{bmatrix} R_\mu \\ 0 \end{bmatrix} \pi^T s = -W \begin{bmatrix} Q_r \\ 0 \end{bmatrix} \overset{\Delta}{=} \begin{bmatrix} u \\ v \end{bmatrix}$, olmaktadır ve $s = \pi R_\mu^{-1} u$ elde edilmektedir.

(2) Trust-Region yarıçapı Δ_k nasıl yenilenir.

Yukarıda anlatıldığı gibi Δ_k seçimi fonksiyonun gerçek azalması ile kestirilmiş azalmasının oranına bağlıdır. Lineer olmayan en küçük kareler durumunda, bu oran

$p = \frac{\|r(x_k)\|^2 - \|r(x_k + s_k)\|^2}{\|r(x_k)\|^2 - \|r(x_k) + J(x_k) s_k\|^2}$ olarak ifade edilmektedir.

Bu ifade lineerleştirilmiş model ile lineer olmayan fonksiyon arasındaki uygunluğu ölçmektedir. Örnek olarak, eğer $r(x)$ lineer ise $p = 1$ olacaktır. Eğer $J(x_k)^T r(x_k) \neq 0$ ise $\|s_k \to 0\|$ olduğunda $p \to 1$ olacaktır. Eğer $\|r(x_k + s_k)\| \geq \|r(x_k)\|$ ise $p \leq 0$ olacaktır.

Yuvarlama hatası nedeniyle $p = \frac{\|r(x_k)\|^2 - \|r(x_k + s_k)\|^2}{\|r(x_k)\|^2 - \|r(x_k) + J(x_k) s_k\|^2}$ nin hesaplanması zor olabilir. Bu ifade koruyucu form şeklinde yazılacaktır. Her iki tarafı $\left(J_k^T J_k + \mu_k D_k^T D_k\right) s = -J_k^T r_k$ ile çarparsak $-2 r_k^T J_k^T s = 2 s^T J_k^T J_k s + 2 \mu_k s D_k^T D_k s$ elde edilir, buradan $r_k^T r_k - r_k^T r_k - 2 r_k^T J_k^T s - s^T J_k^T J_k s = s^T J_k^T J_k s + 2 \mu_k s^T D_k^T D_k s$ ifadesi yazılarak

$\|r_k\|^2 - \|r_k + J_k s\|^2 = \|J_k s\|^2 + 2\mu_k \|D_k s\|^2$ elde edilmiştir.

Bu denklemleri yerine koyarak $p = \dfrac{1 - \left[\dfrac{\|r(x_k + s_k)\|}{\|r(x_k)\|}\right]^2}{\left[\dfrac{\|J_k s\|}{\|r(x_k)\|}\right]^2 + 2\left[\dfrac{\mu_k^{1/2}\|D_k s\|}{\|r(x_k)\|}\right]^2}$ ifadesi elde edilmektedir.

$\|r_k\|^2 - \|r_k + J_k s\|^2 = \|J_k s\|^2 + 2\mu_k \|D_k s\|^2$ ifadesini $\|J_k s\| \le \|r(x_k)\|, \mu_k^{1/2}\|D_k s\| \le \|r(x_k)\|$ olarak yazabiliriz.

Ayrıca yuvarlama hatası dikkate almadığında oran ifadesinin payı negatif olamayacaktır.

(3) Levenberg-Marquardt parametresinin bulunması.

More algoritmasında eğer $|\phi(\mu)| \le \sigma\Delta, \sigma \in (0,1)$ ise burada $\phi(\mu) = \left\|D\left(J^T J + \mu D^T D\right)^{-1} J^T r\right\| - \Delta$ olarak ele alınmaktadır, bu durumda $\mu > 0$ Levenberg-Marquardt parametresi olarak kabul edilmektedir, burada σ terimi $\|D_k s(\mu)\|$ deki hatayı göstermektedir. Eğer $\phi(0) \le 0$ ise $\mu = 0$ gerekli parametredir. Bunun için biz sadece $\phi(0) > 0$ durumunu tartışmalıyız. ϕ sürekli ve azalan fonksiyon olduğu için $\mu \to \infty$ olduğunda $\phi(\mu) \to -\Delta$ olmaktadır. Buradan $\mu^* > 0$ olduğunda $\phi(\mu^*) = 0$ ifadesi elde edilmektedir. Levenberg-Marquardt parametresine karar vermek için $\mu_0 > 0$ dan başlamalıyız ve $\{\mu_k\} \to \mu^*$ dizisi üretmeliyiz.

$\phi(\mu) = \left\|D\left(J^T J + \mu D^T D\right)^{-1} J^T r\right\| - \Delta$ den $\phi(\mu) = \left\|\left(\bar{J}^T \bar{J} + \mu I\right)^{-1} \bar{J}^T r\right\| - \Delta$ burada $\bar{J} = JD^{-1}$'dir. $\bar{J} = U\sum V^T$ ifadesi $\bar{J}$ nin tekil değeridir ve buradan $\phi(\mu) = \sum_{i=1}^{n}\left[\dfrac{\sigma_i^2 z_i^2}{\left(\sigma_i^2 + \mu\right)^2}\right]^{1/2} - \Delta$ ifadesi elde edilmektedir. Burada $z = U^T r, \sigma_1, ..., \sigma_n$,

J nin tekil değerleridir. Bunun sonucunda $\phi(\mu)=\dfrac{a}{b+\mu}\equiv\phi(\mu)$ olduğu ve a ve b nin $\phi(\mu_k)=\phi(\mu_k),\ \phi'(\mu_k)=\phi'(\mu_k)$ dikkate alarak seçildiği bulunur. $\phi(\mu_{k+1})=0$ olduğunda $\mu_{k+1}=\mu_k-\left[\dfrac{\phi(\mu_k)+\Delta}{\Delta}\right]\left[\dfrac{\phi(\mu_k)}{\phi'(\mu_k)}\right]$

μ_{k+1} in hesaplanması için More algoritması aşağıdaki tekniği tasarlamıştır.

$$u_0=\frac{\left\|\left(JD^{-1}\right)r\right\|}{\Delta}$$

$$l_0=\begin{cases}-\phi(0)/\phi'(0), & \text{eğer } J \text{ tekil değilse}\\ 0, & \text{aksi durumda}\end{cases}$$

(a) Eğer $\mu_k\notin(l_k,u_k),\mu_k=\max\left\{0.001u_k,(l_ku_k)^{1/2}\right\}$ ayarla.

(b) $\phi(\mu_k)$ ve $\phi'(\mu_k)$'yı hesapla.

u_k yı yenile:

$$u_{k+1}=\begin{cases}\mu_k, & \text{eğer } \phi(\mu_k)<0 \text{ ise}\\ u_k, & \text{aksi durumda.}\end{cases}$$

l_k'yı yenile:

$$l_{k+1}=\max\left\{l_k,\mu_k-\frac{\phi(\mu_k)}{\phi'(\mu_k)}\right\}$$

(c) μ_{k+1} terimini $\mu_{k+1}=\mu_k-\left[\dfrac{\phi(\mu_k)+\Delta}{\Delta}\right]\left[\dfrac{\phi(\mu_k)}{\phi'(\mu_k)}\right]$ ifadesinden hesapla.

Yukarıdaki algoritma μ_k nün alt ve üst sınırını vermektedir. (a) şıkkında eğer μ_k nın (l_k,u_k) içinde değilse l_k ile μ_k yer değiştirecektir. (b) şıkkında ϕ nin dışbükeyliği Newton iterasyonu l_k yı yenilemek için kullanılabilmektedir. $\{\mu_k\}$ serisi algoritmanın μ^* e yakınsaması için hesaplanacaktır.

$\phi(\mu)=\left\|\left(J^TJ+\mu I\right)^{-1}J^Tr\right\|-\Delta$ ifadesinden $\phi'(\mu)$ aşağıdaki gibi hesaplanmaktadır:

$\phi'(\mu) = \dfrac{\left(D^T q(\mu)\right)^T \left(J^T J + \mu D^T D\right)^{-1} \left(D^T q(\mu)\right)}{\left\|q(\mu)\right\|}$ burada $q(\mu) = Ds(\mu)$.

$\begin{bmatrix} Q & 0 \\ 0 & \pi^T \end{bmatrix} \begin{bmatrix} J_k \\ \mu_k^{1/2} D_k \end{bmatrix} \pi = \begin{bmatrix} R \\ 0 \\ D_\mu \end{bmatrix}$ ve $W \begin{bmatrix} R \\ 0 \\ D_\mu \end{bmatrix} = \begin{bmatrix} R_\mu \\ 0 \\ 0 \end{bmatrix}$ kullanarak aşağıdaki ifadeyi elde etmekteyiz:

$$\pi^T \left(J^T J + \mu D^T D\right) \pi = R_\mu{}^T R_\mu$$

Buradan $\phi'(\mu) = -\left\|q(\mu)\right\| \left\| R_\mu{}^{-T} \left(\dfrac{\pi^T D^T q(\mu)}{\left\|q(\mu)\right\|} \right) \right\|^2$ elde edilir.

(4) Ölçü matrisinin yenilenmesi:

Levenberg-Marquardt metodunda D_k diyagonal matristir. Algoritmada $D_k = diag\left(d_1^{(k)}, ..., d_n^{(k)}\right)$, burada $\begin{aligned} d_i^{(0)} &= \left\|\partial_i r(x_0)\right\|, \\ d_i^{(k)} &= \max\left\{d_i^{(k-1)}, \left\|\partial_i r(x_k)\right\|\right\}, k \geq 1 \end{aligned}$ ifadeleri geçerlidir.

Son olarak More versiyon Levenberg-Marquardt algoritmasını aşağıdaki gibidir.

Algoritma (More Versiyon):

(a) $\sigma \in (0,1)$ eğer $\left\|D_k J_k^{-} r_k\right\| \leq (1+\sigma)\Delta_k$ ise, $\mu = 0$ ayarla ve $s_k = -J_k^{-} r_k$ dir. Aksi durumda $\mu_k > 0$ belirle

$$\begin{bmatrix} J_k \\ \mu_k^{1/2} D_k \end{bmatrix} s_k \cong -\begin{bmatrix} r_k \\ 0 \end{bmatrix}$$

Buradan $(1-\sigma)\Delta_k \leq \left\|\left\|D_k s_k\right\|\right\| \leq (1+\sigma)\Delta_k$ elde edilmektedir.

(b) Fonksiyonun gerçek azalma ve kestirilmiş azalama arasındaki p_k oran hesaplanmaktadır.

(c) Eğer $p_k \leq 0.0001$ ise $x_{k+1} = x_k$ ve $J_{k+1} = J_k$ ayarlanacaktır.

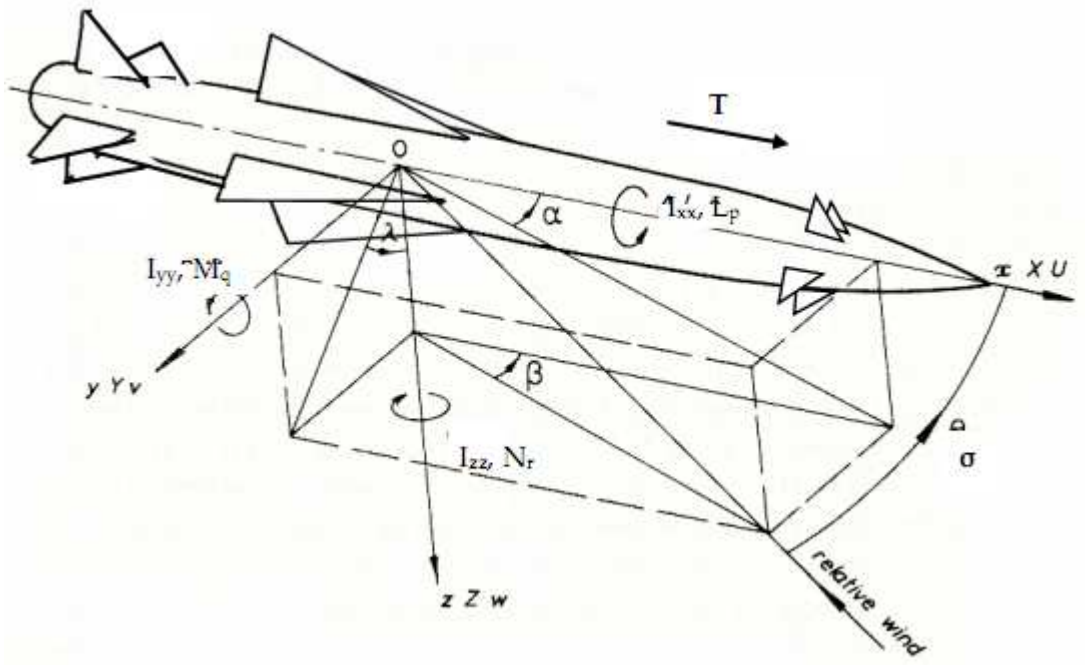
Eğer $p_k > 0.0001$ ise $x_{k+1} = x_k + s_k$ ayarla ve J_{k+1} hesaplanır.

(d) Eğer $p_k \le \frac{1}{4}$ ise $\Delta_{k+1} \in \left[\frac{1}{10}\Delta_k, \frac{1}{2}\Delta_k\right]$. Eğer $p_k \in \left[\frac{1}{4}, \frac{1}{3}\right]$ ve $\mu_k = 0$ ikisinden birisi veya $p_k \ge \frac{3}{4}$ ise $\Delta_{k+1} = 2\|D_k s_k\|$

(e) $D_k = diag\left(d_1^{(k)}, ..., d_n^{(k)}\right)$, ifadesini kullanarak D_k yenile.

4. Füze Body Denklemleri

Aşağıda füze sisteminin genel şeması verilmektedir. Bu şekil doğrultusunda üzerindeki notasyonlar dikkate alınarak aşağıda füzenin dinamik denklemleri açıklanmıştır. Çalışmada kullanılan füze verileri güdüm kontrolü yapılmış gerçek bir füze modelinden elde edilmiştir. Bu verileri kullanarak lineer olmayan sistemin lineer olmayan parametre kestirimi metotları ile ölçülemeyen parametreleri kestirilmiştir.



Şekil 4.1: Füze sisteminin genel şeması

Bu şemada L,M,N parametreleri sırasıyla dönme, yunuslama ve sapma momentlerini göstermektedir. P,Q,R parametreleri açısal hızları ve ϕ,θ,ψ parametreleri ise sırasıyla dönme, yunuslama ve sapma açılarını ifade etmektedir.

Newton kanunlarını kullanarak hareket denklemlerinin aşağıdaki gibi elde ederiz. İlk olarak sistem üzerindeki toplam kuvveti genel olarak $\sum F = \frac{d}{dt}\left[mV_t\right]_I$ ifadesi ile toplam momenti ise $\sum M = \left[\frac{dH}{dt}\right]_I$ ifadesi ile ifade ederiz.

Bu durumda denge denklemleri $\sum F = \sum F_0 + \sum \Delta F$ şeklinde ifade edilecektir. Burada $\sum F_0$ denge kuvvetlerinin toplamını ifade eder. Denge kuvvetlerinin toplamı kaldırma, direnç, itme ve yerçekimi kuvvetlerinin toplamını ifade etmektedir. Aynı şekilde toplam moment ifadesi $\sum M = \sum M_0 + \sum \Delta M$ olarak yazılacaktır. Denge momentlerinin toplamı $\sum M_0$, kaldırma, direnç ve itme momentlerinin toplamıdır.

Bu denklemleri sadeleştirirsek aşağıdaki genel denklemleri elde ederiz.

$$\sum \Delta F = \frac{d}{dt}\left[mV\right]_t \quad ; \quad \sum \Delta M = \left[\frac{dH}{dt}\right]_I \tag{4.1}$$

Dinamik denklemlerin elde edilmesi sürecinde füze kütlesinin sabit olduğu, füzenin rijit bir kütle olduğu, yerküresinin ataletsel referans kabul edildiği ve atmosfer yer küresine göre sabit olduğu kabul edilmiştir.

$$\sum \Delta F = \left[m\frac{dV_T}{dt}\right]_I \quad \Rightarrow \quad \sum \Delta F = \left[m\frac{dV_T}{dt}\right]_E \tag{4.2}$$

Yukarıdaki denklemlerde yer alan toplam vektörel türev alınırsa $\left[\frac{dV_T}{dt}\right]_E = V_T \frac{dV_T}{dt} + \omega \times V_T$ denklemi elde edilmektedir.

Burada ω toplam açısal hızı, $V_T \frac{dV_T}{dt}$ ifadesi lineer hızdaki değişimi göstermektedir.

Buradan $V_T = iU + jV + kW$ ve $\omega = iP + jQ + kR$ olarak ifade edilebilmektedir.

X, Y ve Z eksenleri, i, j ve k birim vektörleri göstermektedir. Yerine koyulduğunda $V_T \frac{dV_T}{dt} = i\dot{U} + j\dot{V} + k\dot{W}$ ifadesi elde edilir.

Hız çarpmı aşağıdaki gibi matris haline dönüştürülmüştür:

$$\omega \times V_T = \begin{vmatrix} i & j & k \\ P & Q & R \\ U & V & W \end{vmatrix} \tag{4.3}$$

$$\omega \times V_T = i\left(WQ - VR\right) + j\left(UR - WP\right) + k\left(VP - UQ\right)$$

Çarpanlarına ayırma ile sadeleştirildiğinde $\sum \Delta F = i\sum \Delta F_x + j\sum \Delta F_y + k\sum \Delta F_z$ toplam kuvvet ifadesi elde edilmektedir.

Son olarak x,y ve z yönündeki lineer hareket denklemleri aşağıdaki gibi elde edilmektedir.

$$\sum \Delta F_x = m\left(\dot{u} + wq - vr\right)$$

$$\sum \Delta F_y = m\left(\dot{v} + ur - wp\right) \qquad \textbf{(4.4)}$$

$$\sum \Delta F_z = m\left(\dot{w} - uq + vp\right)$$

Buradan x,y ve z yönündeki füze kuvvet denklemlerinin sadeleştirilmiş son halini aşağıdaki gibi yazabiliriz:

$$\begin{aligned} m\left(\dot{u} + wq - vr\right) &= X + T + g_x m \\ m\left(\dot{v} + ur - wp\right) &= Y + g_y m \\ m\left(\dot{w} - uq + vp\right) &= Z + g_z m \end{aligned} \qquad \textbf{(4.5)}$$

Bu ifadelerdeki p, q, r parametreleri açısal oranlar, u, v, w parametreleri eksenlere göre hız bileşenlerini, X, Y, Z parametreleri eksenlere göre aerodinamik kuvvet bileşenlerini ifade etmektedir.

Açısal hareket denklemlerini elde etmek için $\sum \Delta M = \left[\frac{dH}{dt}\right]_I$ denklemine gerek vardır. Burada H açısal momentumdur.

$$V_{\tan} = \omega \times R$$

$$dM = \left(\omega \times r\right) dm$$

$$dH = r \times \left(\omega \times r\right) dm; \qquad H = \int dH$$

$$H = \int r \times \left(\omega \times r\right) dm \qquad \textbf{(4.6)}$$

Açısal momentum denklemini elde etmek için aşağıdaki hız çarpımlarını matris halinde yazarak sadeleştirme yapılacaktır.

$$\omega = iP + jQ + kR \quad ; \quad r = ix + jy + kz$$

$$\omega \times r = \begin{vmatrix} i & j & k \\ P & Q & R \\ x & y & z \end{vmatrix} \tag{4.7}$$

$$\omega \times r = i\left(zQ - yR\right) + j\left(xR - zP\right) + k\left(yP - xQ\right)$$

$$r \times \left(\omega \times r\right) = \begin{vmatrix} i & j & k \\ x & y & z \\ zQ - yR & xR - zP & yP - xQ \end{vmatrix} \tag{4.8}$$

$$r \times \left(\omega \times r\right) = i\left[\left(y^2 + z^2\right)P - xyQ - xzR\right] + j\left[\left(z^2 + x^2\right)Q - yzR - xyP\right] + \\ + k\left[\left(x^2 + y^2\right)R - xzP - yzQ\right]$$

Yukarıdaki denklemlerden açısal moment aşağıdaki halini almaktadır.

$$H = \int i\left[\left(y^2 + z^2\right)P - xyQ - xzR\right]dm + \int j\left[\left(z^2 + x^2\right)Q - yzR - xyP\right]dm + \\ + \int k\left[\left(x^2 + y^2\right)R - xzP - yzQ\right]dm \tag{4.9}$$

$$H_x = PI_x - RI_{xz}$$

$$H_y = QI_y$$

$$H_z = RI_z - PI_{xz}$$

Yukarıdaki sadeleştirmeleri aşağıdaki toplam açısal momentum denkleminde yerine koyularak istenen sonuç elde edilmektedir.

$$\sum \Delta M = 1_H \frac{dH}{dt} + \omega \times H \tag{4.10}$$

$$\frac{dH_x}{dt} = \dot{P}\, I_x - \dot{R}\, I_{xz}$$

$$\frac{dH_y}{dt} = \dot{Q}\, I_y$$

$$\frac{dH_z}{dt} = \dot{R} I_z - \dot{P} I_{xz}$$

$$\omega \times H = \begin{vmatrix} i & j & k \\ P & Q & R \\ H_x & H_y & H_z \end{vmatrix}$$

$$\omega \times H = i\left(QH_z - RH_y\right) + j\left(RH_x - PH_z\right) + k\left(PH_y - QH_x\right)$$

Toplam açısal momentum denklemi aşağıdaki gibi ifade edilirse:

$$\sum \Delta M = i\sum \Delta L + j\sum \Delta \mathrm{M} + k\sum \Delta \mathrm{N} \qquad \textbf{(4.11)}$$

Açısal hareket denklemlerinin son hali aşağıdaki gibi elde edilmektedir.

$$\sum \Delta L = \dot{P} I_x - \dot{R} I_{xz} + QR\left(I_z - I_y\right) - PQI_{xz}$$

$$\sum \Delta M = \dot{Q} I_y + PR\left(I_x - I_z\right) + \left(P^2 - R^2\right) I_{xz} \qquad \textbf{(4.12)}$$

$$\sum \Delta N = \dot{R} I_z - \dot{P} I_{xz} + PQ\left(I_y - I_x\right) + QRI_{xz}$$

Bu denklemleri elde etmek bazı kabullere yapılmaktadır. Bu kabulleri OX ve OZ simetri düzlemindedir, füzenin kütlesi sabittir, füze rijit modeldir ve yeryüzü atalet referansıdır olarak özetleyebiliriz.

4.1 Füzenin Yere Göre Davranışı

Yukarıdaki füzenin genel şemasında belirtilen notasyonlarını kullanarak füzenin yere göre davranışı aşağıdaki denklemlerle ifade edilebilmektedir.

x, y ve z yönündeki açısal hızlar aşağıdaki gibi sırasıyla elde edilmektedir.

$$P = \dot{\phi} - \dot{\psi}\sin\theta$$
$$Q = \dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\theta \qquad \textbf{(4.13)}$$
$$R = -\dot{\theta}\sin\phi + \dot{\psi}\cos\theta\cos\phi$$

Bu açısal hızlara göre sapma açılarının türevleri aşağıdaki gibi elde edilmiştir.

$$\dot{\theta} = Q\cos\phi - R\sin\phi$$
$$\dot{\phi} = P + Q\sin\phi\tan\theta + R\cos\phi\tan\theta = P + \dot{\psi}\sin\theta$$

$$\dot{\psi} = Q\frac{\sin\phi}{\cos\theta} + R\frac{\cos\phi}{\cos\theta}$$

Her bir yörünge için yer çekim etkisi aşağıdaki gibidir:

OX : $-mg\sin\theta$

OY : $mg\cos\theta\sin\phi$

OZ : $mg\cos\theta\cos\phi$

Füze moment denklemlerinin sadeleştirilmiş hali aşağıdaki gibidir:

$$L = I_{xx}\dot{p} - \left(I_{yy} - I_{zz}\right)qr + I_{yz}\left(r^2 - q^2\right) - I_{zx}\left(pq + \dot{r}\right) + I_{xy}\left(rp - \dot{q}\right)$$

$$M = I_{yy}\dot{q} - \left(I_{zz} - I_{xx}\right)rp + I_{zx}\left(p^2 - r^2\right) - I_{xy}\left(qr + \dot{p}\right) + I_{yz}\left(pq - \dot{r}\right) \qquad \textbf{(4.14)}$$

$$N = I_{zz}\dot{r} - \left(I_{xx} - I_{yy}\right)pq + I_{xy}\left(q^2 - p^2\right) - I_{yz}\left(rp + \dot{q}\right) + I_{zx}\left(qr - \dot{p}\right)$$

Füzenin durum uzay modelini elde etmek için ilk olarak kuvvet denklemleri sadeleştirilmektedir.

$$\begin{aligned}
m\left(\dot{u} + wq - vr\right) &= X + T + g_x m \quad &\Rightarrow \quad \dot{u} &= \frac{X}{m} + \frac{T}{m} + g_x + vr - wq \\
m\left(\dot{v} + ur - wp\right) &= Y + g_y m \quad &\Rightarrow \quad \dot{v} &= \frac{Y}{m} + g_y + wp - ur \\
m\left(\dot{w} - uq + vp\right) &= Z + g_z m \quad &\Rightarrow \quad \dot{w} &= \frac{Z}{m} + g_z + uq - vp
\end{aligned}$$

Bu denklemleri durum uzay modelinde yazarsak aşağıdaki durum uzay modelini elde ederiz.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} uq \\ ur \\ vp \\ vr \\ wp \\ wq \end{bmatrix} + \begin{bmatrix} \frac{X}{m} + \frac{T}{m} \\ \frac{Y}{m} \\ \frac{Z}{m} \end{bmatrix} + \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \qquad \textbf{(4.15)}$$

Moment denklemleri aşağıdaki gibi sadeleştirilirse durum uzay modeli elde edilmektedir.

$$L = I_{xx}\dot{p} - \left(I_{yy} - I_{zz}\right)qr + I_{yz}\left(r^2 - q^2\right) - I_{zx}\left(pq + \dot{r}\right) + I_{xy}\left(rp - \dot{q}\right)$$

$$M = I_{yy}\dot{q} - \left(I_{zz} - I_{xx}\right)rp + I_{zx}\left(p^2 - r^2\right) - I_{xy}\left(qr + \dot{p}\right) + I_{yz}\left(pq - \dot{r}\right)$$

$$N = I_{zz}\dot{r} - \left(I_{xx} - I_{yy}\right)pq + I_{xy}\left(q^2 - p^2\right) - I_{yz}\left(rp + \dot{q}\right) + I_{zx}\left(qr - \dot{p}\right)$$

$$L = I_{xx}\dot{p} - I_{yy}qr + I_{zz}qr + I_{yz}r^2 - I_{yz}q^2 - I_{zx}pq - I_{zx}\dot{r} + I_{xy}rp - I_{xy}\dot{q}$$

$$M = I_{yy}\dot{q} - I_{zz}rp + I_{xx}rp + I_{zx}p^2 - I_{zx}r^2 - I_{xy}qr - I_{xy}\dot{p} + I_{yz}pq - I_{yz}\dot{r}$$

$$N = I_{zz}\dot{r} - I_{xx}pq + I_{yy}pq + I_{xy}q^2 - I_{xy}p^2 - I_{yz}rp - I_{yz}\dot{q} + I_{zx}qr - I_{zx}\dot{p}$$

Yukarıdaki denklem sadeleştirildiğinde durum uzay modelini kolayca yazabileceğimiz denklemler elde edilmektedir.

$$L = I_{xx}\dot{p} - I_{xy}\dot{q} - I_{zx}\dot{r} - I_{yy}qr + I_{zz}qr - I_{yz}r^2 - I_{yz}q^2 - I_{zx}pq + I_{xy}rp$$

$$M = I_{yy}\dot{q} - I_{xy}\dot{p} - I_{yz}\dot{r} - I_{zz}rp + I_{xx}rp + I_{zx}p^2 - I_{zx}r^2 - I_{xy}qr + I_{yz}pq \qquad \textbf{(4.16)}$$

$$N = I_{zz}\dot{r} - I_{yz}\dot{q} - I_{zx}\dot{p} - I_{xx}pq + I_{yy}pq + I_{xy}q^2 - I_{xy}p^2 - I_{yz}rp + I_{zx}qr$$

Denklemleri durum uzay formuna dönüştürürsek aşağıdaki durum uzay formu elde edilmektedir.

$$\begin{bmatrix} I_{xx} & -I_{xy} & -I_{zx} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{yz} & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & I_{zx} & -I_{xy} & I_{yz} & \left(I_{yy} - I_{zz}\right) & -I_{yz} \\ -I_{zx} & -I_{yz} & \left(I_{zz} - I_{xx}\right) & 0 & I_{xy} & I_{zx} \\ I_{xy} & \left(I_{xx} - I_{yy}\right) & I_{yz} & -I_{xy} & -I_{zx} & 0 \end{bmatrix} \begin{bmatrix} p^2 \\ pq \\ pr \\ q^2 \\ qr \\ r^2 \end{bmatrix} + \begin{bmatrix} L \\ M \\ N \end{bmatrix}$$

$$A=\begin{bmatrix} I_{xx} & -I_{xy} & -I_{zx} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{yz} & I_{zz} \end{bmatrix}$$,

$$B=\begin{bmatrix} 0 & I_{zx} & -I_{xy} & I_{yz} & \left(I_{yy}-I_{zz}\right) & -I_{yz} \\ -I_{zx} & -I_{yz} & \left(I_{zz}-I_{xx}\right) & 0 & I_{xy} & I_{zx} \\ I_{xy} & \left(I_{xx}-I_{yy}\right) & I_{yz} & -I_{xy} & -I_{zx} & 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}=\left[\left[A\right]^{-1}\cdot\left[B\right]\right]\begin{bmatrix} p^2 \\ pq \\ pr \\ q^2 \\ qr \\ r^2 \end{bmatrix}+\left[\left[A\right]^{-1}\right]\begin{bmatrix} L \\ M \\ N \end{bmatrix}$$

$$A^{-1}=\frac{1}{\Delta}\begin{bmatrix} \left(I_{yy}I_{zz}-{I_{yz}}^2\right) & \left(I_{zz}I_{xy}+I_{yz}I_{zx}\right) & \left(I_{yz}I_{xy}+I_{yy}I_{zx}\right) \\ \left(I_{zz}I_{xy}+I_{yz}I_{zx}\right) & \left(I_{xx}I_{zz}-{I_{zx}}^2\right) & \left(I_{xx}I_{yz}+I_{zx}I_{xy}\right) \\ \left(I_{yz}I_{xy}+I_{yy}I_{zx}\right) & \left(I_{xx}I_{yz}+I_{xy}I_{zx}\right) & \left(I_{xx}I_{yy}-{I_{xy}}^2\right) \end{bmatrix}$$

$$\Delta=\left(I_{xx}I_{yy}I_{zz}-I_{xx}{I_{yz}}^2-I_{yy}{I_{zx}}^2-I_{zz}{I_{xy}}^2-2\cdot I_{yz}I_{zx}I_{xy}\right)$$

Altıncı dereceden bir füze sisteminin durum uzay modeli aşağıdaki gibi olmaktadır.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}=\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & & 0 & & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & & 0 & & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & & 0 & & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & & \left[\left[A\right]^{-1}\left[B\right]\right] & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \end{bmatrix}\cdot\begin{bmatrix} uq \\ ur \\ vp \\ vr \\ wp \\ wq \\ p^2 \\ pq \\ pr \\ q^2 \\ qr \\ r^2 \end{bmatrix}+$$

$$+\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & & & \\ 0 & 0 & 0 & & [A]^{-1} & \\ 0 & 0 & 0 & & & \end{bmatrix} \cdot \begin{bmatrix} \bar{X} + \bar{T} \\ \bar{Y} \\ \bar{Z} \\ L \\ M \\ N \end{bmatrix} + \begin{bmatrix} g_x \\ g_y \\ g_z \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \textbf{(4.17)}$$

$$\bar{X} = \frac{X}{m} \ , \ \bar{Y} = \frac{Y}{m} \ , \ \bar{Z} = \frac{Z}{m} \ , \ \bar{T} = \frac{T}{m}$$

Eğer $I_{yz} = I_{zx} = I_{xy} = 0$ ise durum uzay modelini daha da sadeleştirebiliriz.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \bar{I}_{xx} \\ 0 & \bar{I}_{yy} & 0 \\ \bar{I}_{zz} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} pq \\ pr \\ qr \end{bmatrix} + \begin{bmatrix} \bar{L} \\ \bar{M} \\ \bar{N} \end{bmatrix}$$

$$\bar{I}_{xx} = \frac{I_{yy} - I_{zz}}{I_{xx}} \ , \ \bar{I}_{yy} = \frac{I_{zz} - I_{xx}}{I_{yy}} \ , \ \bar{I}_{zz} = \frac{I_{xx} - I_{yy}}{I_{zz}}$$

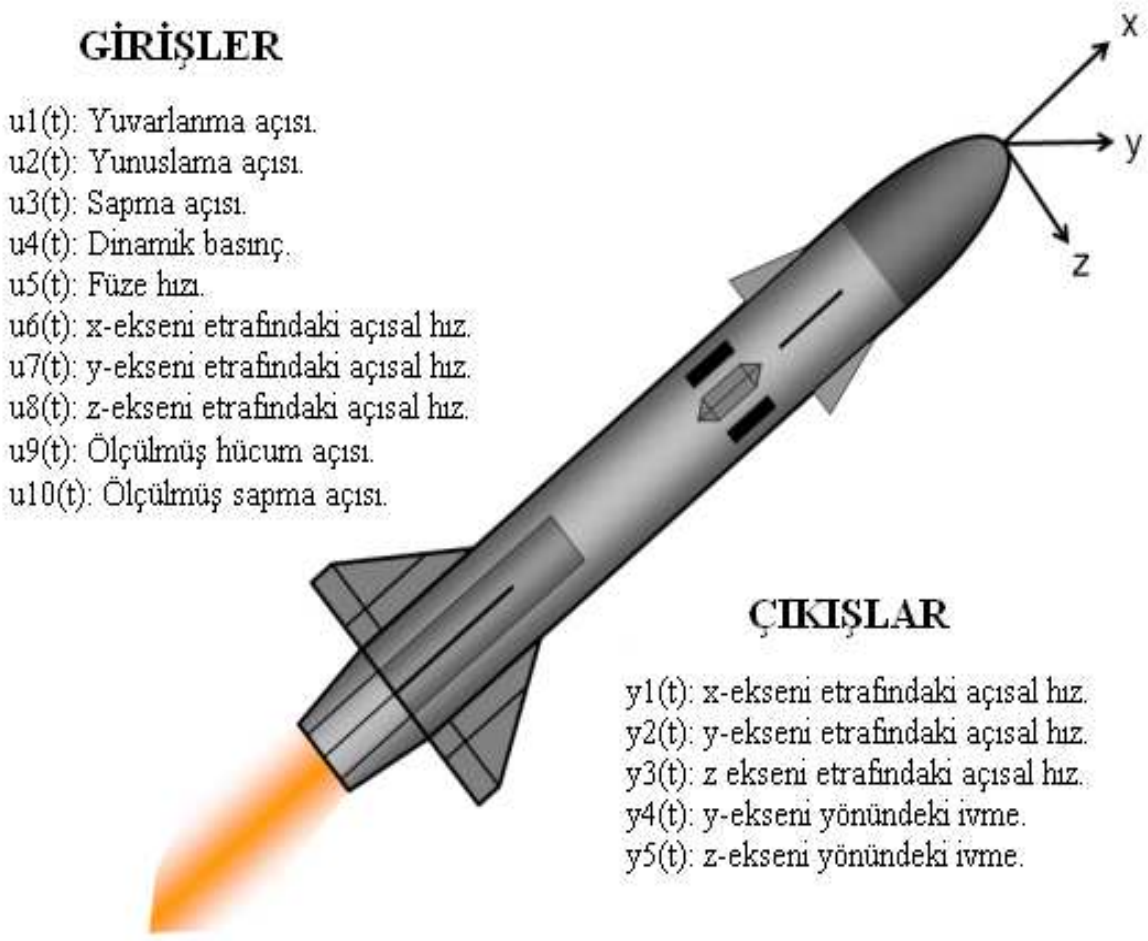
$$\bar{L} = \frac{L}{I_{xx}} \ , \ \bar{M} = \frac{M}{I_{yy}} \ , \ \bar{N} = \frac{N}{I_{zz}}$$

En son olarak altıncı dereceden füze sisteminin durum uzay modelinin son hali aşağıdaki gibi elde edilmektedir.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{I}_{xx} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{I}_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \bar{I}_{zz} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} uq \\ ur \\ vp \\ vr \\ wp \\ wq \\ p^2 \\ pq \\ pr \\ q^2 \\ qr \\ r^2 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \bar{X} + \bar{T} \\ \bar{Y} \\ \bar{Z} \\ \bar{L} \\ \bar{M} \\ \bar{N} \end{bmatrix} + \begin{bmatrix} g_x \\ g_y \\ g_z \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

5. GÜDÜMLÜ FÜZE SİSTEMİNİN PARAMETRE KESTİRİMİ

Bu bölümde çalışmanın amacı olan çok parametreli bir lineer olmayan sistemin parametrelerinin lineer olmayan parametre kestirimi metotlarını kullanarak analizi gerçekleştirilecektir. Parametre kestirimi olarak lineer olmayan Grey-Box ve lineer olmayan ARX modellerin belirtilen sisteme uygulanması yapılarak parametre kestirimi yapılacaktır. Sistem olarak 10 adet girişi 5 adet çıkışı olan MIMO güdümlü füze sistemi incelenecektir. Füze sistemi Saab Bofors Dynamics AB tarafından modellenmiş gerçek bir füzedir. Füze sistemini genel hali aşağıdaki gibidir:



Şekil 5.1: Giriş ve çıkışları belirtilmiş güdümlü füze sistemi

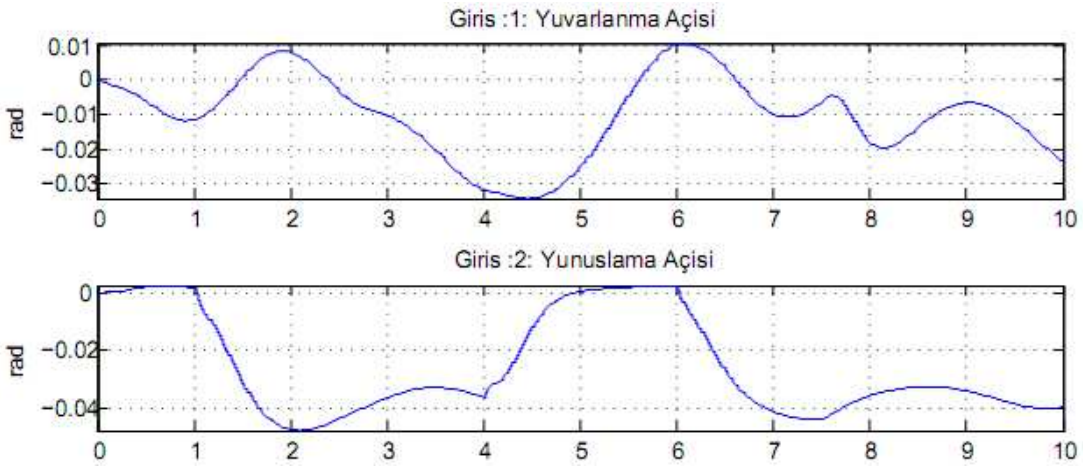
5.1 Grey-Box Modeli Kestirimi

İlk olarak modelin her bir giriş ve çıkış için elde edilmiş 501 adet örnekleme verisi yüklenmektedir. Örnekleme zamanı 0.02 saniyedir. Veriler gerçek füze sisteminin özenle hazırlanmış modelinden elde edilmiştir. Modelin yapısı Newton kanunları kullanılarak kuvvet ve moment denge denklemleri ve temel aerodinamik bağıntılara göre modellenmiştir.

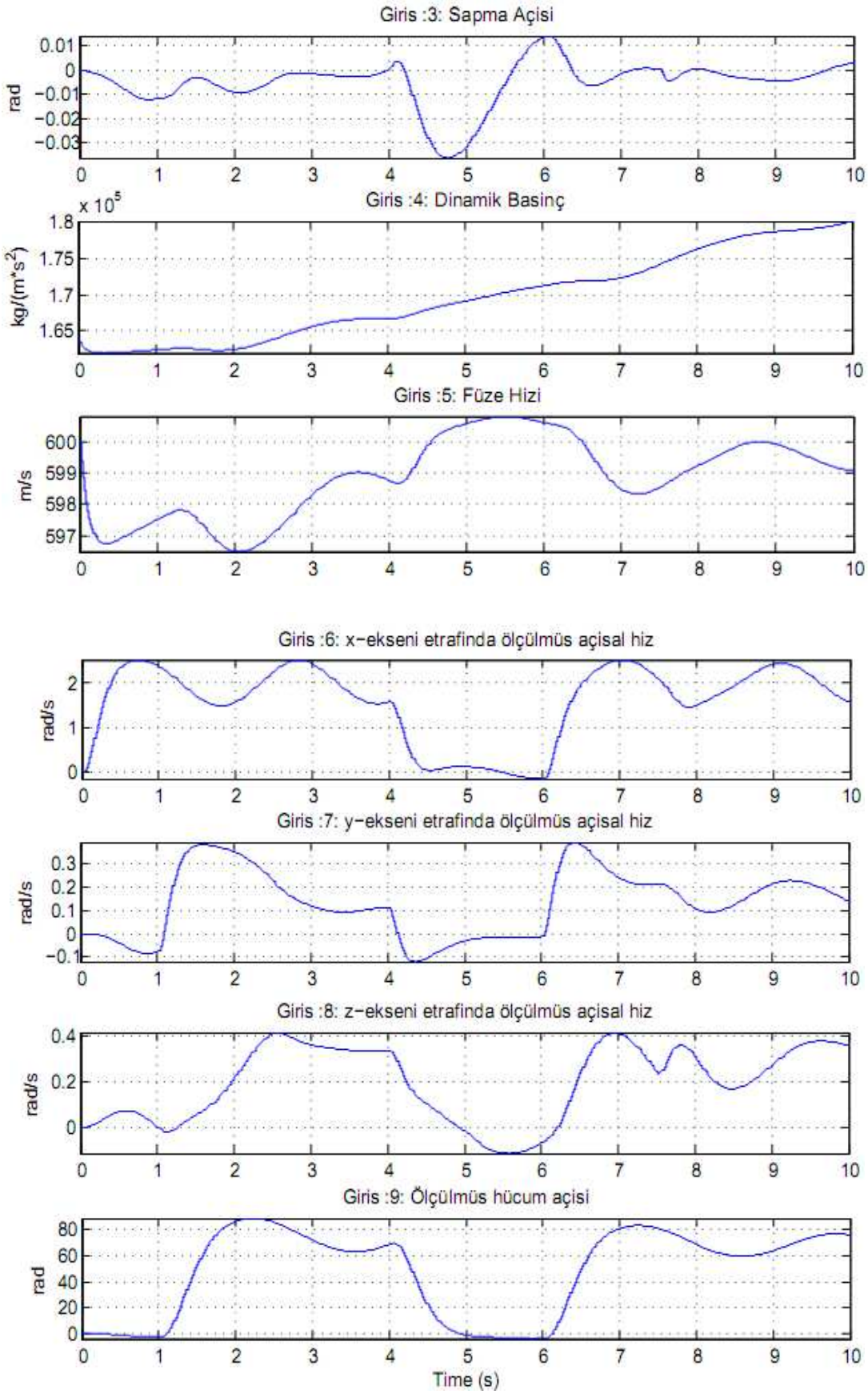
Model 10 girişli 5 çıkışlı ve 5 adet durum içermektedir. Sistem toplam 23 farklı parametre ele alınmaktadır. Bu parametrelerden 7 tanesi sabit değer alınmaktadır, geri kalan 16 tanesi serbesttir ve kestirilecektir. Lineer olmayan Grey-Box model oluşturulmadan önce incelenen parametrelerin ilk değerlerini belirlemek gerekmektedir.

Lineer olmayan grey-box model ile parametre kestirimi yapılırken optimizasyon metodu olarak birçok çeşit metot kullanılabilmektedir. Bu çalışmada Gauss-Newton, Levenberg-Marquardt metodu ve Trust-Region metodu kullanılmıştır. Her bir analiz sonucu değerlendirilmiştir.

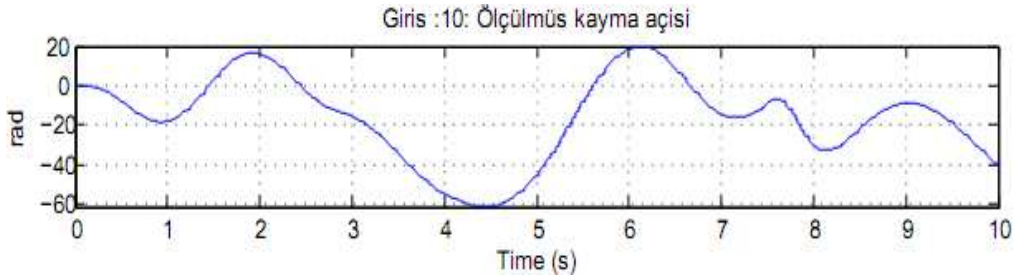
İlk olarak ölçülmüş girişler incelenmiştir. Yukarıdaki şemadan da anlaşılacağı gibi füzenin 10 adet giriş vardır. Bunlar yuvarlanma açısı, yunuslama açısı, sapma açısı, dinamik basınç, füze hızı, x-ekseni etrafındaki açısal hız, y-ekseni etrafındaki açısal hız, z-ekseni etrafındaki açısal hız, hücum açısı ve sapma açısıdır. Bu girişler gerçek güdümlü füze modelinden ölçülmüştür. İlk olarak bu değerlere göre parametre kestirimi yapılacaktır. Aşağıda giriş parametrelerinin ölçülmüş değerlere göre grafikleri verilmiştir.



Şekil 5.2: Sistemin ölçülmüş girişlerinin zamana göre değişimi

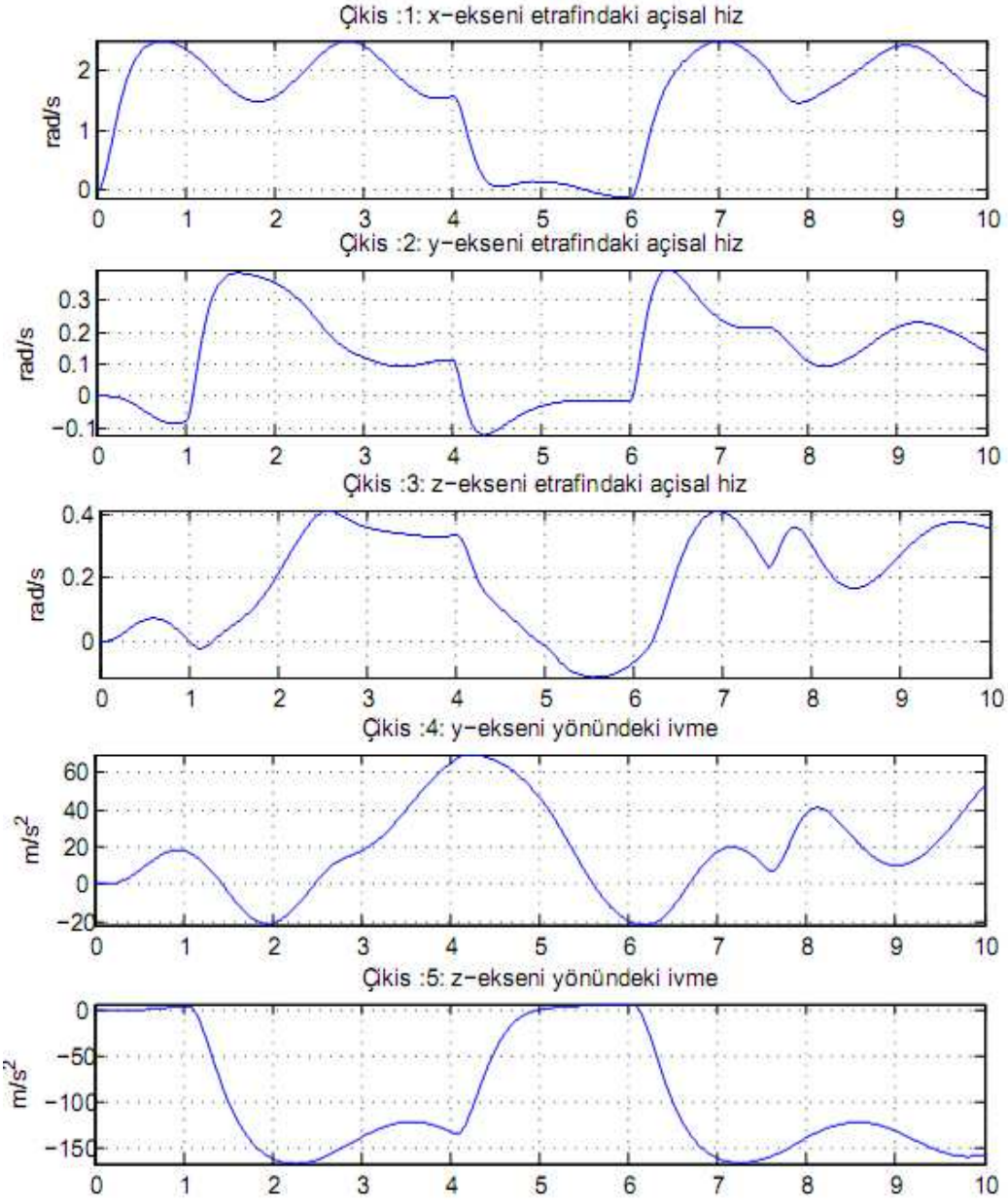


Şekil 5.2: (devam) Sistemin ölçülmüş girişlerinin zamana göre değişimi



Şekil 5.2: (devam) Sistemin ölçülmüş girişlerinin zamana göre değişimi

Aşağıda ölçülmüş değerlere göre sistemin 5 adet çıkış parametrelerinin grafikleri verilmiştir:



Şekil 5.3: Sistemin ölçülmüş çıkışlarının zaman göre değişimi

İlk bakışta bazı giriş ve çıkışların ölçülmüş değerlerin değişimi garip olduğu görülebilir ama kullanılan füze modeli birkaç integratör içermektedir ve buda kararsız simülasyon davranışına sebep olmaktadır. Bundan kaçınmak için ölçümler lineer olmayan gözlemci (observer) yolu ile geri beslenecektir.

Bundan sonraki süreçte parametre kestirimi uygulaması için lineer olmayan Grey-Box modeli tanımlanacaktır ve güdümlü füzenin dinamiğinin modellenmesi anlatılacaktır. Uygun yapı Newton'un kuvvet ve moment denge denklemlerine göre elde edilebilir. Model yapısını tamamlamak için birkaç temel aerodinamik bağıntılarda dikkate alınmıştır. Aşağıda model detaylarını ihmal ederek çıkış denklemleri ve durum denklemleri gösterilmiştir ve program M-file olarak yazılmıştır.

```
function [dx, y] = missile_m(t, x, u, F, M, C, d, A, I, m, K,
varargin)
%MISSILE_M  Lineer olmayan füze sistemi.


% Çıkış denklemleri.
y = [x(1);                                  ... % x- ekseni etrafındaki
açısal hız.
     x(2);                                  ... % y- ekseni etrafındaki
açısal hız.
     x(3);                                  ... % z- ekseni etrafındaki
açısal hız.
     -A*u(4)*(F(1)*x(5)+F(2)*u(3))/m;       ... % y-ekseni yönündeki
ivme.
     -A*u(4)*(F(3)*x(4)+F(4)*u(2))/m        ... % z-ekseni yönündeki
ivme.
    ];

% Durum denklemleri.
dx = [1/I(1)*(d*A*(M(1)*x(5)+0.5*M(2)*d*x(1)/u(5)+       ... % x-
ekseni etrafındaki açısal hız.
           M(3)*u(1))*u(4)-(I(3)-I(2))*x(2)*x(3))+       ...
           K*(u(6)-x(1));                                ...
        1/I(2)*(d*A*(M(4)*x(4)+0.5*M(5)*d*x(2)/u(5)+     ... % y-
ekseni etrafındaki açısal hız.
           M(6)*u(2))*u(4)-(I(1)-I(3))*x(1)*x(3))+       ...
           K*(u(7)-x(2));                                ...
        1/I(3)*(d*A*(M(7)*x(5)+M(8)*x(4)*x(5)+           ... % z-
ekseni etrafındaki açısal hız.
           0.5*M(9)*d*x(3)/u(5)+M(10)*u(1)+              ...
           M(11)*u(3))*u(4)-(I(2)-I(1))*x(1)*x(2))+      ...
           K*(u(8)-x(3));                                ...
        (-A*u(4)*(F(3)*x(4)+F(4)*u(2)))/(m*u(5))-        ... % Hücum
açısı.
           x(1)*x(5)+x(2)+K*(u(9)/u(5)-x(4))+C*x(5)^2;   ...
        (-A*u(4)*(F(1)*x(5)+F(2)*u(3)))/(m*u(5))-        ... % Kayma
açısı.
           x(3)+x(1)*x(4)+K*(u(10)/u(5)-x(5))            ...
       ];
```

Lineer olmayan Grey-Box modelini yaratmadan önce güvendiğimiz ilk giriş parametreleri sağlamamız gerekmektedir. Bu çalışmada 23 farklı parametre sağlanmıştır. Bunlar 4 adet aerodinamik kuvvet katsayıları (F), 11 adet aerodinamik moment katsayıları (M), aerodinamik düzeltme faktörü (C), füze çapı (d), füze referans alanı (A), x-y-z eksenlerindeki atalet momentleri (I), füze kütlesi (m) ve geri besleme kazancı (K) olarak isimlendirebiliriz. İlk parametre değerleri fiziki sonuçlardan elde edilmiştir ve tahminlere dayanarak elde edilmiştir. İlk giriş değerleri aşağıdaki gibi verilmiştir:

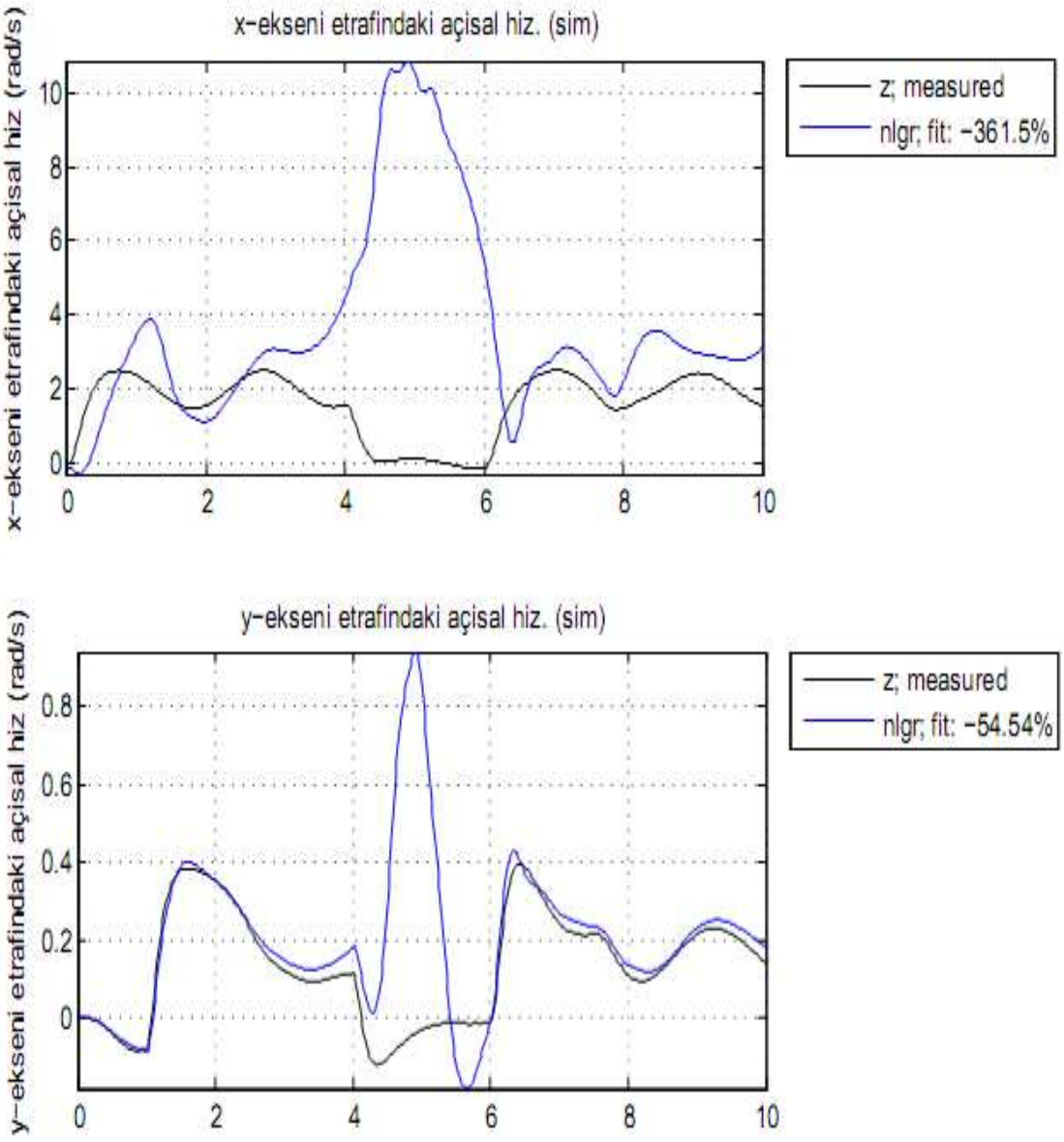
1. Aerodinamik kuvvet katsayıları (F) : [20; -6; 35; 13] $\frac{1}{rad \cdot m^2}$
2. Aerodinamik moment katsayıları (M) : [-1; 15; 3 -16; -1800; -50; 23; -200; -2000; -17; -50] $\frac{1}{rad \cdot s}$
3. Aerodinamik düzeltme faktörü (C) : -5 $\frac{1}{rad \cdot s}$
4. Füze çapı (d) : 0.17 m
5. Füze referans alanı (A) : 0.0227 m^2
6. x-y-z eksenlerindeki atalet momentleri (I) : [0.5; 110; 110] $kg \cdot m^2$
7. Füze kütlesi (m) : 107 kg
8. Geri besleme kazancı (K) : 6

Ayrıca füze sisteminin aynı durumda 5 adet durumunu tanımlamamız gerekir:

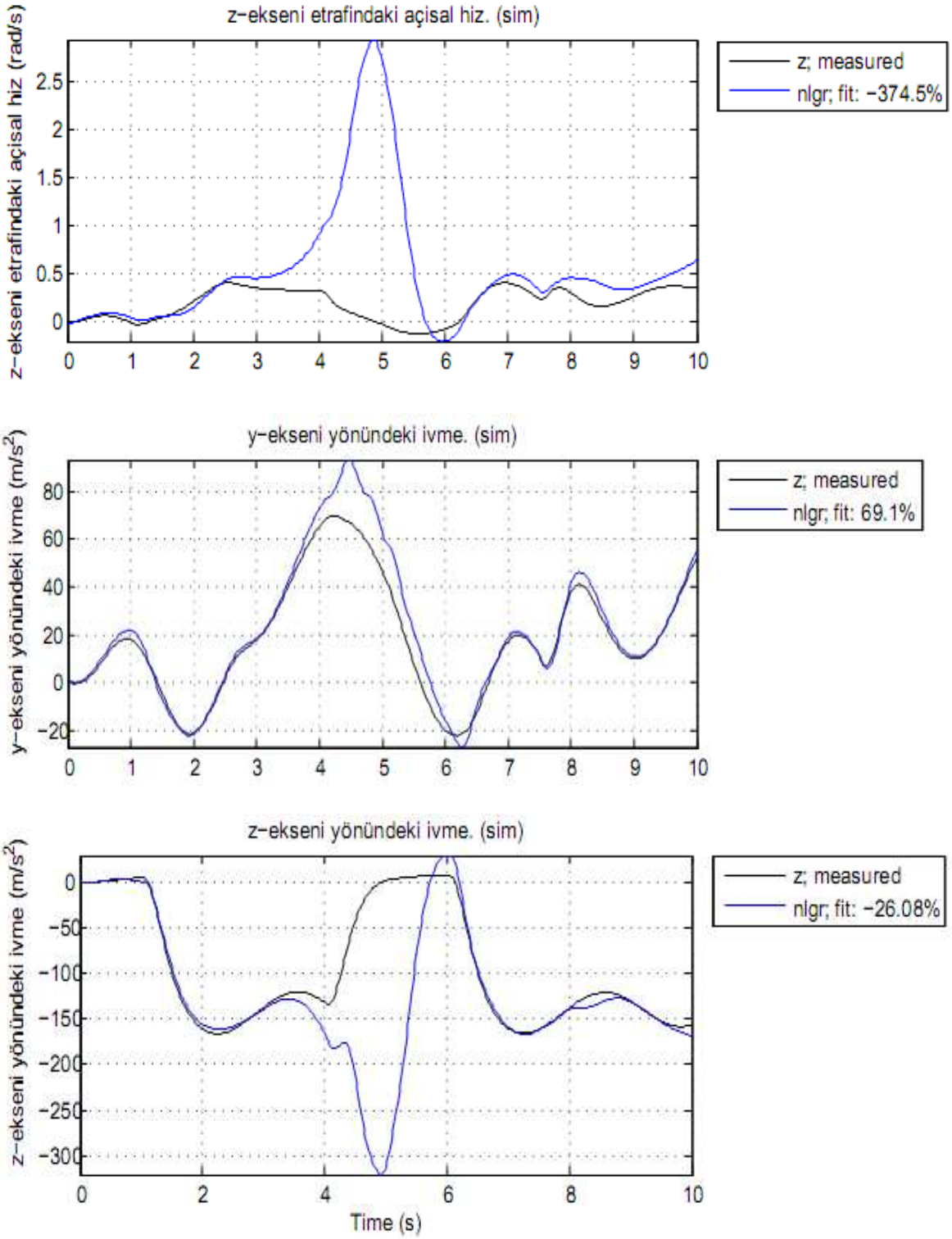
1. x-ekseni etrafındaki açısal hız: 0 $\frac{rad}{s}$
2. y-ekseni etrafındaki açısal hız: 0 $\frac{rad}{s}$
3. z-ekseni etrafındaki açısal hız: 0 $\frac{rad}{s}$
4. Hücum açısı: 0 rad
5. Kayma açısı: 0 rad

Parametre ve ilk durum verilerini kullanarak grey-box modelin yapısı tanımlanmıştır. Füze sisteminin giriş ve çıkış sinyalleri, IDDATA modelinden uygun bir şekilde kopyalanarak sonraki belirlemeler yapılmıştır. 10 giriş sinyali, 5 durum ve 5 çıkış sinyalleri içeren lineer olmayan grey-box modeli oluşturulmuştur. Önceden belirlendiği gibi model ayrıca 23 parametre içermektedir. Bunlardan 7 tanesi sabit, 16 tanesi serbesttir.

16 serbest parametreyi kestirmeden önce sistemin ilk parametre vektörünü kullanarak simülasyon yapılmıştır. Bu şekilde ilk modelin kalitesini anlamak adına yararlı bilgiler elde edilmiştir. Aşağıda ilk parametre vektörüne göre grafikler verilmiştir:

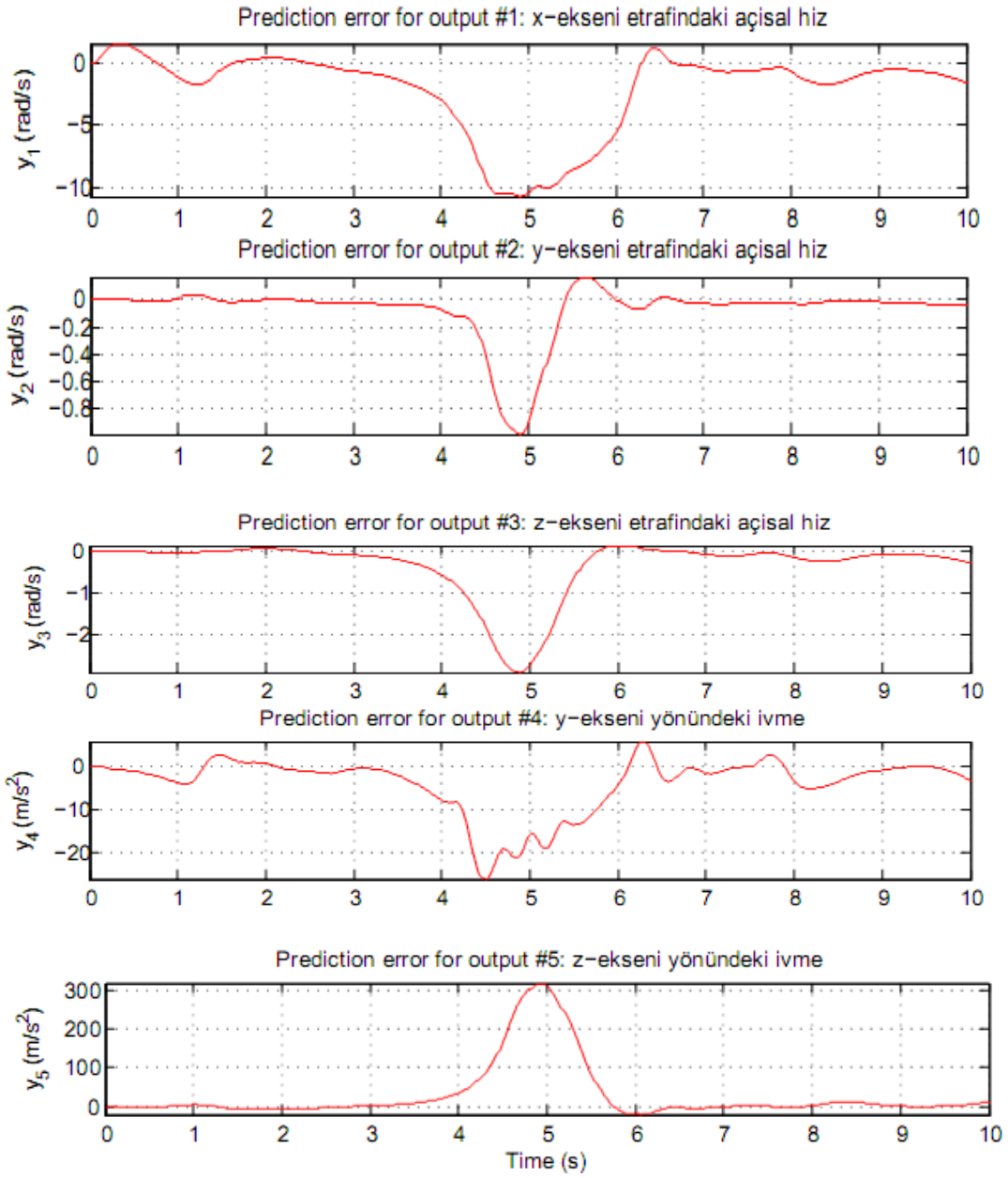


Şekil 5.4: İlk parametrelere göre ölçülmüş model ile kestirim karşılaştırması



Şekil 5.4: (devam) İlk parametrelere göre ölçülmüş model ile kestirim karşılaştırması

İlk güdümlü füze modelinin ölçülmüş çıkış ve simülasyon edilmiş çıkışlarının karşılaştırmasında 4 – 6 saniyeleri dışında gerçek ve simülasyon sinyallerinin grafikten kötü olmadığını görebiliyoruz. Bunu aşağıdaki hata tahmini grafiğinden de açık bir şekilde görebiliriz:



Şekil 5.5: İlk güdümlü füze modeli hatalarının tahmini

Yukarıdaki incelemelerden görüldüğü gibi bu ilk modelin parametre kestirimi için başlama noktası olarak ele almak akla yatkındır. Parametre kestirimi için pem komutu altında ilk olarak Trust-Region, ikinci olarak Gauss-Newton ve son olarak Levenberg-Marquardt optimizasyon metotları kullanarak model incelenecektir.

5.1.1 Trust-Region metodu

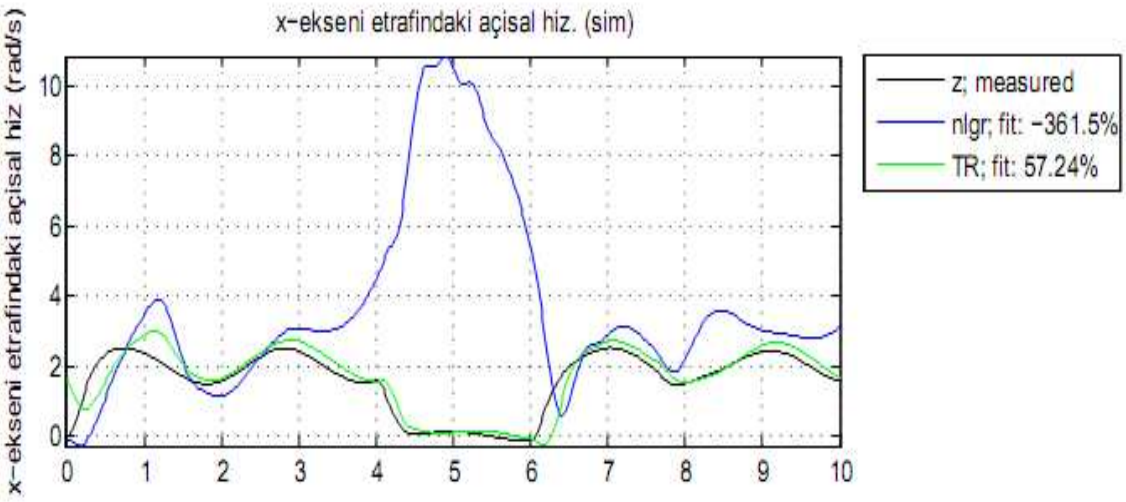
16 serbest parametrenin kestirimi için ilk olarak Trust-Region optimizasyon metodu kullanılmıştır. Kullanılan bilgisayarda kestirim zamanı 48.55 saniye ve iterasyon birim zamanı: 2.427 saniye olarak hesaplanmıştır.

Trust-Region iterasyonu aşağıda çizelge halinde verilmiştir.

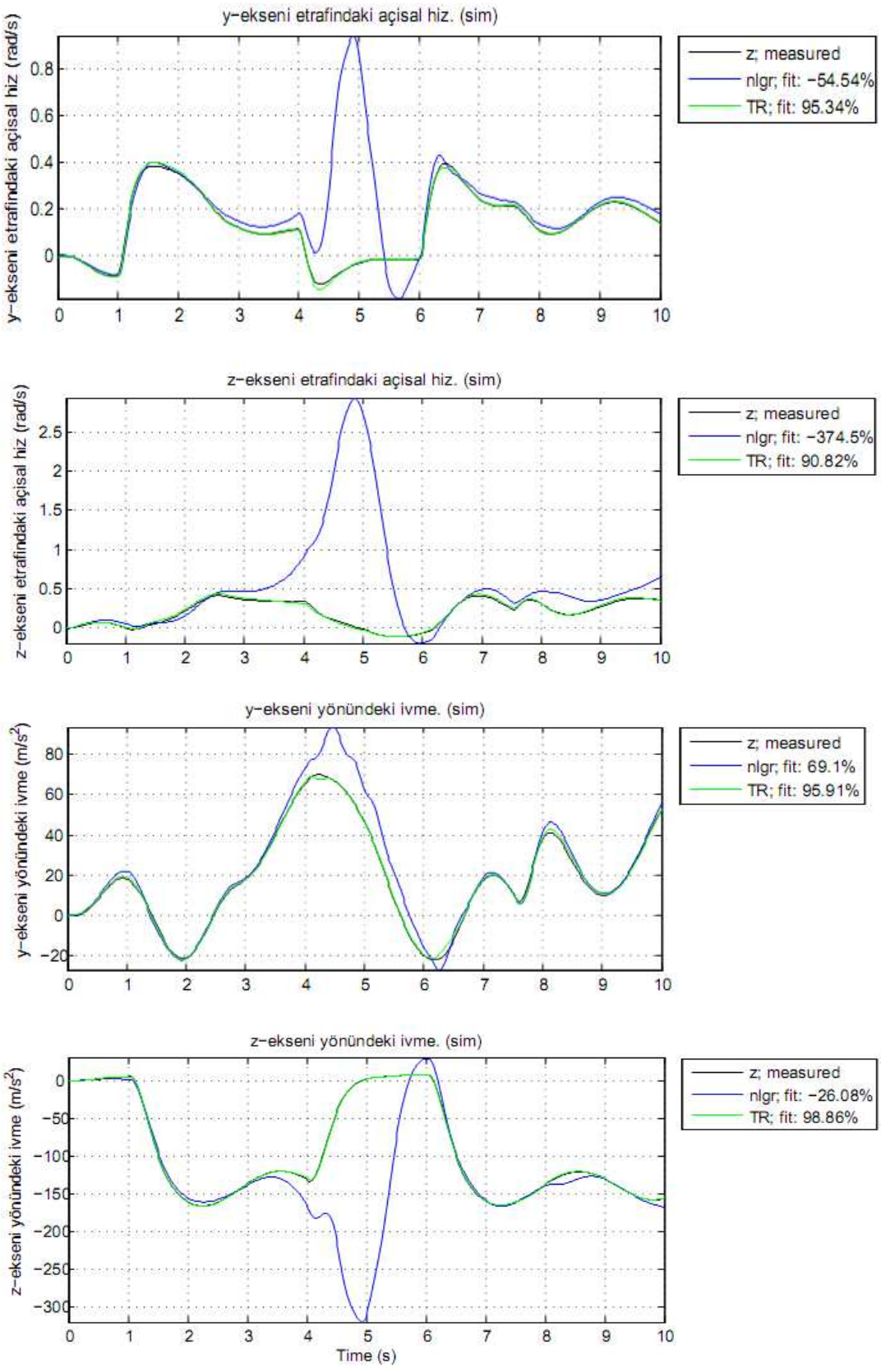
Çizelge 5.1: Trust-Region iterasyon sonuçları

Iterasyon	Amaç Fonksiyonu	Norm Adımları	Gradiyent
0	1389.23	-	-
1	576.647	10	4.41e+005
2	409.863	10	3.82e+004
3	369.655	10	9.3e+003
4	345.151	20	1.38e+004
5	323.765	40	1.08e+004
6	294.091	80	1.48e+004
7	241.874	160	1.98e+004
8	100.93	320	2.25e+005
9	34.6073	640	9.26e+005
10	7.3802	1.28e+003	1.14e+005
11	7.3802	1.52e+003	1.14e+005
12	4.04828	381	6.95e+004
13	1.26636	761	5.16e+004
14	0.375938	301	6.27e+003
15	0.356682	431	3.59e+003
16	0.352838	105	441
17	0.352269	71.5	256
18	0.352251	37.4	166
19	0.352214	9.3	45
20	0.352212	4.88	24.5

Trust-Region optimizasyon metodunu kullanarak kestirilmiş modelin doğruluğunu değerlendirmek ve geliştirmek için ilk model ve kestirilmiş modelin simülasyonunu karşılaştıracağız. Ölçülmüş çıkışlar ve simülasyonu yapılmış çıkışların karşılaştırması aşağıdaki grafiklerde verilmiştir.

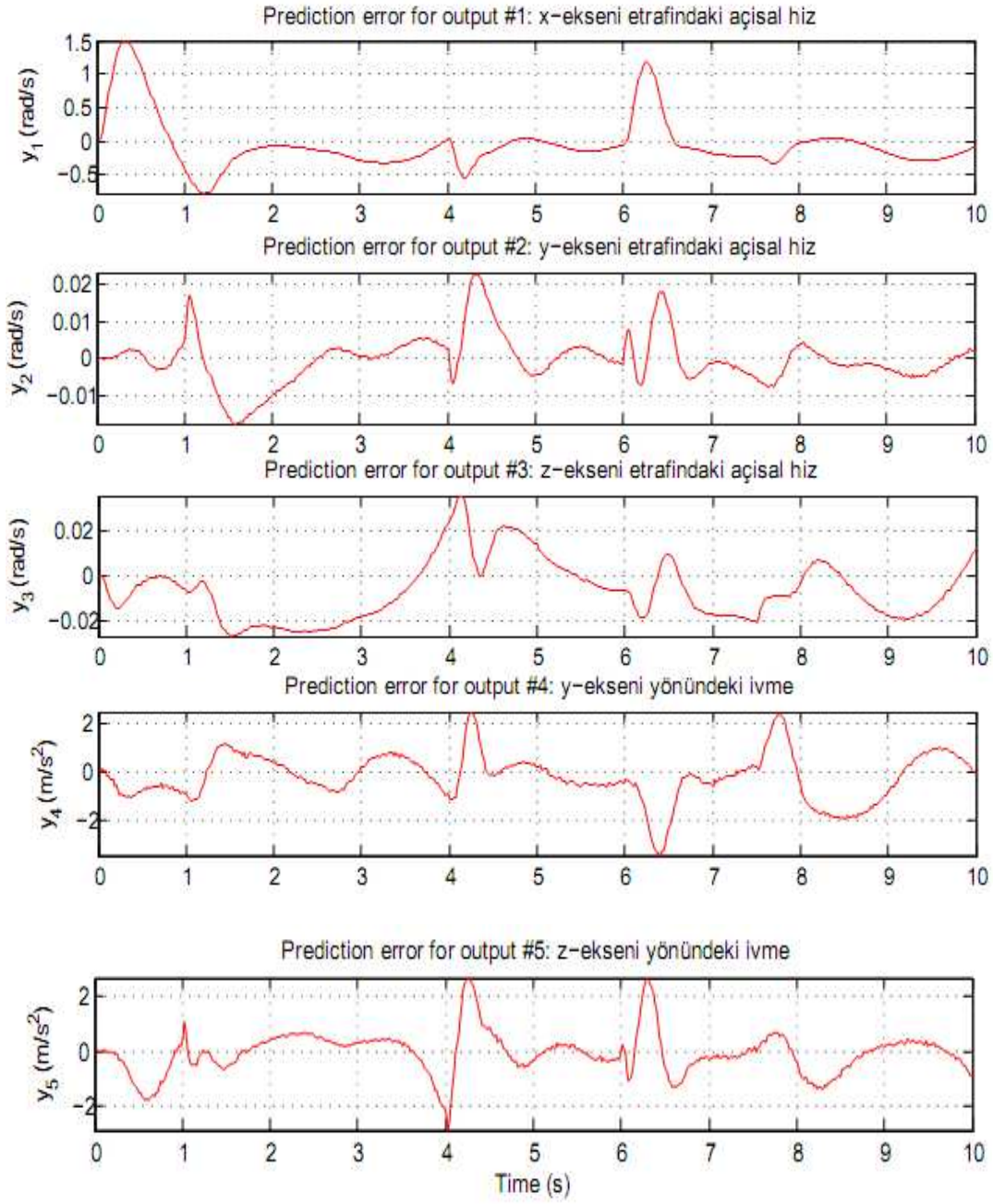


Şekil 5.6: Ölçülmüş ve Trust –Region simülasyon çıkışlarının karşılaştırılması



Şekil 5.6: (devam) Ölçülmüş ve Trust –Region simülasyon çıkışlarının karşılaştırılması

Bu grafik ilk model ile elde edilmiş simülasyon sonuçlarının gelişimini göstermektedir. Burada ilk modele göre 4 – 6 saniyeler arasındaki bölgenin daha iyi kesinlik kazandığını görmekteyiz. Aşağıda Trust-Region metodunun uygulandığı kestirim için hata tahmin grafiği verilmiştir:



Şekil 5.7: Trust-Region metodu ile kestirilmiş güdümlü füzenin tahmin hataları

Aşağıda bu çalışmanın, kestirilmiş füze modelinin bazı detayları çizelge halinde verilmiştir.

Çizelge 5.2: Trust-Region kestirim sonuçları

Parametreler	Parametre İsimi	Değer	Standart Sapma	Durum
p1(1)	Aerodinamik kuvvet katsayısı [1/(rad*m^2]	21,3246	0,348672	kestirilmiş
p1(2)		-7,78593	0,186189	kestirilmiş
p1(3)		36,2971	0,752835	kestirilmiş
p1(4)		13,0201	1,41627	kestirilmiş
p2(1)	Aerodinamik moment katsayısı [1/rad]	-1,03446	0,075018	kestirilmiş
p2(2)		15,337	0,882165	kestirilmiş
p2(3)		2,96768	0,202111	kestirilmiş
p2(4)		-18,7409	0,356223	kestirilmiş
p2(5)		-1160,03	234,007	kestirilmiş
p2(6)		-57,0413	1,36136	kestirilmiş
p2(7)		34,2341	1,45059	kestirilmiş
p2(8)		-208,39	8,23738	kestirilmiş
p2(9)		-2480,44	278,64	kestirilmiş
p2(10)		-32,5455	3,165	kestirilmiş
p2(11)		-50,9551	1,6996	kestirilmiş
p3	Aerodinamik düzeltme faktörü [1/(s*rad)]	-0,652	0,715564	kestirilmiş
p4	Füze çapı [m]	0,17	0	sabit
p5	Füze referans alanı [m^2]	0,0227	0	sabit
p6(1)	x-y-z yönündeki atalet moentleri [kg*m^2]	0,5	0	sabit
p6(2)		110	0	sabit
p6(3)		110	0	sabit
p7	Füzenin kütlesi [kg]	107	0	sabit
p8	Geribesleme kazancı [-]	6	0	sabit

5.1.2 Gauss-Newton metodu

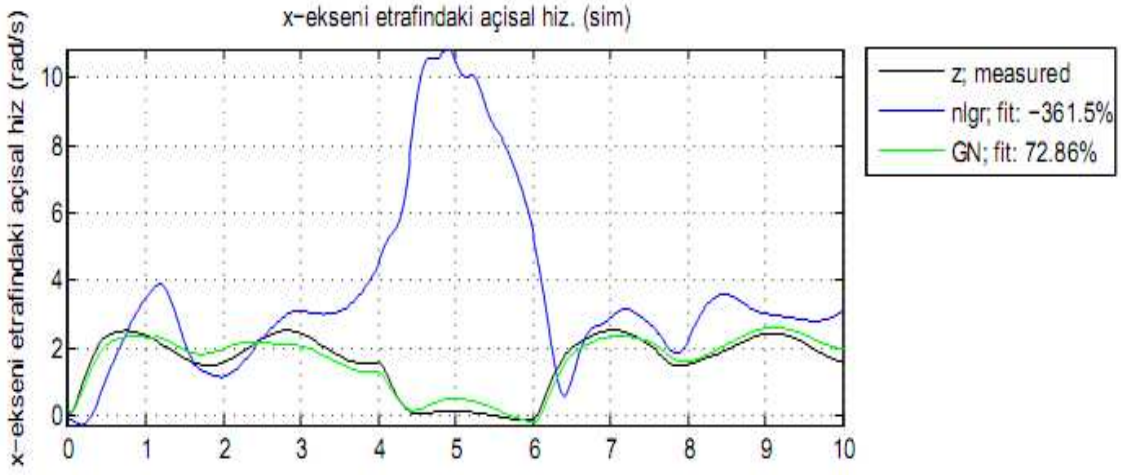
16 serbest parametrenin kestirimi için ikinci olarak Gauss-Newton optimizasyon metodu kullanılmıştır. Kullanılan bilgisayarda kestirim zamanı 75.11 saniye ve iterasyon birim zamanı: 3.756 saniye olarak hesaplanmıştır.

Gauss-Newton iterasyon sonuçları aşağıdaki çizelgede verilmiştir.

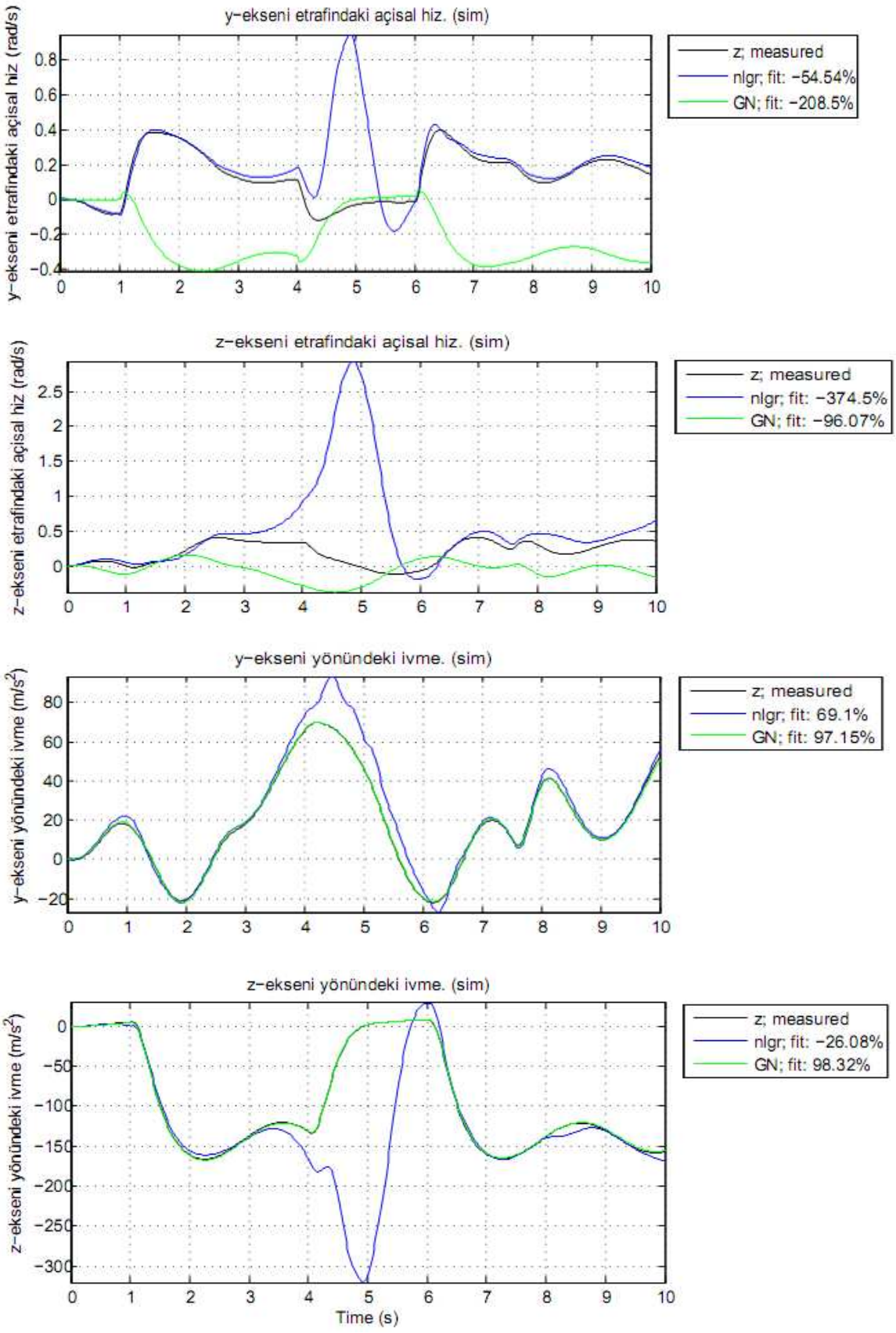
Çizelge 5.3: Gauss-Newton iterasyon sonuçları

Iterasyon	Amaç Fonksiyonu	Norm Adımları	Gradiyent
0	6946.14	-	4.68e+006
1	1556.72	3.09e+003	1.39e+005
2	334.447	1.65e+004	2.75e+004
3	141.76	1.49e+004	7.22e+003
4	90.5417	1.45e+004	5.16e+004
5	49.7591	1.4e+004	2.8e+004
6	23.6538	2.21e+004	4.42e+003
7	23.0437	1.35e+004	2.62e+004
8	17.4386	2.16e+004	8.64e+003
9	5.76322	1.63e+004	5.09e+003
10	3.10637	607	257
11	2.45518	9.03e+003	262
12	2.44717	2.83e+003	460
13	2.43933	3.18e+003	642
14	2.42144	3.39e+003	782
15	2.38667	3.41e+003	860
16	2.32241	6.45e+003	1.11e+003
17	2.1781	1.07e+004	1.07e+003
18	2.16605	5.26e+003	278
19	2.10943	2.75e+003	93.6
20	2.06139	3.83e+003	248

Gauss-Newton optimizasyon metodunu kullanarak kestirilmiş modelin doğruluğunu değerlendirmek ve geliştirmek için ilk model ve kestirilmiş modelin simülasyonunu karşılaştıracağız. Ölçülmüş çıkışlar ve simülasyonu yapılmış çıkışların karşılaştırması aşağıdaki grafiklerde verilmiştir.

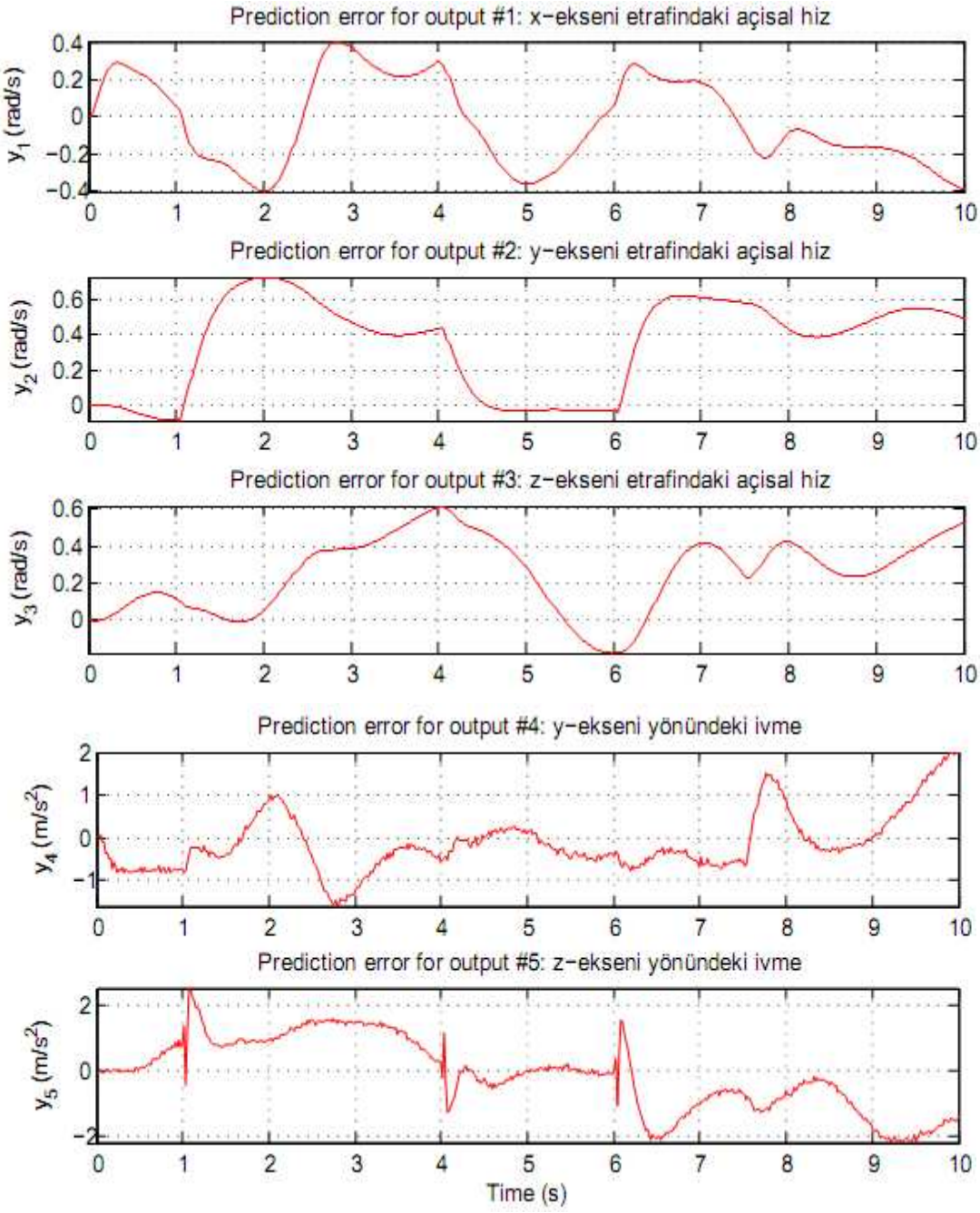


Şekil 5.8: Ölçülmüş ve Gauss-Newton simülasyon çıkışlarının karşılaştırılması



Şekil 5.8: (devam) Ölçülmüş ve Gauss-Newton simülasyon çıkışlarının karşılaştırılması

Bu grafiklerden anlaşıldığı gibi y – ekseni etrafındaki açısal hız dışında diğer 4 çıkışta ilk model ile elde edilmiş simülasyon sonuçlarına göre gelişim gösterilmiştir. Burada ilk modele göre 4 – 6 saniyeler arasındaki bölgenin daha iyi kesinlik kazandığını görmekteyiz. Aşağıda Gauss-Newton metodunun uygulandığı kestirim için hata tahmin grafiği verilmiştir:



Şekil 5.9: Gauss-Newton metodu ile kestirilmiş güdümlü füzenin tahmin hataları

Aşağıda bu çalışmanın, kestirilmiş füze modelinin bazı detayları çizelge halinde verilmiştir.

Çizelge 5.4: Gauss-Newton kestirim sonuçları

Parametreler	Parametre İsimi	Değer	Standart Sapma	Durum
p1(1)	Aerodinamik kuvvet katsayısı [1/(rad*m^2]	94,3587	3,47017	kestirilmiş
p1(2)		-5,35047	0,282662	kestirilmiş
p1(3)		603,048	28,493	kestirilmiş
p1(4)		321,856	22,1434	kestirilmiş
p2(1)	Aerodinamik moment katsayısı [1/rad]	8,27496	0,949733	kestirilmiş
p2(2)		-22,9249	2,31851	kestirilmiş
p2(3)		-5,20616	0,497653	kestirilmiş
p2(4)		-1162,17	103,517	kestirilmiş
p2(5)		-100647	5917,18	kestirilmiş
p2(6)		-678,754	60,7764	kestirilmiş
p2(7)		219,917	16,6954	kestirilmiş
p2(8)		-902,37	149,551	kestirilmiş
p2(9)		-45729,7	1949,76	kestirilmiş
p2(10)		-42,3662	5,514	kestirilmiş
p2(11)		-4,8358	1,6178	kestirilmiş
p3	Aerodinamik düzeltme faktörü [1/(s*rad)]	-26,28	3,17578	kestirilmiş
p4	Füze çapı [m]	0,17	0	sabit
p5	Füze referans alanı [m^2]	0,0227	0	sabit
p6(1)	x-y-z yönündeki atalet moentleri [kg*m^2]	0,5	0	sabit
p6(2)		110	0	sabit
p6(3)		110	0	sabit
p7	Füzenin kütlesi [kg]	107	0	sabit
p8	Geribesleme kazancı [-]	6	0	sabit

5.1.3 Levenberg-Marquardt metodu

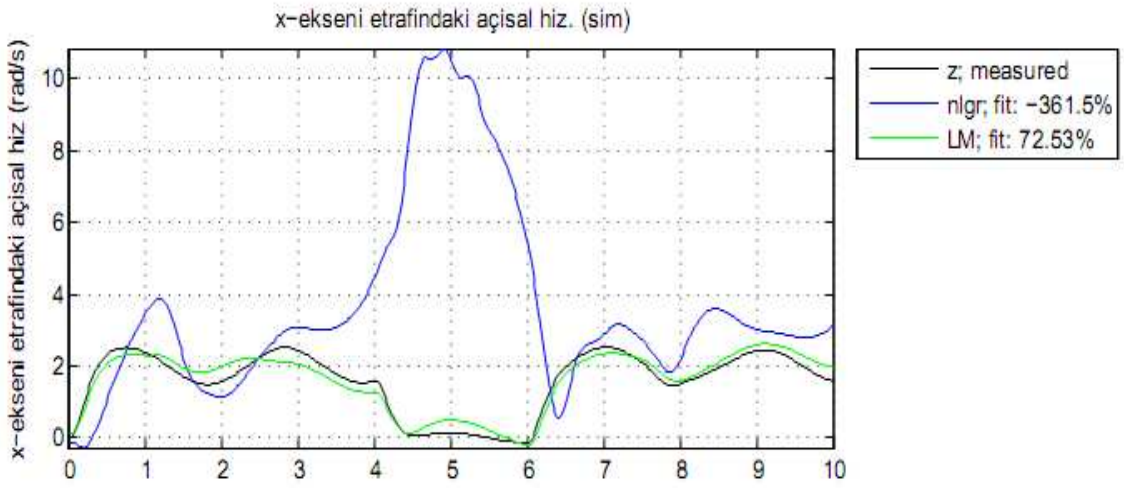
16 serbest parametrenin kestirimi için ikinci olarak Levenberg-Marquardt optimizasyon metodu kullanılmıştır. Kullanılan bilgisayarda kestirim zamanı 82.19 saniye ve iterasyon birim zamanı: 4.109 saniye olarak hesaplanmıştır.

Levenberg-Marquardt iterasyonu aşağıdaki çizelgede gösterilmiştir:

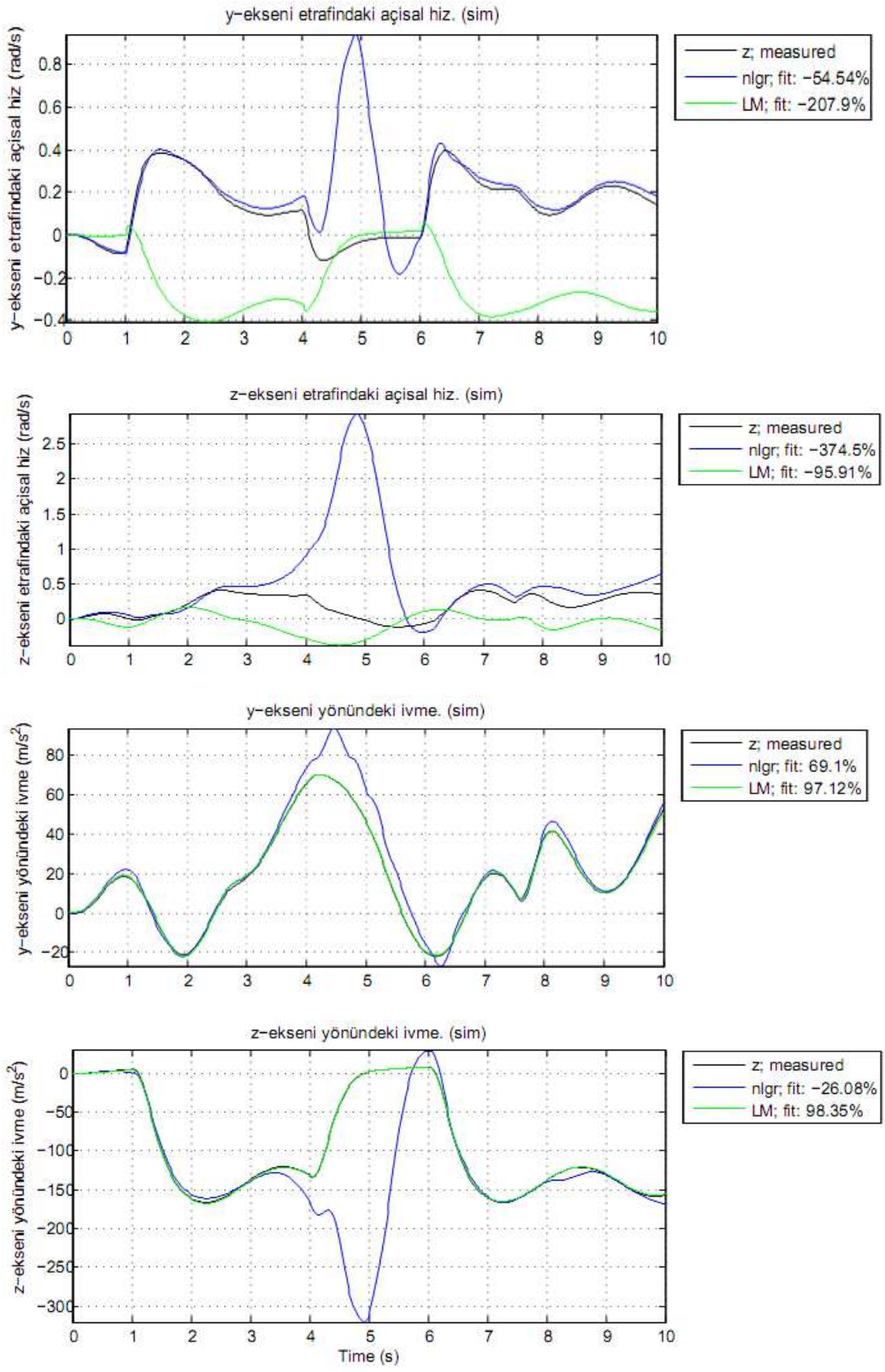
Çizelge 5.5: Levenberg-Marquardt iterasyon sonuçları

Iterasyon	Amaç Fonksiyonu	Norm Adımları	Gradiyent
0	6946.14	-	4.68e+006
1	1556.72	3.09e+003	1.39e+005
2	334.447	1.65e+004	2.75e+004
3	141.76	1.49e+004	7.22e+003
4	90.5417	1.45e+004	5.16e+004
5	49.7591	1.4e+004	2.8e+004
6	23.6538	2.21e+004	4.42e+003
7	16.2553	100	1.02e+004
8	14.1526	2.43e+004	8.77e+003
9	5.41259	762	8.1e+003
10	2.60431	602	1.86e+003
11	2.26934	1.13e+003	476
12	2.13932	3.24e+003	127
13	2.12462	2e+004	705
14	2.07145	260	367
15	2.03835	710	9.7
16	2.03782	1.47e+003	88.5
17	2.0318	5.4e+003	121
18	2.02809	1.28e+003	29.9
19	2.02736	370	1.54
20	2.02732	113	0.0564

Levenberg-Marquardt optimizasyon metodunu kullanarak kestirilmiş modelin doğruluğunu değerlendirmek ve geliştirmek için ilk model ve kestirilmiş modelin simülasyonunu karşılaştıracağız. Ölçülmüş çıkışlar ve simülasyonu yapılmış çıkışların karşılaştırması aşağıdaki grafiklerde verilmiştir.

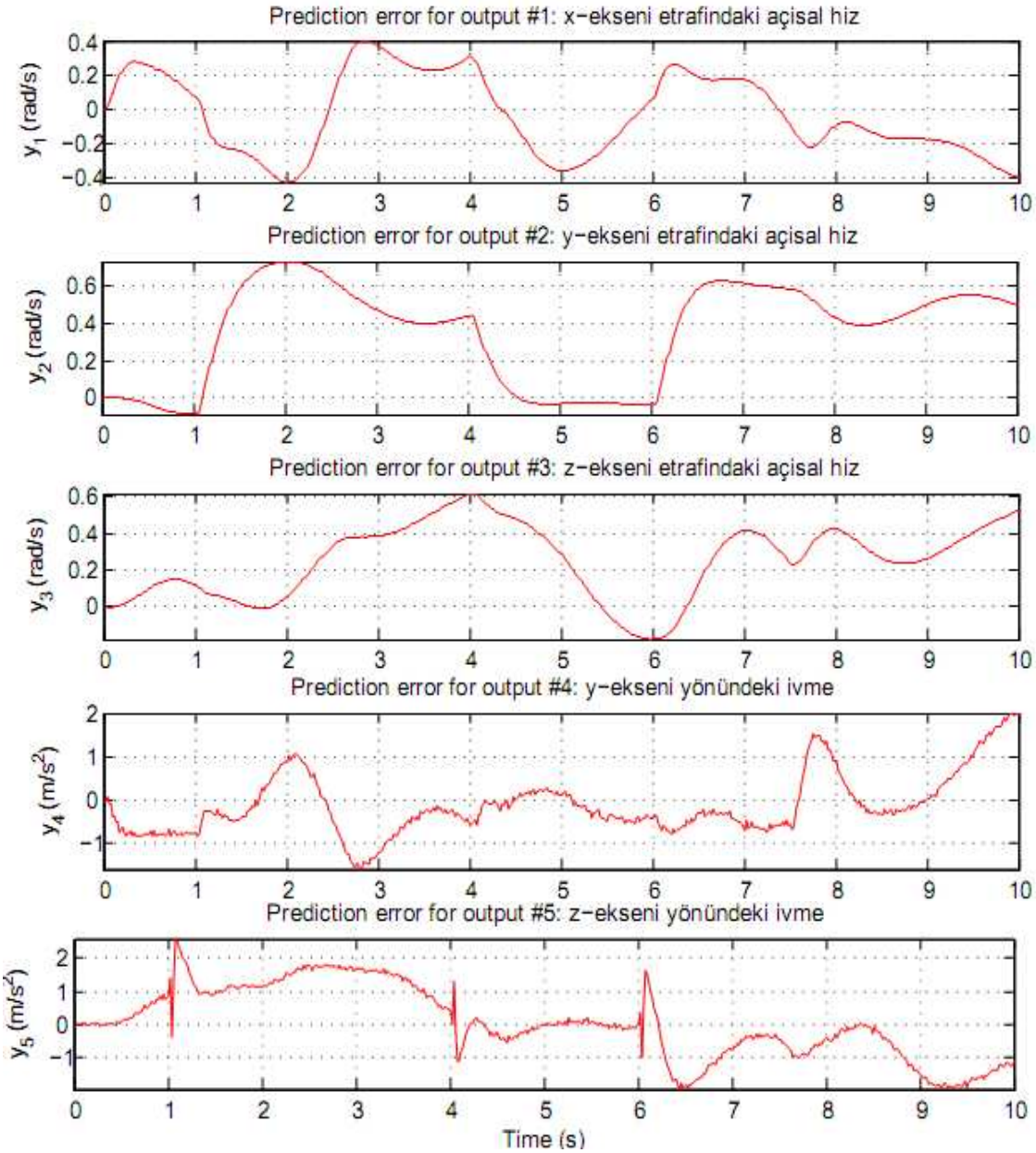


Şekil 5.10: Ölçülmüş ve Levenberg-Marquardt simülasyon çıkışları



Şekil 5.10: (devam) Ölçülmüş ve Levenberg-Marquardt simülasyon çıkışları

Bu grafiklerden anlaşıldığı gibi Gauss-Newton metodunda olduğu gibi bir tek y-ekseni etrafındaki açısal hız çıkışının kestirim sonucu kötüye gitmiştir. Diğer 4 çıkış parametreleri ilk model ile elde edilmiş simülasyon sonuçlarına göre gelişime göstermiştir. Burada ilk modele göre 4 – 6 saniyeler arasındaki bölgenin daha iyi kesinlik kazandığını görmekteyiz. Aşağıda Levenberg-Marquardt metodunun uygulandığı kestirim için hata tahmin grafiği verilmiştir:



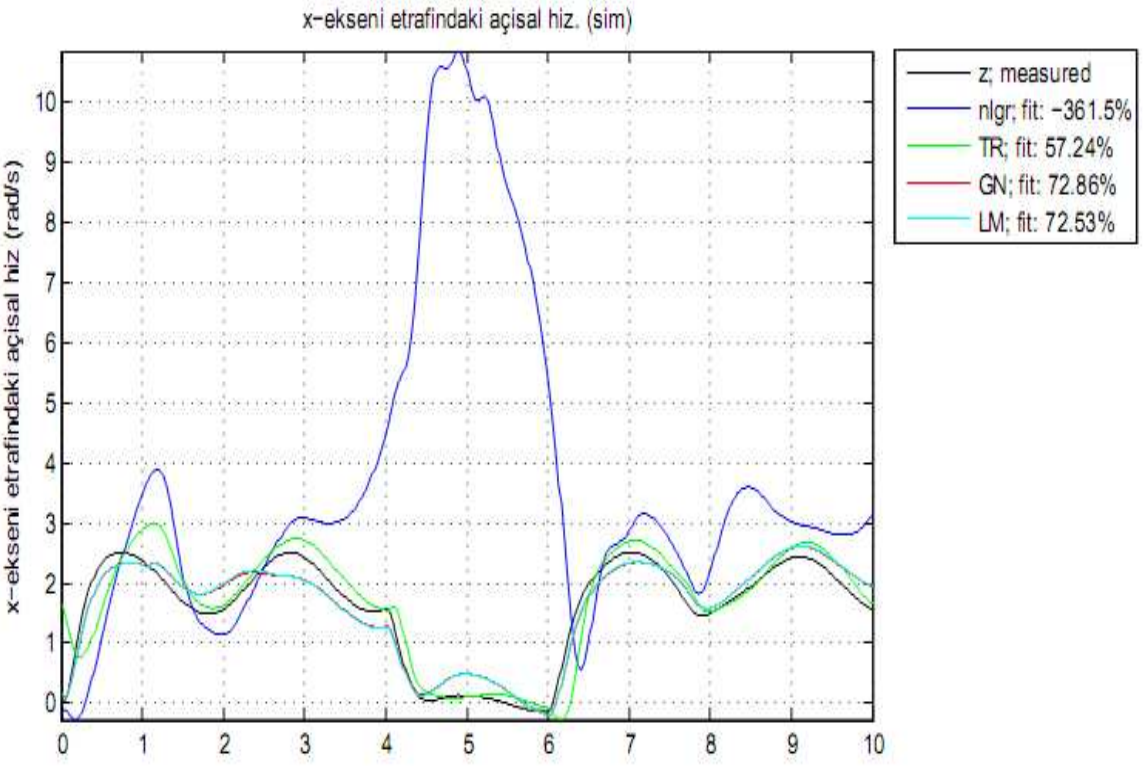
Şekil 5.11: Levenberg-Marquardt metodu ile kestirilmiş güdümlü füzenin tahmin hataları

Aşağıda bu çalışmanın, kestirilmiş füze modelinin bazı detayları tablo halinde verilmiştir:

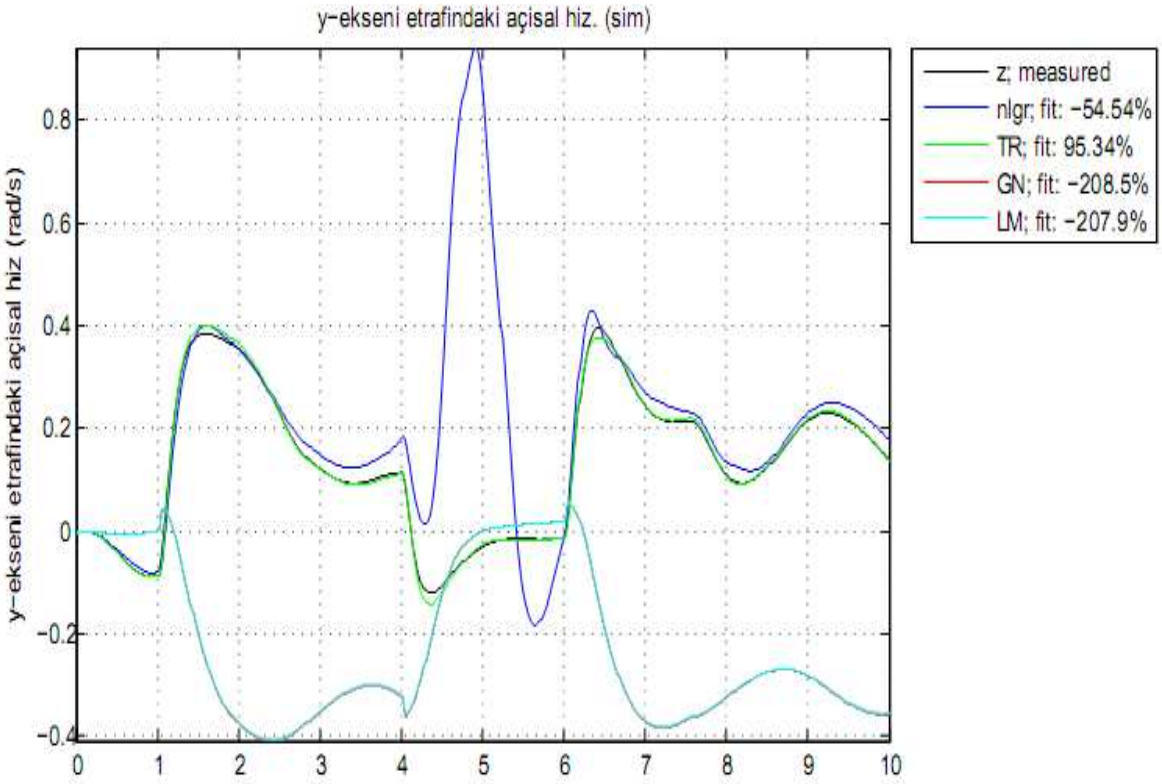
Çizelge 5.6: Levenberg-Marquardt kestirim sonuçları

Parametreler	Parametre İsimi	Değer	Standart Sapma	Durum
p1(1)	Aerodinamik kuvvet katsayısı [1/(rad*m^2]	94,1382	3,38518	kestirilmiş
p1(2)		-5,38404	0,274498	kestirilmiş
p1(3)		606,373	27,962	kestirilmiş
p1(4)		325,587	21,9272	kestirilmiş
p2(1)	Aerodinamik moment katsayısı [1/rad]	8,54838	0,959443	kestirilmiş
p2(2)		-23,6132	2,33594	kestirilmiş
p2(3)		-5,38960	0,502491	kestirilmiş
p2(4)		-1174,12	102,434	kestirilmiş
p2(5)		-101100	5841,190	kestirilmiş
p2(6)		-689,995	60,5826	kestirilmiş
p2(7)		221,867	16,5754	kestirilmiş
p2(8)		-877,85	147,609	kestirilmiş
p2(9)		-45668,6	1920,30	kestirilmiş
p2(10)		-44,3638	5,596	kestirilmiş
p2(11)		-4,6519	1,6097	kestirilmiş
p3	Aerodinamik düzeltme faktörü [1/(s*rad)]	-26,2525	3,11871	kestirilmiş
p4	Füze çapı [m]	0,17	0	sabit
p5	Füze referans alanı [m^2]	0,0227	0	sabit
p6(1)	x-y-z yönündeki atalet moentleri [kg*m^2]	0,50	0	sabit
p6(2)		110	0	sabit
p6(3)		110	0	sabit
p7	Füzenin kütlesi [kg]	107	0	sabit
p8	Geribesleme kazancı [-]	6	0	sabit

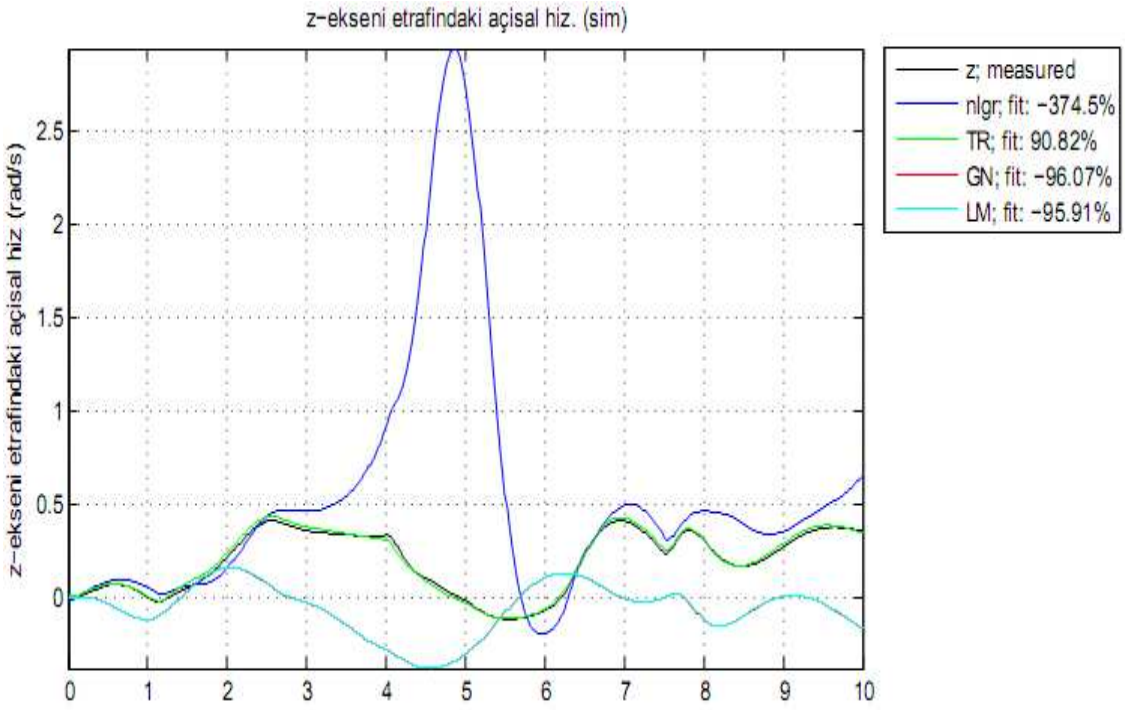
Yukarda sonuçları verilen üç optimizasyon metoduna göre yapılan lineer olmayan grey-box parametre kestirimi modelinin aşağıda karşılaştırması verişmiştir. Her bir optimizasyon metodunun sonuçları grafiklerde görülmektedir.



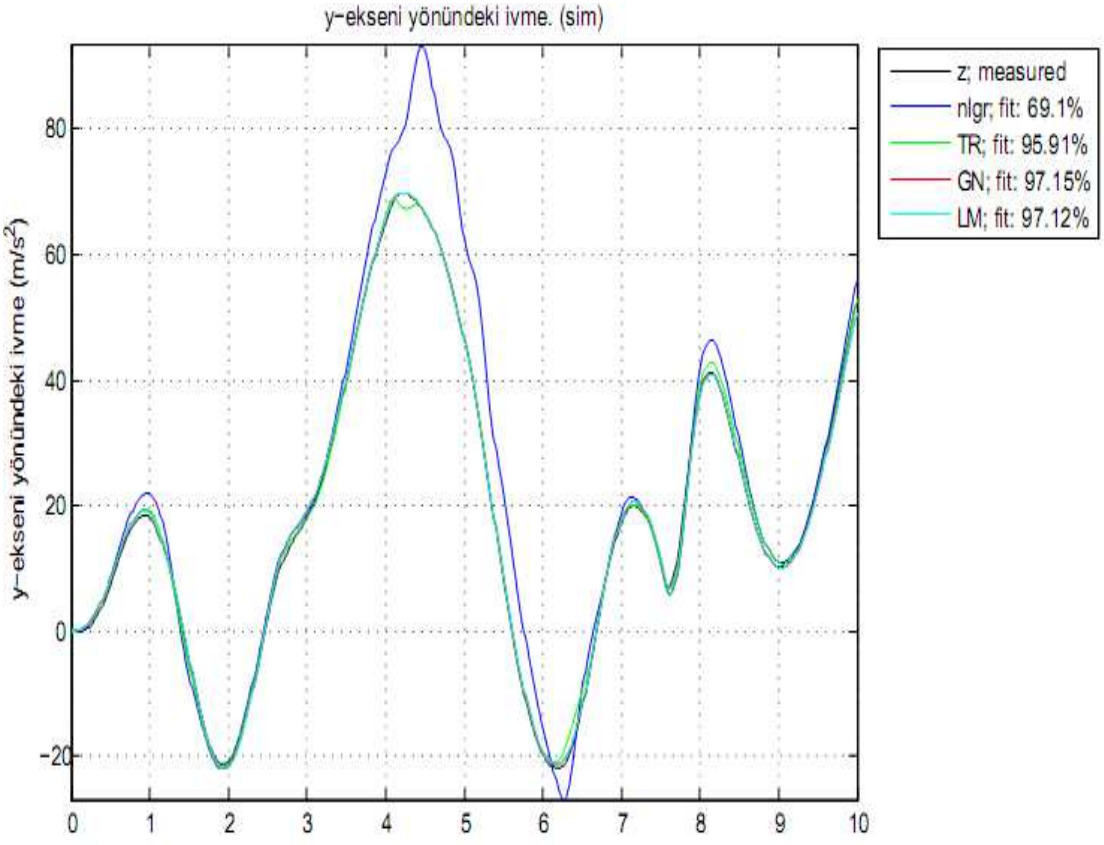
Şekil 5.12: x ekseni etrafındaki açısal hızın kestirimi



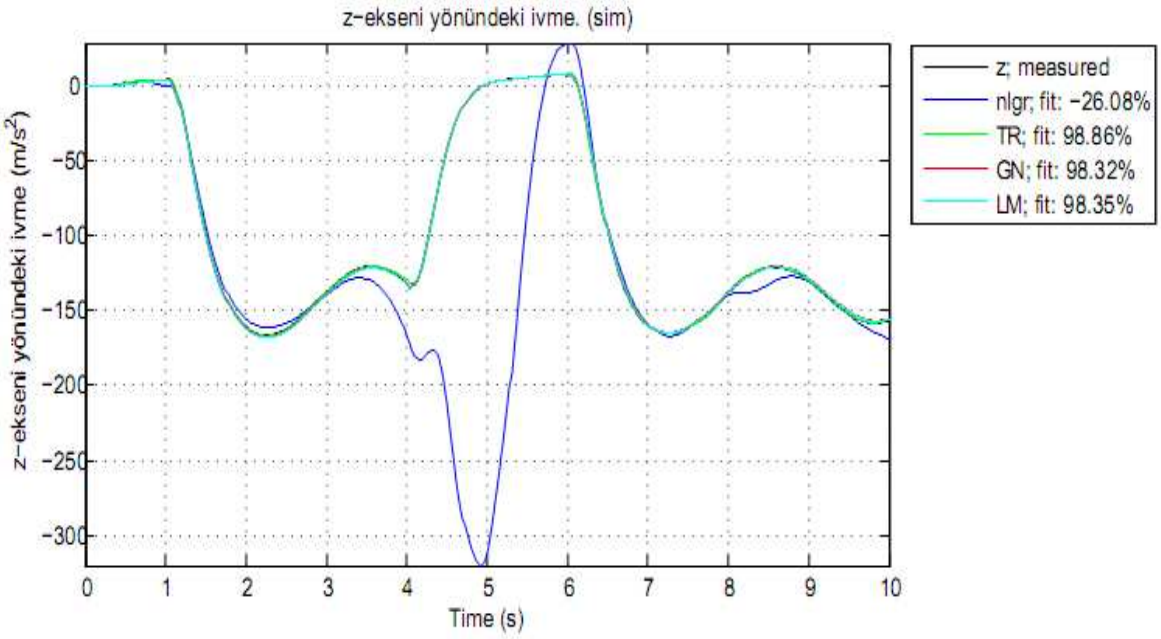
Şekil 5.13: y ekseni etrafındaki açısal hızın kestirimi



Şekil 5.14: z ekseni etrafındaki açısal hızın kestirimi



Şekil 5.15: y ekseni yönündeki ivme



Şekil 5.16: z ekseni yönündeki ivme

Yukarıdaki grafiklerden görüldüğü gibi her bir çıkış parametresi için Trus-Region metodu daha yakın kestirim yapmıştır. Aşağıdaki tabloda her bir optimizasyon metodu için elde edilen sonuçların karşılaştırılması verilmiştir.

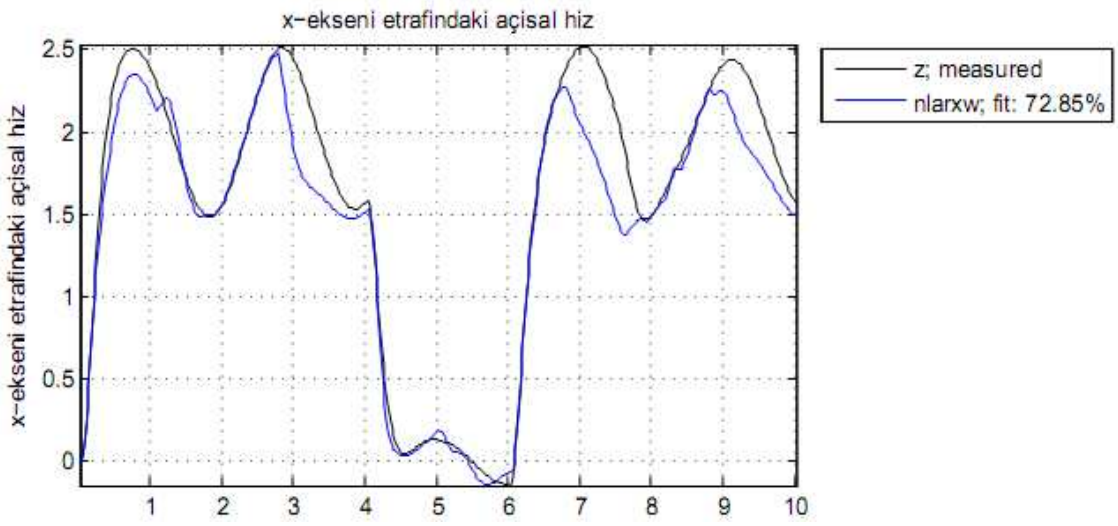
Çizelge 5.7: Her bir optimizasyon metodunun karşılaştırması

Parametreler	Parametre İsimi	İlk Değerler	Trust-Region	Gauss-Newton	Levenberg-Marquardt
p1(1)	Aerodinamik kuvvet katsayısı [1/(rad*m^2]	20	21,3246	94,3587	94,1382
p1(2)		-6	-7,78593	-5,35047	-5,38404
p1(3)		35	36,2971	603,048	606,373
p1(4)		13	13,0201	321,856	325,587
p2(1)	Aerodinamik moment katsayısı [1/rad]	-1	-1,03446	8,27496	8,54838
p2(2)		15	15,337	-22,9249	-23,6132
p2(3)		3	2,96768	-5,20616	-5,38960
p2(4)		-16	-18,7409	-1162,17	-1174,12
p2(5)		-1800	-1160,03	-100647	-101100
p2(6)		-50	-57,0413	-678,754	-689,995
p2(7)		23	34,2341	219,917	221,867
p2(8)		-200	-208,39	-902,37	-877,85

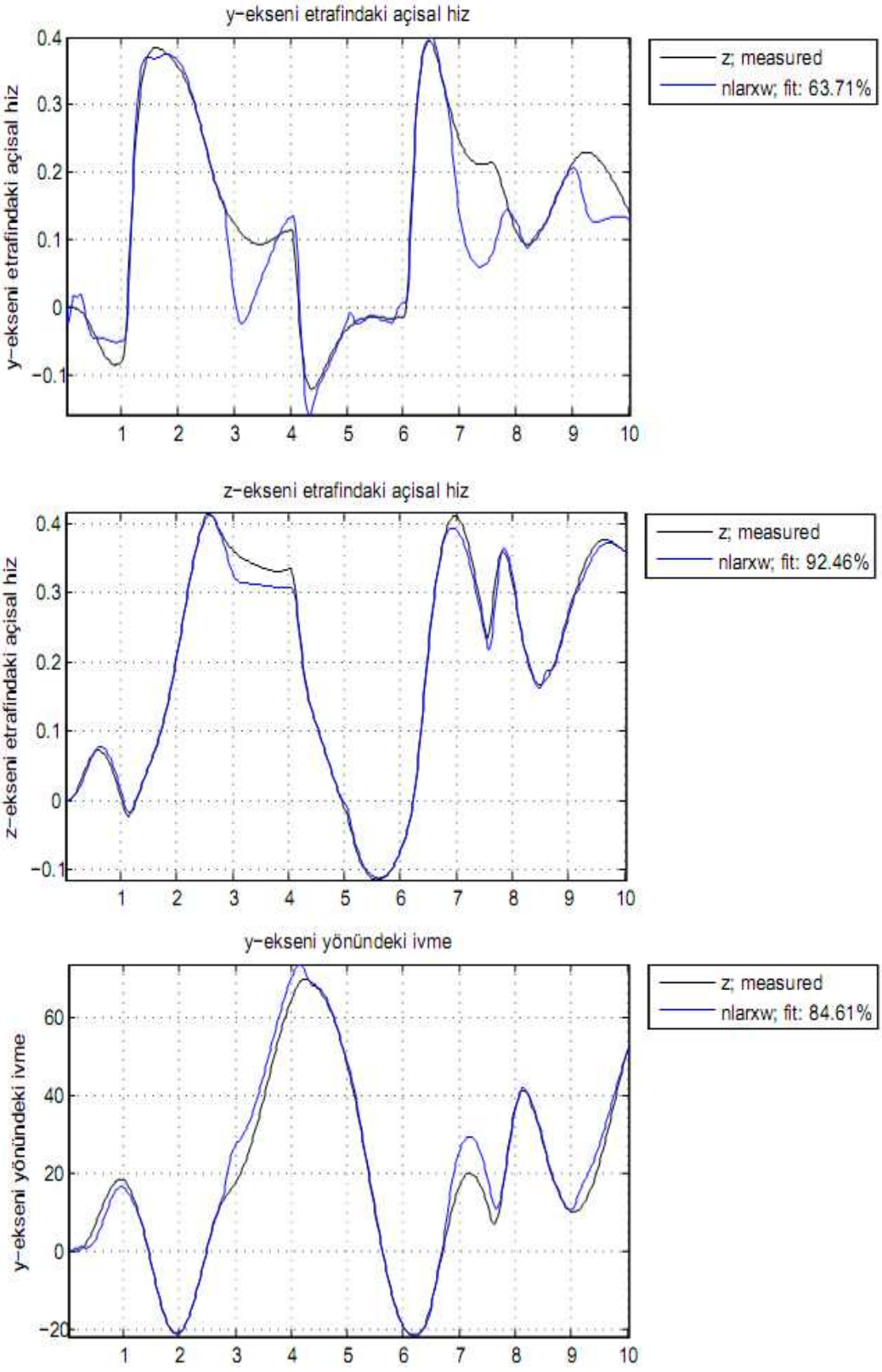
p2(9)		-2000	-2480,44	-45729,7	-45668,6
p2(10)		-17	-32,5455	-42,3662	-44,3638
p2(11)		-50	-50,9551	-4,8358	-4,6519
p3	Aerodinamik düzeltme faktörü [1/(s*rad)]	-5	-0,652	-26,28	-26,2525
p4	Füze çapı [m]	0,17	0,17	0,17	0,17
p5	Füze referans alanı [m^2]	0,0227	0,0227	0,0227	0,0227
p6(1)	x-y-z yönündeki atalet moentleri [kg*m^2]	0,50	0,5	0,5	0,50
p6(2)		110	110	110	110
p6(3)		110	110	110	110
p7	Füzenin kütlesi [kg]	107	107	107	107
p8	Geribesleme kazancı [-]	6	6	6	6

5.2 Lineer Olmayan ARX Modeli Kestirimi

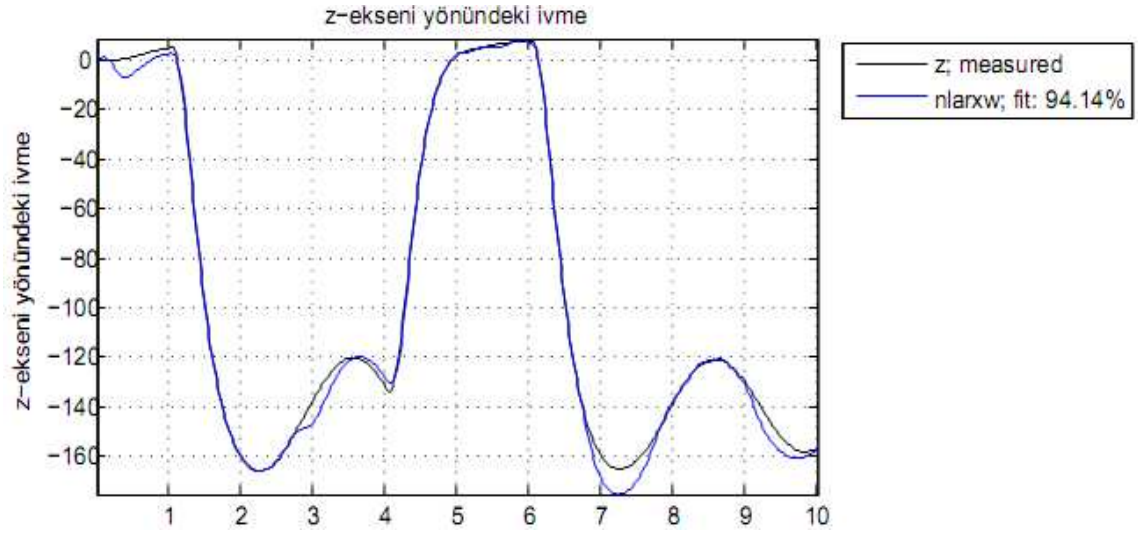
Bundan sonraki süreçte lineer olmayan ARX modeli ile kestirim anlatılacaktır. İlk olarak yukarıda kullanılan veriler yine 10 giriş ve 5 çıkış için kullanılacaktır. Bu verilere kullanarak lineer olmayan ARX modelin yapısına göre kestirim yapılacaktır. Kestirim esnasında iki farklı fonksiyon kullanılacaktır. İlk olarak yukarıda lineer olmayan ARX model konusunda bahsedilen wavenet fonksiyonu kullanılacaktır. Aşağıda wavenet fonksiyonunu kullanılarak elde edilen kestirim sonuçlarının grafikleri verilmiştir.



Şekil 5.17: Wavenet fonksiyonu ile lineer olmayan arx parametre kestirimi

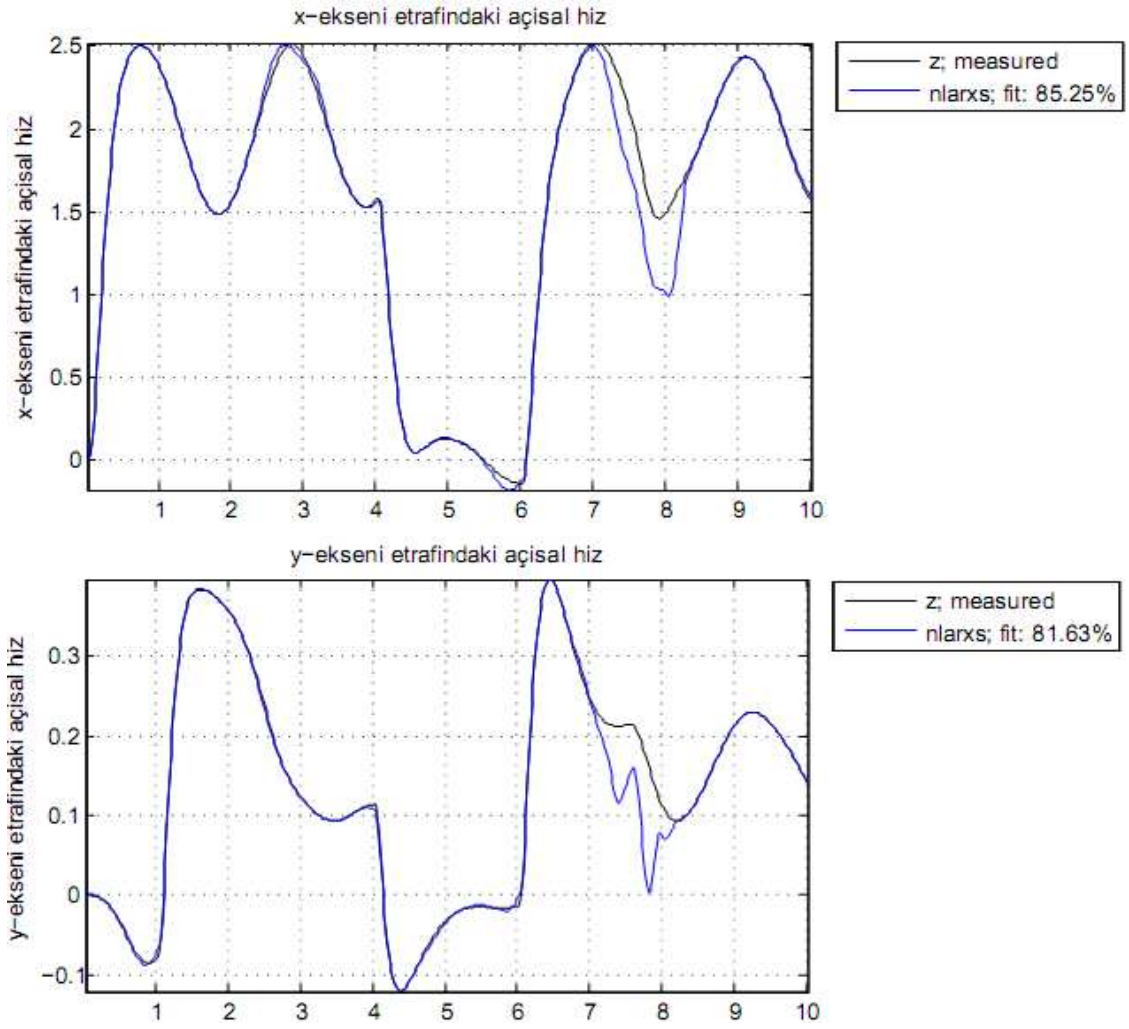


Şekil 5.17: (devam) Wavenet fonksiyonu ile lineer olmayan arx parametre kestirimi

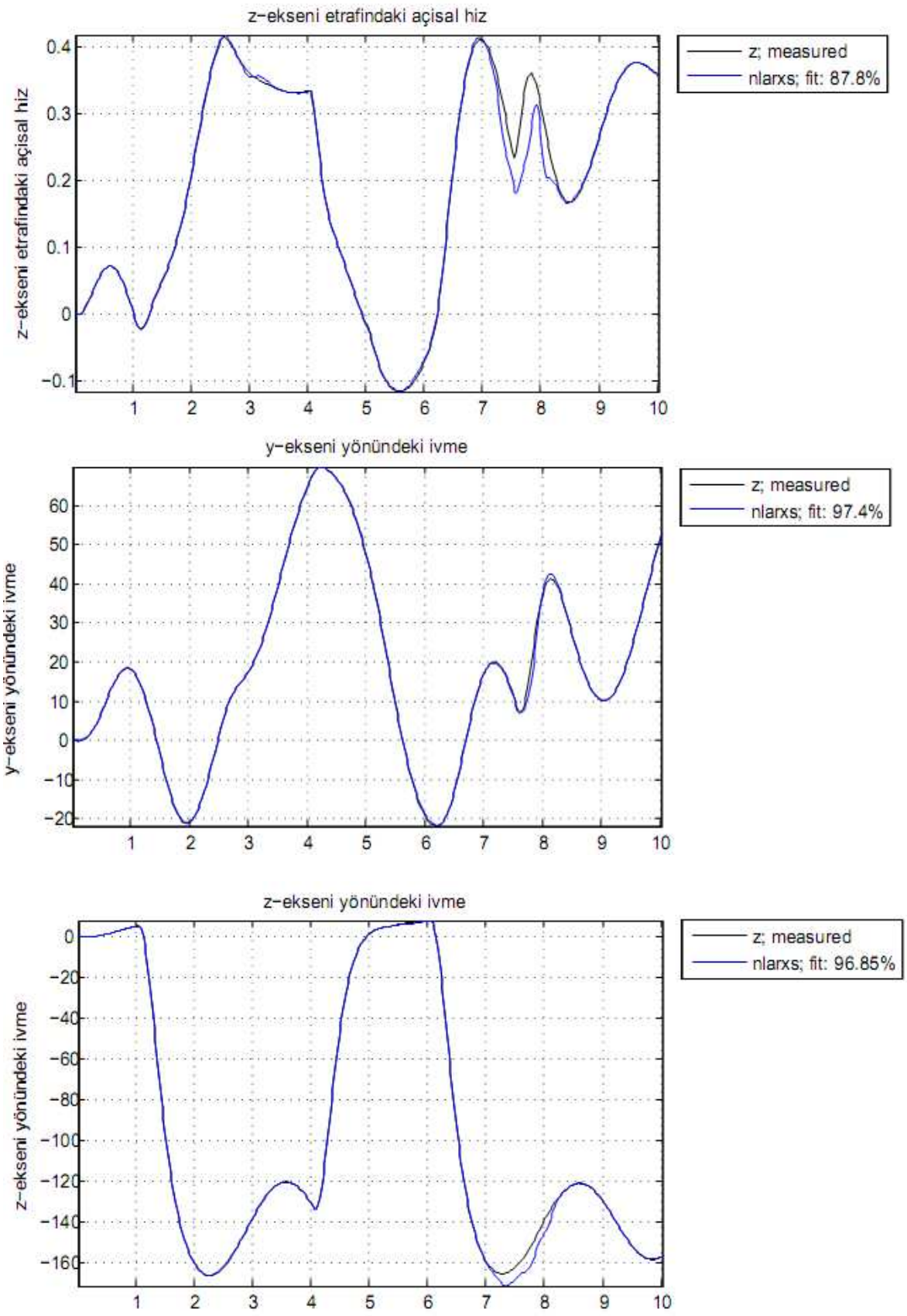


Şekil 5.17: (devam) Wavenet fonksiyonu ile lineer olmayan arx parametre kestirimi

İkinci olarak sigmoident fonksiyonu kullanılarak lineer olmayan parametre kestirimi yapılmıştır. Aşağıdaki grafikler bu uygulamanın sonuçlarını göstermektedir.



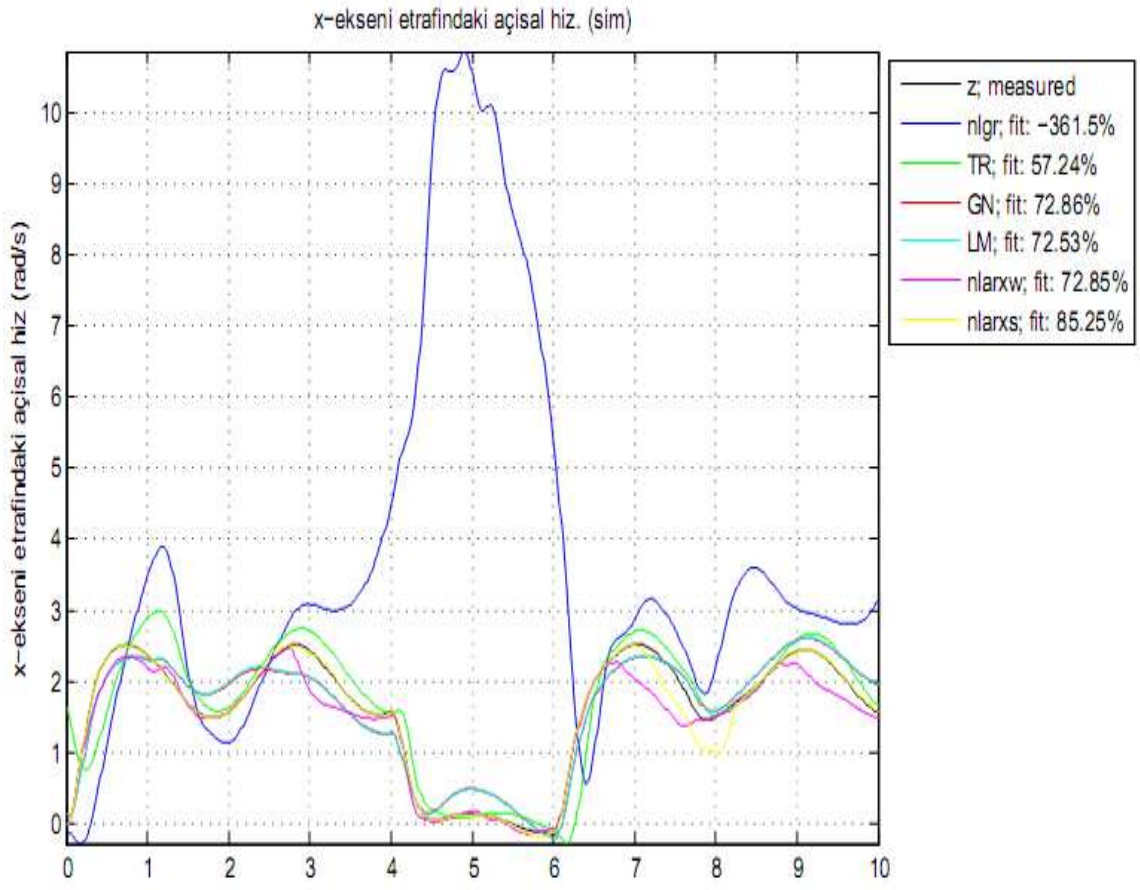
Şekil 5.18: Sigmoident fonksiyonu ile lineer olmayan arx parametre kestirimi



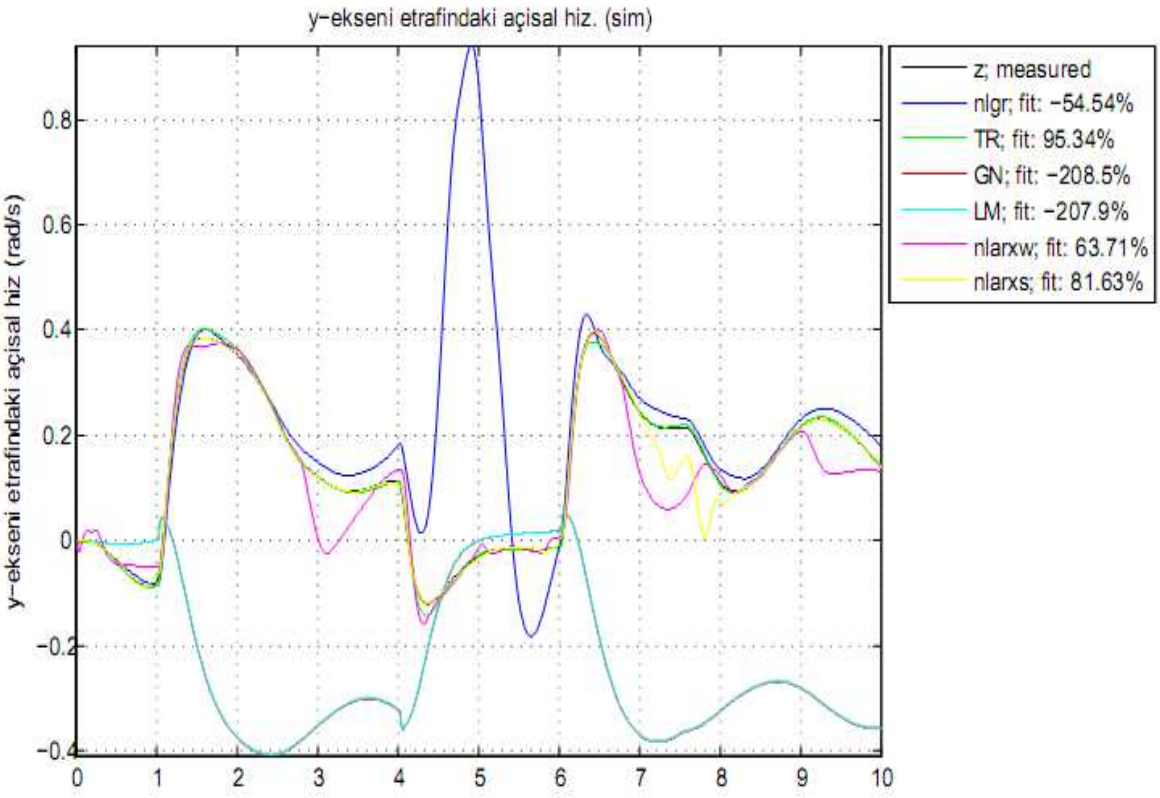
Şekil 5.18: (devam) Sigmoident fonksiyonu ile lineer olmayan arx parametre kestirimi

6. SONUÇLAR VE DEĞERLENDİRME

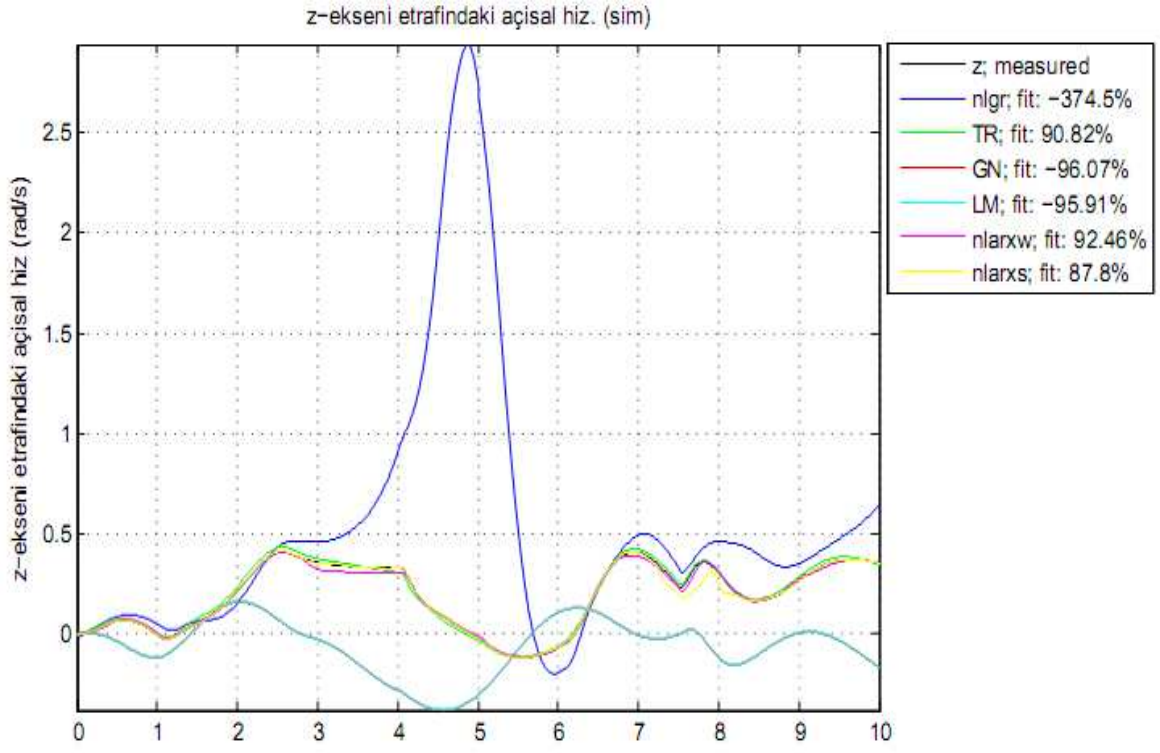
Bu çalışmada lineer olmayan Grey-Box ve lineer olmayan ARX parametre kestirim modellerini kullanarak 10 giriş ve 5 çıkışlı güdümlü füze sisteminin 16 adet bilinmeyen parametresinin kestirimi amaçlanmıştır. Yukarıda güdümlü füze sisteminin verileri kullanılarak lineer olmayan Grey-Box ve lineer olmayan ARX modellerinin uygulaması gerçekleştirildi. Lineer olmayan Grey-Box modeli Trust-Region, Gauss-Newton ve Levenberg-Marquardt optimizasyon metotlarını kullanarak kestirim yapıldı. Lineer olmayan ARX modeli için wavenet ve sigmoidnet fonksiyonları kullanılarak parametre kestirimi yapıldı. Aşağıda her bir sistem çıkışı için bu analizlerin karşılaştırması verilmiştir.



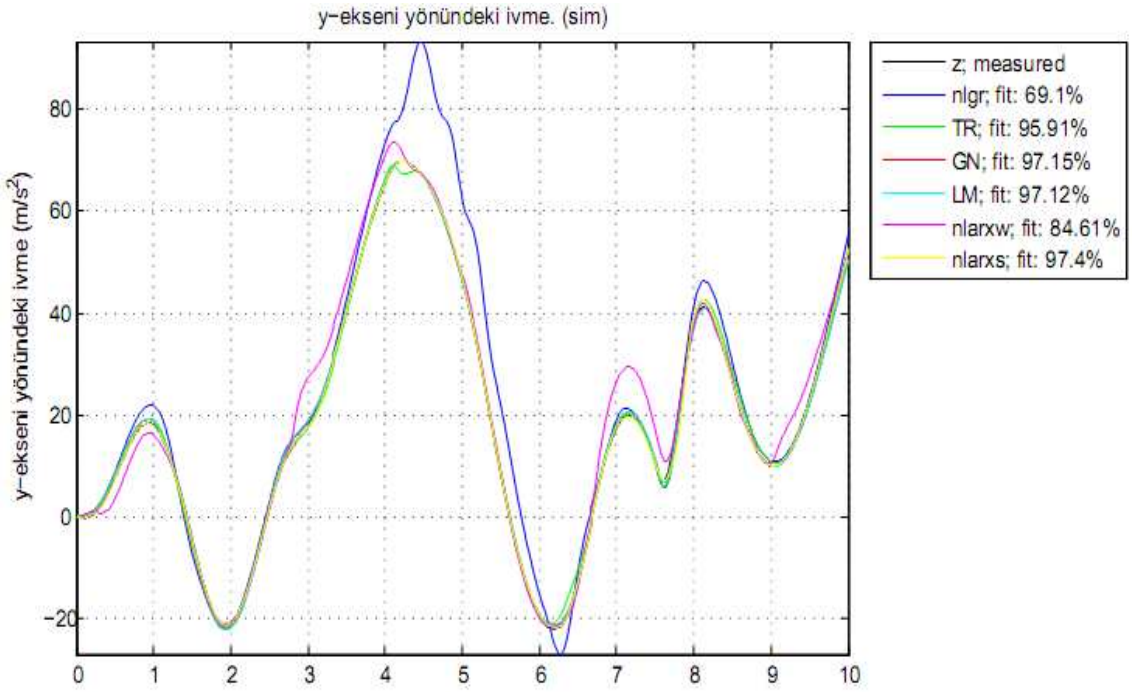
Şekil 6.1: x-ekseni etrafındaki açısal hız



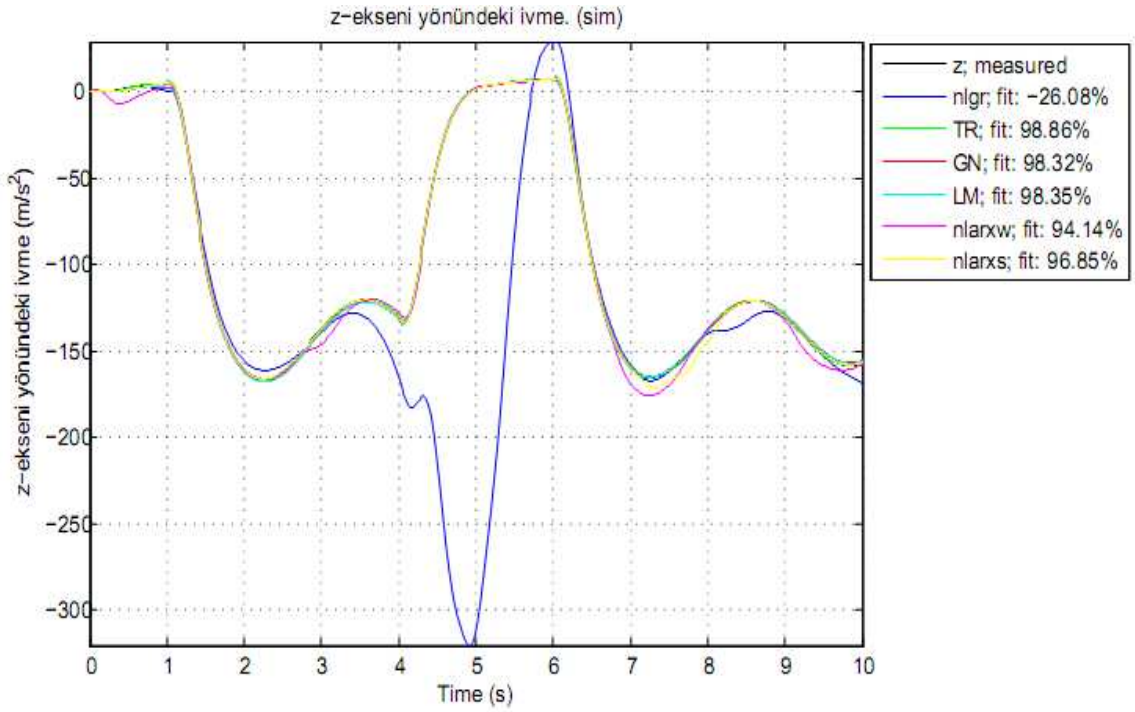
Şekil 6.2: y-ekseni etrafındaki açısal hız



Şekil 6.3: z-ekseni etrafındaki açısal hız



Şekil 6.4: y-ekseni yönündeki ivme



Şekil 6.5: z-ekseni yönündeki ivme

Yukarıdaki sonuçlardan anlaşıldığı gibi lineer olmayan Grey-Box modeli için Trust-Region optimizasyon metodu en uygundur. Lineer olmayan ARX modeli için hem wavenet hem de sigmoidnet fonksiyonları uygun sonuçlar vermektedir. Bu yöntemler kullanılarak ilk parametre değerlerine göre uygun kestirim yapılmaktadır.

KAYNAKLAR

Abdelhakem, D. and Zahari, T., 2010. Identification and Control Of A Small Scale Helicopter

Dimitri, P. B., 1999. Nonlinear Programming.

Erik, W., and Svante, G., 2005. Nonlinear Grey-Box Identification Of Industrial Robots Containing Flexibilities

Eriksson, J. and Gullikson, M. E., 2003. Local Result For The Gauss-Newton Method On Constrained Rank-Deficient Nonlinear Least Squares

Farhan, A. F., and Thanh, L. V.,2002. Mathematical Models for a Missile Autopilot Design

Igor, G., Stephen, G. N., and Ariela, S., 2008. Linear and Nonlinear Optimization.

John, H. B., 1991. Automatic Control of Aircraft and Missiles.

Lennart, L., Qinghua, Z., Peter, L., and Anatolia, J., 2004. Estimation of Grey-Box and Black Box Models for Nonlinear Circuit Datas

Markus, G., 2006. Identification and Estimation for Models Described by Differantial-Algebraic Equations.

Madsen, K., Nielsen, H.B., and Tingleff, O., 2004. Methods For Non-Linear Least Squares Problems

Michael, B. B., 1993. Nonlinear Optimization With Engineering Applications.

Moghbelli, H., Rahideh, A., and Safavi, A. A., 2005. Vector Control of Induction Machines Using Wavenet Based Controller For Traction Applications

Mohammadi, A., Massouminia, M. A., 2002. Missile Aerodynamic Identification Using a Two Step Estimation Procedure

Sam, R., 2009. Levenberg-Marquardt Optimization

Toint, L., 2008. Nonlinear Stepsize Control, Trust Regions and Regularizations For Unconstrained Optimization

Wenyu, S., and Ya-Xiang, Y., 2006. Optmization Theory and Methods

Will, J., and Stelzman, U., 2007. Robustness Evoluation of Crashworthiness Using LS DYNA and Optislang.

Yonathan, B., 1974. Nonlinear Parameter Estimation.

Yuying, L., 1994. A Trust Region and Affine Scaling Method For Nonlinearly Constrained Minimization

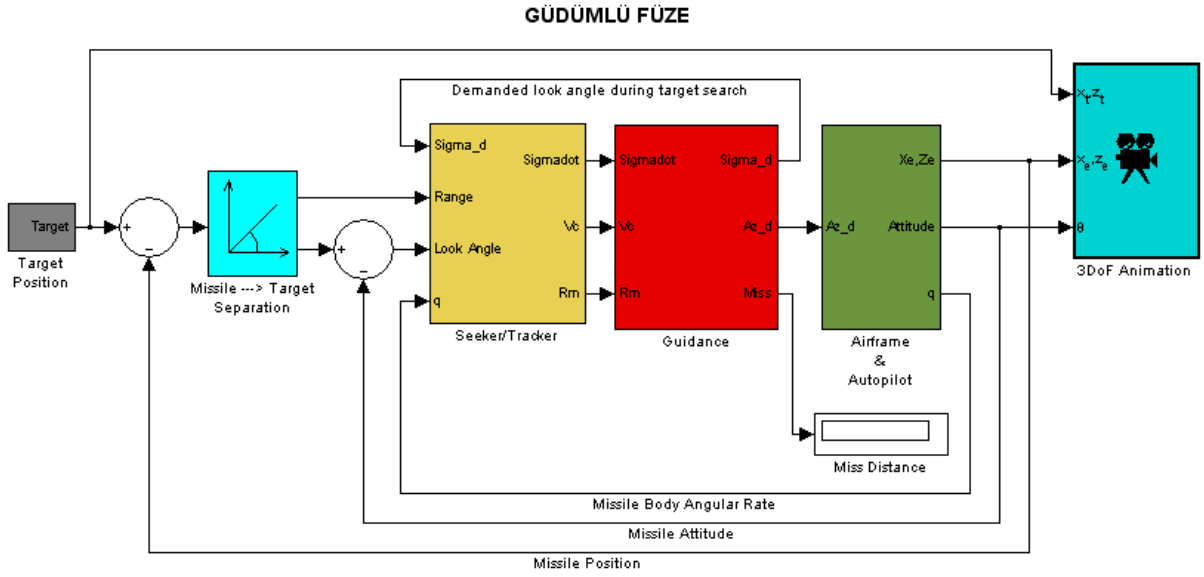
EKLER

EK-A: GÜDÜMLÜ FÜZE SİSTEMİNİN SİMULİNK MODELİ

EK-B: GÜDÜMLÜ FÜZE SİSTEMİNİN PARAMETRE KESTİRİMİ

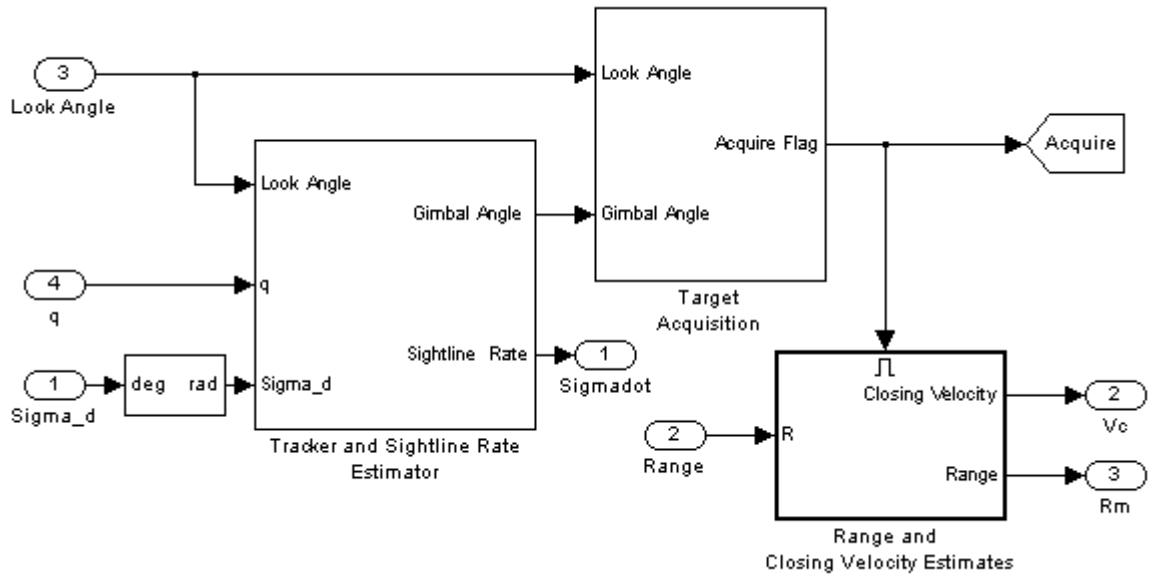
EK-A

Ek-A, güdümlü füze sisteminin simulink modelini içermektedir. Güdümlü füze sisteminin simulink modeli aşağıda gösterildiği gibi hedef bloku, tarayıcı, güdüm ve otopilot olmak üzere 4 ana blok içermektedir.



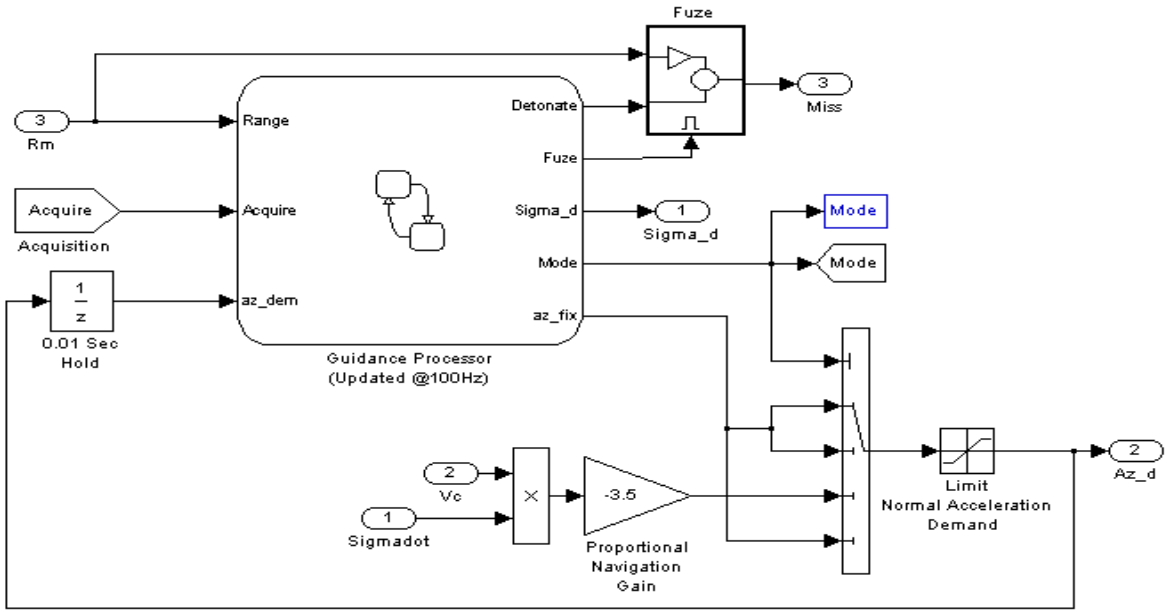
Şekil A.1: Güdümlü füze sisteminin simulink modeli

Aşağıda tarayıcı bloğunun yapısı verilmiştir.



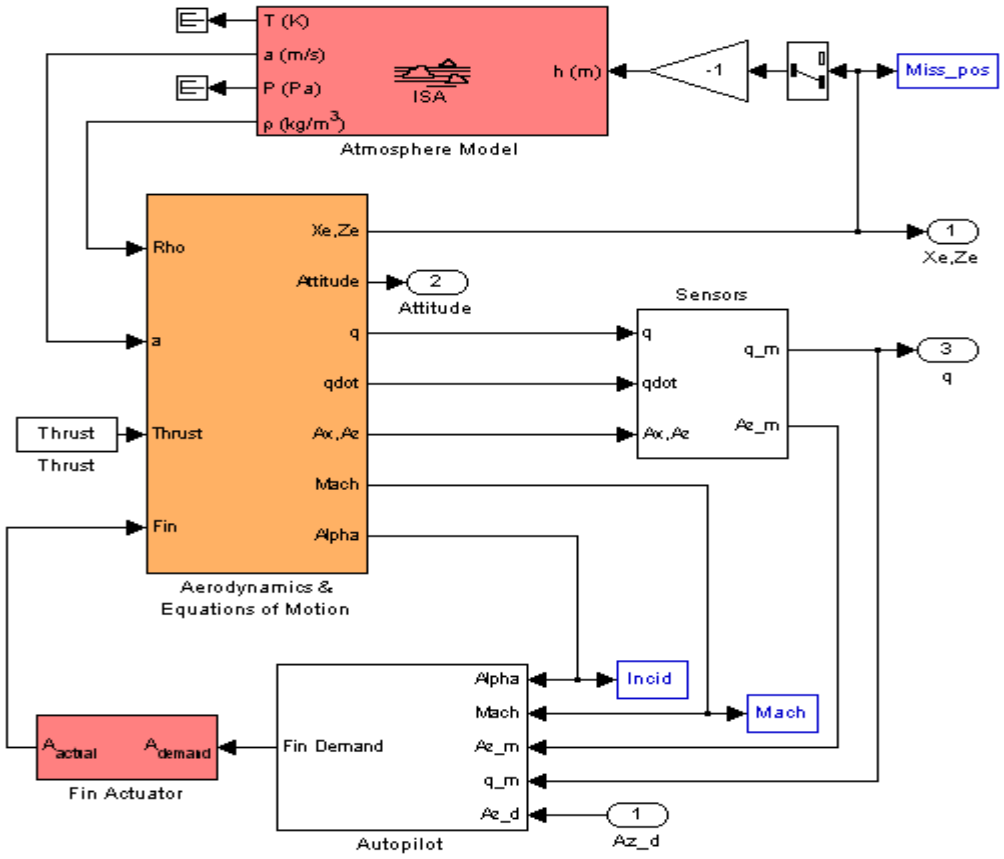
Şekil A.2: Tarayıcı bloğunun bir alt bloğu

Aşağıda güdüm bloğunun alt bloğu verilmiştir.



Şekil A.3: Güdüm bloğunun bir alt bloğu

Aşağıda otopilot bloğunun bir alt bloğu verilmiştir.



Şekil A.4: Otopilot bloğunun bir alt bloğu

EK-B

Aşağıda 5. bölümde anlatılan güdümlü füze parametre kestiriminin Matlab programı verilmiştir.

```
load(fullfile(matlabroot, 'missiledata'));
z = iddata(y, u, 0.02, 'Name', 'Missile');
set(z, 'InputName', {'Yuvarlanma Açisi' 'Yunuslama Açisi'
...
                     'Sapma Açisi' 'Dinamik Basinç'
...
                     'Füze Hizi'
...
                     'x-ekseni etrafinda ölçülmüs açisal hiz'
...
                     'y-ekseni etrafinda ölçülmüs açisal hiz'
...
                     'z-ekseni etrafinda ölçülmüs açisal hiz'
...
                     'Ölçülmüs hücum açisi'
...
                     'Ölçülmüs kayma açisi'},
...
       'InputUnit', {'rad' 'rad' 'rad' 'kg/(m*s^2)' 'm/s'
...
                        'rad/s' 'rad/s' 'rad/s' 'rad' 'rad'});
set(z, 'OutputName', {'x-ekseni etrafindaki açisal hiz'
...
                      'y-ekseni etrafindaki açisal hiz'
...
                      'z-ekseni etrafindaki açisal hiz'
...
                      'y-ekseni yönündeki ivme'
...
                      'z-ekseni yönündeki ivme'},
...
       'OutputUnit', {'rad/s' 'rad/s' 'rad/s' 'm/s^2' 'm/s^2'});
set(z, 'Tstart', 0, 'TimeUnit', 's');
figure('Name', [z.Name ': input data']);
for i = 6:z.Nu
   subplot(z.Nu, 1, i);
   plot(z(:, [], i));
   title(['Giris :' num2str(i) ': ' z.InputName{i}]);
   xlabel('');grid on
   axis('tight');
   if (i > z.Nu-2)
       xlabel([z.Domain ' (' z.TimeUnit ')']);
   end
end
figure('Name', [z.Name ': output data']);
for i = 1:z.Ny
   subplot(z.Ny, 1, i);
   plot(z(:, i, []));
   title(['Çikis :' num2str(i) ': ' z.OutputName{i}]);
   xlabel('');grid on
   axis('tight');
end
xlabel([z.Domain ' (' z.TimeUnit ')']);
Parameters = struct('Name', {'Aerodinamik kuvvet katsayısı'
... % F, 4-by-1 vektör.
```

```
                              'Aerodinamik moment katsayısı'
... % M, 11-by-1 vektör.
                              'Aerodinamik düzeltme faktörü'
... % C, skaler.
                              'Füze çapı'
... % d, skaler.
                              'Füze referans alanı'
... % A, skaler.
                              'x-y-z yönündeki atalet moentleri'
... % I, 3-by-1 vektör.
                              'Füzenin kütlesi'
... % m, skaler.
                              'Geribesleme kazancı'},
... % K, skaler.
                    'Unit', {'1/(rad*m^2), 1/(rad*m^2), 1/(rad*m^2),
1/(rad*m^2)' ...
                              ['1/rad, 1/(s*rad), 1/rad, 1/rad, '
...
                               '1/(s*rad), 1/rad, 1/rad, 1/rad^2, '
...
                               '1/(s*rad), 1/rad, 1/rad']
...
                               '1/(s*rad)' 'm' 'm^2'
...
                               'kg*m^2, kg*m^2,kg*m^2' 'kg' '-'},
...
                    'Value',   {[20.0; -6.0; 35.0; 13.0]
...
                                [-1.0; 15; 3.0; -16.0; -1800; -50;
23.0; -200; -2000; -17.0; -50.0] ...
                                -5.0 0.17 0.0227
...
                                [0.5; 110; 110] 107 6},
...
                    'Minimum', {-Inf(4, 1) -Inf(11, 1) -Inf -Inf -
Inf -Inf(3, 1) -Inf -Inf}, ... % İhmal edilen sınırlar.
                    'Maximum', {Inf(4, 1) Inf(11, 1) Inf Inf Inf
Inf(3, 1) Inf Inf},        ... % İhmal edilen sınırlar.
                    'Fixed',   {false(4, 1) false(11, 1) false true
true true(3, 1) true true});

InitialStates = struct('Name',    {'x-ekseni etrafındaki açısal hız'
...
                                   'y-ekseni etrafındaki açısal hız'
...
                                   'z-ekseni etrafındaki açısal hız'
...
                                   'Hücum açısı' 'Kayma açısı'},
...
                       'Unit',    {'rad/s' 'rad/s' 'rad/s' 'rad'
'rad'},   ...
                       'Value',   {0 0 0 0 0},
...
                       'Minimum', {-Inf -Inf -Inf -Inf -Inf},
... % İhmal edilen sınırlar.
                       'Maximum', {Inf Inf Inf Inf Inf},
... % İhmal edilen sınırlar.
                       'Fixed',   {true true true true true})
  FileName      = 'missile_m';           % Model yapısını tanımlayan
dosya.
```

```
Order        = [5 10 5];                % Model dereceleri [ny nu
nx].
Ts           = 0;                       % Time-continuous sistem.
nlgr = idnlgrey(FileName, Order, Parameters, InitialStates, Ts, ...
                'Name', 'Missile', 'TimeUnit', 's');
set(nlgr, 'InputName', z.InputName, 'InputUnit', z.InputUnit);
set(nlgr, 'OutputName', z.OutputName, 'OutputUnit', z.OutputUnit);

nlgr

figure;
compare(z, nlgr);


figure;
pe(z, nlgr);

duration = clock;
TR = pem(z, nlgr, 'Display', 'on');
duration = etime(clock, duration);

disp(['Kestirim zamanı   : ' num2str(duration, 4) ' saniye']);
disp(['İterasyon birim zamanı: '
num2str(duration/TR.EstimationInfo.Iterations, 4) ' saniye.']);

figure;
compare(z, nlgr, TR);

figure;
pe(z, TR);

present(TR);
```

ÖZGEÇMİŞ

Ad Soyad: Elçin ASLANOV

Doğum Yeri ve Tarihi: Azerbaycan / 25.08.1984

Adres: Çinardere mah. Ayazma cad. Sakarya sok. B2 Blok Daire:11

Pendik/İstanbul

Lisans Üniversite: İstanbul Teknik Üniversitesi