T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

# ALTI EKLEMLİ ROBOT KOLUNUN GENETİK ALGORİTMA VE ELMAN AĞ UYARLAMALI GENELLEŞTİRİLMİŞ ÖNGÖRÜLÜ KONTROLÜ

**DOKTORA TEZİ**

**Elk. Yük. Müh. Burhanettin DURMUŞ**

**Enstitü Anabilim Dalı : ELEKTRİK-ELEKTRONİK MÜH.**

**Enstitü Bilim Dalı : ELEKTRONİK**

**Tez Danışmanı : Doç. Dr. Nejat YUMUŞAK**

**Eylül 2008**

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

# ALTI EKLEMLİ ROBOT KOLUNUN GENETİK ALGORİTMA VE ELMAN AĞ UYARLAMALI GENELLEŞTİRİLMİŞ ÖNGÖRÜLÜ KONTROLÜ

DOKTORA TEZİ

Elk.Yük. Müh. Burhanettin DURMUŞ

Enstitü Anabilim Dalı : ELEKTRİK-ELEKTRONİK MÜH.

Enstitü Bilim Dalı : ELEKTRONİK

Bu tez 09/ 09 /2008 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Prof.Dr. Ekrem KÖKLÜKAYA
Jüri Başkanı

Doç.Dr. Nejat YUMUSAK
Jüri Üyesi

Doç.Dr. Recep KOZAN
Jüri Üyesi

Yrd.Doç.Dr.H. Metin SARAOĞLU
Jüri Üyesi

Yrd.Doç.Dr. A.İhsan ÇANAKOĞLU
Jüri Üyesi

Bu tez Sakarya Üniversitesi Bilimsel Araştırma Projeleri Komisyon Başkanlığı tarafından 2006.50.02.050 numaralı proje kapsamında desteklenmiştir.

# ÖNSÖZ

Bu tez çalışmasında, Model Tabanlı Öngörülü Kontrol (Model Based Predictive Control-MBPC) sınıfına ait olan Genelleştirilmiş Öngörülü Kontrol (Generalized Predictive Control – GPC), Basit Genetik Algoritma uyarlamalı Genelleştirilmiş Öngörülü Kontrol (SGA-GPC), Yapay Sinir Ağlı Genelleştirilmiş Öngörülü Kontrol (Neural Generalized Predictive Control – NGPC) ve Yinelenen Elman Ağ uyarlamalı NGPC (Recurrent Elman's Neural Network implemented NGPC-ENGPC) algoritmaları altı eklemli endüstriyel bir robotik manipülatöre eklem esaslı tork kontrolü için uygulanmıştır. Sürtünme, yük taşıma, yük değişimi ve rasgele bozuculu durumlar altında algoritmaların performansları hem nümerik hem de grafiksel olarak karşılaştırılmıştır.

Bana bu çalışma olanağını sağlayan danışman hocam Doç. Dr. Nejat YUMUŞAK'a, Doç. Dr. Fevzullah TEMURTAŞ'a, Yrd. Doç. Dr. Hasan TEMURTAŞ'a ve ayrıca çalışmalarım sırasında sabırla benden yardımını esirgemeyen eşime sonsuz teşekkürlerimi sunarım.

# İÇİNDEKİLER

# SİMGELER VE KISALTMALAR LİSTESİ

PID : Proportional (Orantılı) + Integral (Tümlev) + Differential (Diferansiyel)

MRAC : Model Referans Adaptif Kontrol (Model Reference Adaptive Control)

MBPC : Model Tabanlı Öngörülü Kontrol (Model Based Predictive Control)

CARIMA : Controlled + Autoregressive + Integrated + Moving + Average

GA : Genetic Algortihm

SGA-GPC : Basit Genetik Algoritma uyarlamalı Genelleştirilmiş Öngörülü Kontrol (Simple Genetic Algorithm implemented Generalized Predictive Control)

GPC : Genelleştirilmiş Öngörülü Kontrol (Generalized Predictive Control)

NGPC : Yapay Sinir Ağlı GPC (Neural GPC)

SISO : Tek Girişli Tek Çıkışlı (Single Input Single Output)

MIMO : Çok Girişli Çok Çıkışlı (Multiple Inputs Multiple Outputs)

$d$ : Proses Zaman Gecikmesi

$u$ : Proses Girişi

$u_{\min}, u_{\max}$ : Proses Girişinin Minimum ve Maksimum Sınır Değerleri

$y$ : Proses Çıkışı

$\hat{y}$ : Öngörülen Proses Çıkışı

$y_r$ : Proses Çıkışı Referans Değerleri

$\xi$ : Gürültü Bileşeni

$A(q^{-1}), B(q^{-1}), C(q^{-1})$ : Proses Modeli Katsayı Polinomları

$E(q^{-1}), F(q^{-1}), G(q^{-1})$ : Genelleştirilmiş Öngörülü Kontrol Parametre Polinomları

$q^{-1}$ : Geriye Doğru Kaydırma Operatörü

$\Delta$ : Fark Alma Operatörü ( $\Delta u = u(t) - u(t-1)$ , $\Delta = 1 - q^{-1}$ )

$n_a, n_b, n_c$ : Polinom mertebeleri

$\mu$ : Unutma Faktörü

$e$ : Hata

$K$ : Kazanç

$N_1$ : Öngörü Ufuğu Başlangıcı (Minimum Maliyet Ufuğu)

$N_2$ : Öngörü Ufuğu Bitimi (Maksimum Maliyet Ufuğu)

$N_u$ : Kontrol Ufuğu

$ym(n+j)$ : Öngörü Ufuğu boyunca ( $j = N_1, \cdots, N_2$ ) Referans Yörünge

$yn(n+j)$ : Öngörü Ufuğu boyunca ( $j = N_1, \cdots, N_2$ ) Öngörülen Nöral Ağ Çıkışları

$u(n+j)$ : Kontrol Ufuğu boyunca ( $j = 1, \cdots, N_u$ ) Öngörülen Kontrol Girişleri

$\lambda$ : Kontrol Giriş Ağırlık Faktörü

$J$ : Maliyet Fonksiyonu

$U(k)$ : Kontrol Giriş Vektörü

$\frac{\partial J}{\partial U}(k)$ : Jacobian

$\frac{\partial^2 J}{\partial U^2}(k)$ : Hessian

$n_d$ : NGPC'de kullanılan Geçmiş Zaman Dilimindeki Giriş Sayısı

$d_d$ : NGPC'de kullanılan Geçmiş Zaman Dilimindeki Çıkış Sayısı

$hid$ : NGPC algoritmasındaki YSA'nın Ara Katman Eleman Sayısı

$f(Net)$ : Transfer Fonksiyon

$\eta, \alpha, b$ : Eğitim Oran, Momentum ve Bias Parametreleri

$\overline{O}_{pk}$ : Ara Katman Çıkışı

$O_{pm}$ : Çıkış Katman Çıkışı

$O_i(n-1)$ : Geciktirilmiş Ara Katman Çıkışı

$E_p$ : Herhangi Bir Veri için Toplam Hata

$E$ : Bütün verileri Kapsayan Toplam Hata

$w_{mk}$ : Yapay Sinir Ağ Ağırlık Değişkenleri

$b_m, \overline{b}_k$ : Yapay Sinir Ağı Katman Bias 'ı

$q$ : Manipülatör Eklem Değişken Vektörü

$\theta_i$ : $i$. Eklem Dönme Açısı ( *Radyan* )

$\dot{\theta}_i$ : $i$. Eklem Dönme Açısal Hızı ( *Radyan / saniye* )

$\ddot{\theta}_i$ : $i.$ Eklem Dönme Açısal İvmesi ( $Radyan/saniye^2$ )

$\theta_i, d_i, a_i, \alpha_i$ : $i.$ Eklem Parametreleri

$A_{i-1}^{i}$ : Koordinat Homojen Dönüşüm Matrisi

$R_i^{i-1}$ : Koordinat Rotasyon Matrisi

$\vec{n}$ : Elin Birim Normal Vektörü

$\vec{s}$ : Elin Birim Kayma Vektörü

$\vec{a}$ : Elin Birim Yaklaşma Vektörü

$\vec{p}$ : Elin Pozisyon Vektörü

L : Lagrangian Fonksiyonu

K : Manipülatörün Toplam Kinetik Enerjisi

P : Manipülatörün Toplam Potansiyel Enerjisi

$J_i$ : $i.$ Uzvun Atalet Matrisi

$m_i$ : Uzuv Kütlesi

$m$ : Manipülatör tarafından Taşınan Yük

$\tau(t)$ : Eklem Tork Vektörü ( $N.n$ )

$g$ : Yerçekimi İvmesi ( $g = 9.8062\ m/sn^2$ )

$D(\theta)$ : İvmelenme il ilgili Simetrik Atalet Matrisi

$H(\theta, \dot{\theta})$ : Koriolis ve Merkezkaç Kuvvet Vektörü

$G(\dot{\theta})$ :Yerçekimi Kuvvet Vektörü

$J(\theta)$ : Manipülatör Jacobian Matrisi

$\vec{x}_i, \vec{y}_i, \vec{z}_i$ : $i.$ Koordinat Çerçevesi

$\vec{x}_0, \vec{y}_0, \vec{z}_0$ : Referans Çerçeve

P(c) : Çaprazlama Olasılığı

P(m) : Mutasyon Olasılığı

# ŞEKİLLER LİSTESİ

# TABLOLAR LİSTESİ

# ÖZET

Anahtar Kelimeler: Robotik Manipülatör, Genelleştirilmiş Öngörülü Kontrol, Nöro Genelleştirilmiş Öngörülü Kontrol, Genetik Algoritma, Yinelenen Elman Yapay Sinir Ağı

Bu tez çalışmasında, Model Tabanlı Öngörülü Kontrol (Model Based Predictive Control-MBPC) sınıfına ait olan Genelleştirilmiş Öngörülü Kontrol (Generalized Predictive Control - GPC), Basit Genetik Algoritma uyarlamalı Genelleştirilmiş Öngörülü Kontrol (SGA-GPC), Newton-Raphson uyarlamalı Yapay Sinir Ağlı Genelleştirilmiş Öngörülü Kontrol (Neural Generalized Predictive Control - NGPC) ve Yinelenen Elman Yapay Sinir Ağ uyarlamalı Genelleştirilmiş Öngörülü Kontrol (Recurrent Elman Network implemented Neural Generalized Predictive Control - ENGPC) algoritmaları altı eklemli endüstriyel bir robotik manipülatöre eklem esaslı yörünge kontrolü için uygulanmıştır. Robotik manipülatörün dinamik modellenmesinde Lagrange-Euler yöntemi kullanılmıştır. Dinamik modellemeye sürtünme etkileri, yük taşıma ve taşınan yükün taşıma esnasında düşmesi durumları da ilave edilmiştir. Ayrıca, kontrolü güçleştirmek için $-0.5\,Nm$ ile $+0.5\,Nm$ arasında rasgele bozucular ilave edilmiştir. Dinamik model, 4. mertebeden Runge-Kutta bütünleştirme yöntemi kullanılarak robot kolu simülatörüne dönüştürülmüştür. Tasarlanan kontrol algoritmalarının performansı eklemlere ait tork, açısal yörünge, açısal hız, açısal hız hataları grafikleri ile eklemlere ait açısal konum hataları, açısal hız hatalarının kareleri ve uç nokta konum hataları üzerinden hem grafiksel hem de nümerik sonuçlarla karşılaştırılmıştır.

# THE SIX-DEGREES-OF-FREEDOM (6-DOF) ROBOTIC MANIPULATOR CONTROL USING GENETIC ALGORITHM AND ELMAN NEURAL NETWORK IMPLEMENTED GENERALIZED PREDICTIVE CONTROL

## SUMMARY


Key Words: Robotic Manipulator, Generalized Predictive Control, Neural Generalized Predictive Control, Genetic Algorithm, Recurrent Elman's Neural Network

In this thesis study, Generalized Predictive Control (GPC), Neural Generalized Predictive Control (NGPC), Simple Genetic Algorithm implemented GPC (SGA-GPC) and Recurrent Elman Neural Network implemented NGPC (ENGPC) algorithms belong to the class of Model Based Predictive Control (MBPC) were investigated and each of them was applied to a 6-DOF (Degrees-Of-Freedom) robotic manipulator as SISO (Single Input Single Output) and MIMO (Multiple Inputs Multiple Outputs) for the trajectory control based joint. Dynamics modeling of the robotic manipulator was made by using the Lagrange-Euler equations. The frictional effects, the state of carrying and falling load were also added to dynamics model. In addition, the random distortions between $-0.5\ Nm$ and $+0.5\ Nm$ were added to the torques applied to the joints in every control step, and the effect to the performance of the distortions was investigated. Dynamics model was transformed into robotic arm simulator by using the fourth-order Runge-Kutta integration method. The trajectory planning for the joints of the robotic arm was designated according to the sinusoidal trajectories principles. The control algorithms were compared with themselves for different examples and cases.

# BÖLÜM 1. GİRİŞ

Günümüz otomasyon sistemlerinin vazgeçilmez elemanı olan robotlar; üretim, hizmet, güvenlik, sağlık gibi birçok alanda yaygın bir şekilde kullanılmaktadır. Otomobil fabrikalarından havacılık sanayiine, ameliyat operatörlüğünden bomba imha görevlerine kadar birçok görevleri üstlenmektedirler. İnsan gücüne oranla daha ucuz olmaları, üretim hızını, kapasitesini ve verimliliğini arttırmaları robotları önemli kılmaktadırlar [1-5].

Geniş bir uygulama alanı bulunan robotlar için yıllardır birçok kontrol metotları geliştirilmiş, dinamik modelleri oluşturulmuş, icra edeceği hareket ve görevleri için değişik yapıda ve ekleme sahip tasarımlar yapılmıştır. Bu tasarımların en yaygın olanları kol benzeri yapılarda olan robot kollarıdır [6].

Robot kolları uzuvların birbirlerine eklemler aracılığı ile bağlandığı ve ilk eklemin sabitlendiği yapılardır. İcra edeceği göreve bağlı olarak uzuvların şekilleri değiştirilebilir ve uzuv sayıları arttırılabilir. Robot kolunda uzuv sayısını arttırmak kolun hareket manevrasını arttırmaktadır. Ancak eklemler arasında yüksek oranda etkileşimler mevcut olduğundan eklem sayısı arttırmak kolun kontrolünü güçleştirecektir. Dolayısıyla eklem sayısının optimum düzeyde tutulması istenir [7].

Robot kolunun dinamik kontrolü, robot kolu eklemlerinin istenilen pozisyon ve hız referansları doğrultusunda hareketlerini sağlayacak giriş bilgilerini (tork / voltaj) üreterek eklemlere vermek şeklindedir. Ancak, robot kolunun dinamik davranışını veren denklemlerin ikinci dereceden doğrusal olmayan diferansiyel denklemler olmaları ve aralarında yüksek oranda etkileşimler bulunması robot kolunun kontrolünü zorlaştırmaktadır. Bu yüzden klasik kontrol sistemlerini kullanan endüstriyel robot kolları belirli bir hız limitinin üzerine kolayca çıkamamakta, sonuç olarak üretim verimliliği sınırlanmaktadır. Ayrıca, robot kolu kontrolörlerinden

beklenen ve her geçen gün artan performans isteklerinden dolayı daha gelişmiş kontrol tekniklerine gereksinim duyulmaktadır [8, 9].

Kullanılan ticari robotların büyük çoğunluğu nispeten basit olan kontrol sistemleri ile donatılmışlardır. Bunlara örnek olarak PI (Proportional Integral) ve PID (Proportional + Integral + Differential ) tipi kontrolörler gösterilebilir. Fakat bu tip kontrol sistemlerinin sadece düşük hızlarda yeterli olabildikleri yapılan çalışmalar ile kanıtlanmıştır [10].

Endüstride kullanılan robot kolu kontrolörleri yukarıda da belirtildiği gibi genellikle basit bir yapıya sahiptirler. Eklemler arasındaki etkileşimlerin önemsiz sayılabilecek düzeyde kalabilmesi için eklem hızları düşük tutulmakta ve her bir eklem ayrı bir PID tipi kontrolör ile denetlenmektedir. Robot kolunun çalışma hızını artırabilmek için eklem hızları arttırıldığında etkileşimler ve robot kolu modelindeki belirsizlikler nedeniyle istenilen yörünge ile gerçekleşen yörünge arasında önemli hatalar ortaya çıkmaktadır. Bu nedenle klasik kontrol sistemlerini kullanan robot kolları, yeterli hassasiyeti yakalamak için düşük hızlarda çalıştırılmaktadır [9].

Robot kolu kontrolü için pek çok yöntem önerilmiştir. Bunlara; klasik kontrolörler [10, 11], optimum kontrolörler [12, 13], bulanık mantık kontrolörler [14, 15], yapay sinir ağlı kontrolörler [16-18] ve öngörülü kontrolörler [19, 20] gibi örnekler verilebilir.

Dubowsky ve Des Forges [21], 1979'da robot kolu kontrolü için model referans adaptif kontrol (MRAC) kanunu tasarlamışlardır. Çalışmalarında robot kolu eklemleri arasındaki etkileşimleri ihmal etmişler ve her bir eklem için referans model olarak doğrusal ikinci dereceden zamana bağlı olmayan bir diferansiyel denklem takımı kullanmışlardır. Robot kolu pozisyon ve hız geri besleme kazançlarının ayarlanması ile kontrol edilmektedir. Dubowsky ve Des Forges [22], 1979'daki diğer bir çalışmalarında istenen durum vektörü ile robot kolunun durum vektörü arasındaki fark olarak tanımlanan hatanın ikinci dereceden bir fonksiyonunu minimize eden bir adaptif kontrol algoritması geliştirmişlerdir.

Çok aşamalı bir maliyet fonksiyonunu minimize etmeye dayanan bir metot ilk olarak 1973'de Peterka ve Aström [23] tarafından gerçekleştirilmiştir. Bu metot durum uzayına dayanan LQG ( Linear - Quadratic - Gaussian ) metodu olup kestirilecek modelin eşdeğer durum uzayı gösterimine çevrilmesi, aşırı parametrelendirmeye ve hesapsal yüke sebep olmaktadır. Bu yükü hafifletmek için Lam [24], LQG yaklaşımını genişletmiş ve ortak Riccati denklemini çevrim başına sadece bir kez kullanan yeni bir metot geliştirmiştir. Clarke ve arkadaşları [25] Lam'ın bu yaklaşımını öz uyarlamalı kontrolöre uyarlamışlardır. Clarke'nin elde ettiği algoritma N adımlı ufuk ve CARIMA (Controlled Autoregressive Integrated Moving Average) modeli kullanmaktadır. Kullanılan CARIMA modeli elde edilen kontrol kanununda tümlev etkiyi de doğal olarak içermektedir. Bu algoritma değişken sistem zaman gecikmeli, minimum olmayan faz ve açık çevrimli kararsız sistemler için de etkili olabilmektedir.

Lee ve Chung [26, 27], geniş bir hareket sahası ve taşınan yük durumu için zamana bağlı bir yörüngeyi mümkün olduğunca yakın takip edecek adaptif pertürbasyon kontrol çalışması yapmışlardır. Önerdikleri adaptif kontrol istenen bir yörünge civarında doğrusallaştırılmış pertürbasyon denklemlerine dayanmaktadır. Kontrol edilecek sistem ayrı veya birlikte hesaplanabilen ileri ve geri besleme bileşenleri ile karakterize edilmektedir. İleri besleme bileşeni hareketin Newton-Euler denklemlerinden pozisyon ve hızları hesaplamaktadır. Geri besleme bileşeni, ardışık en küçük kareler tanılama işlemini içermektedir ve uygun bir öz uyarlamalı kontrol algoritması doğrusallaştırılmış sistem için istenen yörünge boyunca robot kolunun pozisyon ve hız hatalarını azaltan pertürbasyon torklarını hesaplamaktadır. Önerilen adaptif kontrolün performansını değerlendirmek için bir simülasyon çalışması da ilave edilmiştir.

Clarke ve arkadaşları [28-31] tarafından geliştirilen ve kendinden önceki uzun menzilli öngörülü kontrol algoritmalarının bir sentezi niteliğini taşıyan Genelleştirilmiş Öngörülü Kontrol (Generalized Predictive Control - GPC) algoritması bir gerileyen ufuklar metodudur. Robot kontrolü gibi referans yörüngenin önceden programlandığı durumlar için oldukça kullanışlıdır. Clarke [32] algoritmayı

tek uzuvlu bir robot koluna uygulamış ve PID tipi bir kontrolör ile yörünge takibini karşılaştırmıştır.

Kaynak [33-35] tarafından da öngörülü kontrol algoritmalarının robot kollarına uygulanması konusunda çeşitli çalışmalar yapılmıştır. Kaynak çalışmalarında [34], GPC algoritmasına eşdeğer iki tane yeni öngörülü yöntem önermiştir. Elde ettiği simülasyon sonuçları bu algoritmaların robot kollarının yörünge kontrolünde çok verimli bir şekilde kullanılabileceğini göstermiştir. Kaynak yaptığı diğer bir çalışmada [35], öngörülü kontrol algoritmasını iki serbestlik dereceli düzlemsel bir artık robota uygulamıştır. Simülasyon sonuçları istenilen yörüngelerin büyük bir doğrulukla izlendiğini göstermiştir.

Kazan çalışmasında [36], GPC algoritmasını değişik dinamik sistemler için değişik etkenler altında test etmiştir. Ayar parametrelerinin performans üzerindeki etkilerini araştırmıştır. Literatürlerde sözü edilen düşünceler doğrultusunda sonuçlar almıştır. Kazan, daha sonra algoritmayı üç eklemli bir robot koluna uygulamıştır. Tek giriş tek çıkış (Single Input Single Output - SISO) ve çok giriş çok çıkış (Multiple Inputs Multiple Outputs - MIMO) olmak üzere iki ayrı durum için incelemiştir. SISO ve MIMO GPC algoritmalarının taşınan yük ve değişimi, sürtünme etkileri, ölçme hataları gibi değişik bozucular etkisinde yörünge kontrolündeki performansını araştırmıştır. GPC algoritmasının iyi bir kontrol performansı sağladığını göstermiştir.

Doğrusal olmayan özelliklerinden dolayı yapay sinir ağları da robot kollarının öngörülü kontrolünde kullanılmış ve başarılı sonuçlar alınmıştır [19, 37, 38].

Öz [39] çalışmasında, yapay sinir ağlarını incelemiş ve üç eklemli bir robot koluna uygulamıştır. Öz, önce Kazan [36] tarafından üç eklemli bir robot koluna uygulanan GPC algoritmasını araştırmış ve bu algoritmanın iyi bir performans gösterdiğini gözlemlemiştir. GPC algoritmasından elde ettiği kontrol bilgilerini gerçekleştirdiği öngörülü kontrolörün eğitilmesinde kullanmıştır. Gerçekleştirdiği kontrolörün robot kolu kontrolünde GPC algoritmasına yakın sonuçlar verdiğini göstermiştir.

2003 ‘de Hasan Temurtaş üç eklemli bir robot kolunu GPC ve NGPC ile eklem esaslı kontrolünü gerçekleştirmiştir [40]. Temurtaş çalışmasında yapay sinir ağı tekniğinin GPC ‘nin kabiliyetini arttırdığını vurgulamış, NGPC ile eklemlerin referans yörüngelerini çok daha yakın takip ettiğini göstermiştir. Ancak, üç eklemli robot kolunun endüstriyel uygulamalar için yeterli olmadığını belirtmiş ve eklem sayısının arttırılmasını önermiştir.

Görüldüğü gibi, adaptif kontrol ve uzun menzilli öngörülü kontrol teknikleri [41-47], robot kollarının performansında önemli gelişmeler sağlamıştır. Özellikle tekrarlı görevlerde büyük performans göstermişlerdir. Ayrıca, robot kollarına eğitilebilir özellikler de kazandırmışlardır. Yüksek çalışma hızları ve taşınan yükteki değişmeler durumunda ortaya çıkan etkileri dengeleme özelliğine sahiptirler.

Öte yandan yapılan işe bağlı olarak birden fazla robot kolunun ortak çalışmalar için organize edildiği çalışmalar da yapılmıştır. Cassemiro iki adet üç eklemli robot kolunu PID ve GPC kontrolör kullanarak ortak bir tasarım gerçekleştirmiştir [48]. Çalışmada robot kolunun esnekliği vurgulanmış, GPC ‘nin PID ye göre daha hızlı ve daha az hatalı kontrol gerçeklediğini özellikle bozuculu durumlarda daha başarılı olduğunu gösterilmiştir.

Gelişen uygulama alanları ile birlikte robot kollarında daha fazla ekleme ihtiyaç duyulmuştur. Örneğin bir tutma görevi için kolun uç noktasına el yapısına benzer bir tutucuya ihtiyaç duyulur. Robot kolunda eklem sayısını arttırmak kolun hareket kabiliyetini arttırmaktadır. Ancak eklemler arasında aşırı derecede etkileşim var olduğundan eklem sayısını arttırmak dinamik modeli daha da karmaşık hale getirmekte ve kontrolü güçleştirmektedir. Bu nedenle optimum eklem sayısı istenir.

Üç eklemli robot kollarından sonra 5 ve 6 eklemli robot kolları üretilmiş ve büyük bir kabul görmüştür. Bunlar arasında yer alan Puma 560 6 eklemli bir robot koludur. İnsan koluna benzerliği ile göze çarpan robot kolu, insan kolunun hareket esnekliğini üzerinde bulunduran bir yapıda üretilmiştir. Başta küçük parçaların elle tutulması gibi uygulamalarda olmak üzere bu kol özellikle eğitim amaçlı olarak araştırmacılarca yaygın bir şekilde kullanılmaktadır. Literatürde Puma 560 kontrolü

üzerine yapılan birçok çalışma mevcuttur. Armstrong ve ekibi kolun dinamik modelini oluşturarak tüm parametre değerlerini hesaplamışlardır [49].

Khatib 2002 'deki çalışmasında Puma 560 kolunun uç elemanına taktığı temizleme elemanı ile uçak kabin camı temizleme uygulaması yapmıştır [50]. Kolun kontrolünü mobil olarak gerçeklemiştir.

Sonuç olarak altı ekleme sahip Puma 560 robot kolu, otomotiv panel montajı, elektronik devre baskı yazımı, radyo-televizyon setlerinin montajı gibi yüksek hassasiyet gerektiren görevler ile eczacılık ve gıda sektöründe paketleme gibi alanlarda elle tutma, taşıma ve kaldırma görevleri için kullanılan bir endüstriyel robot koludur. Sahip olduğu bu özelliklerinden dolayı tez çalışmasında bu robot kolu tercih edilmiştir.

Günümüz kontrol sistemlerinde optimizasyona olan talebin artması, problemlere hızlı ve kolay çözüm veren yeni çözüm yöntemleri arayışına neden olmuştur. Özellikle sert (hard) optimizasyon teknikleri yerine, yumuşak hesaplama (soft computing) ve evrimsel algoritma (evolutionary algorithm) kullanımı ön plana çıkmıştır. Evrimsel yaklaşımlara dayanan genetik algoritmalar da, bu arayışlar içinde önemli bir yer tutmaya başlamıştır. Uygulama başarıları artan ve sürekli geliştirilmeye çalışılan genetik algoritmalar, kontrol algoritmaları ile birlikte kullanılarak hibrid (hybrid) çözümler sunmaktadır.

Doğal seçim ilkelerine dayanan bir arama ve optimizasyon yöntemi olan genetik algoritmalar, fonksiyon optimizasyonu, çizelgeleme, mekanik öğrenme, tasarım, hücresel üretim gibi alanlarda başarılı uygulamaları bulunmaktadır [51]. Geleneksel optimizasyon yöntemlerine göre farklılıkları olan genetik algoritmalar, parametre kümesi yerine onların kodlanmış biçimlerini kullanırlar. Olasılık kurallarına göre çalışan genetik algoritmalar, yalnızca amaç fonksiyonuna gereksinim duyar. Çözüm uzayının tamamını değil belirli bir kısmını tararlar. Böylece, etkin arama yaparak çok daha kısa bir sürede çözüme ulaşırlar [51].

Yukarıda bahsedilen avantajlarından dolayı genetik algoritmalar öngörülü kontrol tekniklerine uyarlanmış ve parametre kestirimi için kullanılmıştır. Ferragud [52] doğrusal olmayan sistemlerin parametre kestirimi için genetik algoritma kullanmıştır. Fılalı [53] ise genetik algoritma ile GPC algoritmasını birleştirmiş, endüstriyel işlemlerde optimal performansa ulaşmak için ayar parametrelerinin kestiriminde genetik algoritmayı kullanmıştır.

Robot uygulamalarında ise genetik algoritmalar yörünge planlamasında kullanılmıştır. Monteiro [54] 5 eklemli bir robot kolu kontrolü gerçeklemiş, yörünge planlamasını genetik algoritma ile gerçekleştirmiştir. Yanrong Hu [55], mobil bir robotun optimal yolu bulması için genetik algoritmayı kullanmış ve genetik operatörlerin etkisini vurgulamıştır.

Görüldüğü gibi doğrusal öngörü modeli kullanan metotlar için genetik algoritmalar öngörü modelin başarımını iyileştirmektedir. Öte yandan robot kontrolü gibi doğrusal olmayan sistemler için de yapay sinir ağ tabanlı kontrol algoritmaları başarılı olmuştur ve son yirmi yıldır yapay sinir ağları doğrusal olmayan yapıları ile bu alandaki çözümün odak noktası olmuştur [40, 56, 57]. Ayrıca YSA ‘lar doğrusal olmayan sistemlerin öngörülü kontrolü için kontrol süresince oluşabilecek istenmeyen durumlar karşısında daha dengeleyici davranışlar sergilemiş ve kontrol algoritmalarının adaptif özelliklerini arttırmıştır [40].

Kullanılan yapay sinir ağ modelinin yapısı algoritmanın performansını doğrudan etkileyen bir unsurdur. Ağın öğrenme süresi ve öngörü işlemlerinin başarısı bu yapının seçimine bağlıdır. Uzun bir öğrenme süresi ve hatalı öngörü işlemi gerçek-zaman uygulamaları için bir dezavantajdır [58]. Yapay sinir ağ uyarlamalı GPC (NGPC) algoritmalarında şimdiye kadar kullanılan ağ modelleri ileri beslemeli ağ modelleridir. Bu ağlar, öngörülü kontrolün başarımını arttırmalarına karşın ağırlıkların güncellenmesi ve öğrenme süreleri nedeniyle gerçek-zaman uygulamaları için problem teşkil etmektedirler.

Elman ağı gibi, statik ağ yerine dinamik ağların kullanımı hem öğrenme süresini kısaltacak hem de ağın adaptif özelliğini geliştirecektir. Elman ağı diğer ağların

aksine sahip olduğu gizli bağlam katmanı (context hidden layer) ile zaman geciktirilmiş (delaying) giriş ve çıkışları olmaksızın geçmiş bilgileri hatırlayabilmektedir. Dolayısıyla katmanlar arasındaki ağırlık bileşenlerinin ayarlanması diğer ağlara nispeten daha hızlıdır. Bu durum öğrenme kabiliyetini ve ağın performansını doğrudan arttırır.

Elman ağı ilk olarak 1990 yılında zaman serileri için önerilmiştir [59]. Kremer elman ağını diğer ağlar ile karşılaştırmış ve hesaplama gücünü ortaya koymuştur [60]. Sonraki çalışmalarda bu ağ yapısı zaman serilerinin tahmininde kullanılmıştır [61, 62].

Ayrıca dinamik sistemler için Elman ağ tabanlı kontrol algoritmaları tasarlanmış ve ağın öğrenme süresini kısaltarak algoritmaların performansını geliştirdiği vurgulanmıştır [58].

Öğrenme ve öngörme işlemlerinde harcanan süre dinamik sistemlerin gerçek-zaman yapay sinir ağ tabanlı kontrolünde problemlere yol açacağından, daha hızlı öğrenen ağ modellerine ihtiyaç duyulur. Bu ihtiyaç dinamik bir ağ yapısına sahip yinelenen elman ağı ile karşılanabilir. Sonuç olarak, elman ağının öğrenme hızı işlem zamanını kısaltacağından gerçek-zaman uygulamaları için daha etkin olacaktır.

Bu tez çalışmasında, ilk olarak Clarke ve ekibi [28-32] tarafından tanıtılan Genelleştirilmiş Öngörülü Kontrol (GPC: Generalized Predictive Control) ve Soloway [56, 63-65] tarafından tanıtılan Newton-Raphson uyarlamalı Yapay Sinir Ağlı Genelleştirilmiş Öngörülü Kontrol (NGPC: Neural Generalized Predictive Control) algoritmaları incelenmiştir. GPC ‘nin optimizasyonu için Basit Genetik Algoritma Uyarlamalı GPC (SGA-GPC: Simple Genetic Algorithm implemented GPC), NGPC algoritmasında ise ağın performansını iyileştirmek ve öğrenme hızını arttırmak için Yinelenen Elman Ağ Uyarlamalı NGPC (Recuurrent Elman’s Neural Network implemented NGPC-ENGPC) algoritmaları geliştirilmiştir. Algoritmaların her biri SISO ve MIMO olmak üzere iki şekilde tasarlanarak altı eklemli endüstriyel bir robotik manipülatöre yörünge kontrolü için uygulanmıştır. Robotik manipülatörün dinamik modelinde Lagrange-Euler yöntemi kullanılmıştır. Dinamik

modele sürtünme, yük taşıma ve kontrol algoritmalarının adaptif özelliklerinin karşılaştırılması için taşınan yükün taşıma esnasında düşmesi durumları da ilave edilmiştir. Ayrıca, eklem torklarına rasgele bozucular eklenerek kontrol güçleştirilmiş ve bu durumda algoritmaların performansları incelenmiştir. Robot kolunun her bir eklemine ait takip etmesi istenilen konum referans ve hız referans yörüngeleri sinüzoidal yörünge esaslarına göre belirlenmiştir. Gerekli bütün yazılımlar tek bir paket program halinde Borland Delphi 7.0 programlama dili kullanılarak gerçekleştirilmiştir. Tasarlanan kontrol algoritmalarının performansı eklemlere ait tork, açısal yörünge, açısal hız, açısal hız hataları grafikleri ile eklemlere ait açısal konum hataları, açısal hız hatalarının kareleri ve uç nokta konum hataları üzerinden hem grafiksel hem de nümerik sonuçlarla karşılaştırılmıştır. Tasarlanan algoritmaların ENGPC MIMO ve NGPC MIMO hariç, başarılı sonuçlar verdikleri gözlemlenmiştir. NGPC algoritması için önerilen Yinelenen Elman YSA modeli algoritmanın performansını önemli ölçüde geliştirmiştir. Hareket başlangıcı, yük değişimleri ve rasgele bozuculu durumlar gibi belirsizlikler karşısında öngörülerin doğruluğunu arttırmıştır. Ayrıca değişen sistem dinamikleri karşısında daha güçlü bir direnç göstererek hızlı bir adaptasyon sağlamıştır. Sistem oturma zamanını %50 oranında azaltmıştır. GPC algoritmasına uyarlanan Genetik Algoritma, değişken sistem dinamikleri karşısında adaptasyon yeteneği zayıf olmasına rağmen, açı ve hız hataları yönünden oldukça başarılıdır. Özellikle tüm eklemlerin tek bir kontrolör tarafından kontrol edildiği MIMO tasarımda uç nokta koordinat hatalarını önemli ölçüde azaltmıştır.

# BÖLÜM 2. ROBOT KOLU KİNEMATİĞİ VE ALTI EKLEMLİ ROBOT KOLUNUN DİNAMİK MODELİ

Bu bölümde, ilk olarak robot kollarının genel tanıtımı yapılarak robot kolu kinematiği anlatılacaktır. Daha sonra tez çalışmasında gerçeklenen altı eklemli robotik manipülatörün dinamik modeli verilecektir.

## 2.1. Robot Kolları ve Kinematiği

Robot kollarının kinematiği, kolun L-E (Lagrange-Euler) denklemlerinin çıkartılması ve yörünge planlamaları bu bölümde anlatılacaktır.

### 2.1.1. Robotların genel tanıtımı

Robotların şimdiye kadar birçok farklı tanımı yapılmıştır. Sözlükteki karşılığı, "genellikle insanların gerçekleştirdikleri işlevleri yerine getiren otomatik araçlar" olarak tanımlanmaktadır. Ancak bu tanıma göre mesela bir çamaşır makinesi de robot sayılabilmektedir.

Robotun, Amerikan Robot Enstitüsü tarafından yapılan tanımı ise, "malzemelerin, parçaların ve araçların hareket ettirilebilmesi için tasarlanmış olan çok fonksiyonlu ve programlanabilir manipülatör veya farklı görevleri yerine getirebilmek için değişken programlı hareketleri gerçekleştirebilen özel araç" şeklindedir.

Robot bir kaide üzerinde en az bir kol, tutma organları (genellikle pensler, vantuzlar, veya elektromıknatıslar), pinomatik, hidrolik veya elektriksel sensörler ile konumu ve basınç algılayıcılarıyla, bilgi işlem organlarıyla donatılmış kontrollü-mekanik manipülatörlerdir [66]. Sanayi robotunun en kapsamlı tanımı ve robot tiplerinin sınıflandırılması ISO 8373 standardında belirlenmiştir. Bu standarda göre bir robot

şöyle tanımlanır: "Endüstriyel uygulamalarda kullanılan, üç veya daha fazla programlanabilir ekseni olan, otomatik kontrollü, yeniden programlanabilir, çok amaçlı, bir yerde sabit duran veya hareket edebilen manipülatör."

Yukarıdaki tanımlarda da görüldüğü gibi robot; canlılara benzer işlevleri olan ve davranış biçimleri sergileyen makinelerdir. Temel olarak bir robotun üç niteliğe sahip olması gerektiğini belirtmişlerdir: İşlem yapma yetisi; bir işlemi yerine getirebilmelidir, işlemin sonucunu belirleme yetisi; işlem yaptıktan sonra mutlaka işlemin sonucunu belirlemelidir, karar verme yetisi; işlem sonucuna ya da dış etkenlere göre bir yargı kurabilmelidir. Bu yapıları bünyesinde barındıran bir sisteme genel olarak "Robot" adını verebiliriz.

Robotların kullanım alanı daha çok seri üretim gerektiren sahalar, montaj işleri ve nükleer reaktör çekirdekleri gibi insanların çalışmasının mümkün olmadığı veya çok riskli olduğu yerlerdir. Güç gerektiren işlerde, sıcak, soğuk ve tehlikeli ortamlarda kolayca çalışabilmeleri, hızlı, emniyetli, seri, hassas ve ekonomik olmaları, çalışan elemanlar gibi sağlık, emniyet, mola gibi çeşitli ihtiyaçlarının olmayışları, tekrarlı ve monoton işlerin niteliğini değiştirme özellikleri gelişen endüstride robotların etkinliklerini artırmıştır. Günümüzde robotlar fabrika dışında; evlerde, ofislerde, bankalarda, restoranlarda ve yaşamın her sahasında kullanım alanları bulmaktadır.

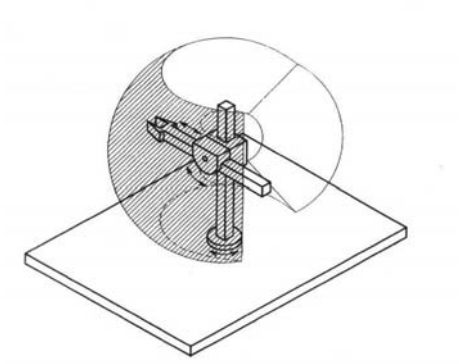
Bildiğimiz manada robotların gelişimi 1950'li yılların başında görülmeye başladı. Bu yıllarda Unimate tarafından tasarlanmış ve endüstriyel robotların ilklerinden olan Unimate 2000 serisi görüldü. Bu robot kolu otomobil birleştirme hattında kullanılmış, sonuçta büyük bir artış ve verimlilik kaydedilmiştir. Aynı yıllarda birleştirme işlerinde kullanılmak amacıyla tasarlanmış programlanabilir evrensel makine olan PUMA robot ailesi görüldü. 5 ve 6 eksenli PUMA-550 ve PUMA-560 bu ailenin en iyi modelleridir. 1970'li yılların başında Cincinatti Milicron firması tarafından bilgisayar kontrollü, genel amaçlı $T^3$ robotu üretildi [40].

Günümüzde Üniversiteler ve birçok ticari kuruluş değişik amaçlar için çeşitli biçimlerde robot kolları üretmektedirler.

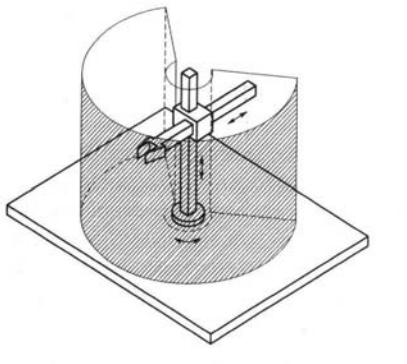
### 2.1.1.1. Robot kol çeşitleri

Çoğu uygulamada robotlar zemine tutturulmuş bir temel üzerine oturtulur. Robotun gövdesi bu temele, kolu da gövdeye tutturulmuştur. Kolun ucunda bilek kısmı vardır. Bilek çok çeşitli hareketleri yapabilecek şekilde birçok parçadan oluşur. Bileğin uç kısmında el vardır. Bu ele uç eleman (end-effecter) adı verilir. Uç eleman robot kolunun kullanım amacına göre farklı şekillerde olabilir. Genel amaçlı robotların özel uygulamalar için kullanılmalarına imkân verir.

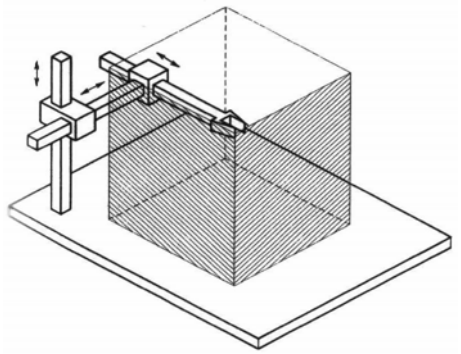
Endüstriyel robotlar değişik tip ve boyutlarda bulunmaktadır. Çeşitli kol hareketlerini yapabilirler ve farklı hareket sistemlerine sahiptirler. Endüstriyel robotlar genel olarak Şekil 2.1'de gösterildiği gibi dört biçimde üretilmektedirler.



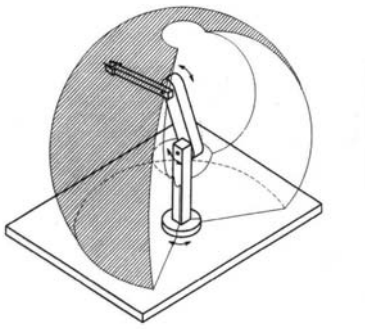
(a) Küresel koordinat biçimi



(b) Silindirik koordinat biçimi



(c) Kartezyen koordinat biçimi



(d) Eklemli kol biçimi

Şekil 2.1. Robot kol biçimleri

Bu dört tip robot kol biçiminin her birinin kendine özgü avantaj ve dezavantajları vardır. Hareketin tekrarlanması bakımından kartezyen koordinat biçim, uzanabilirlik

bakımından eklemli kol biçim, taşıma kapasitesi bakımından ise silindirik koordinat biçim daha avantajlıdır.

#### 2.1.1.2. Robot kolu hareketleri

Endüstriyel robotlar görevlerini gerçekleştirmek için gövde, kol ve bilek olmak üzere bir seri hareket yaparlar. Robot hareketleri genel olarak iki kategoride sınıflandırılır.

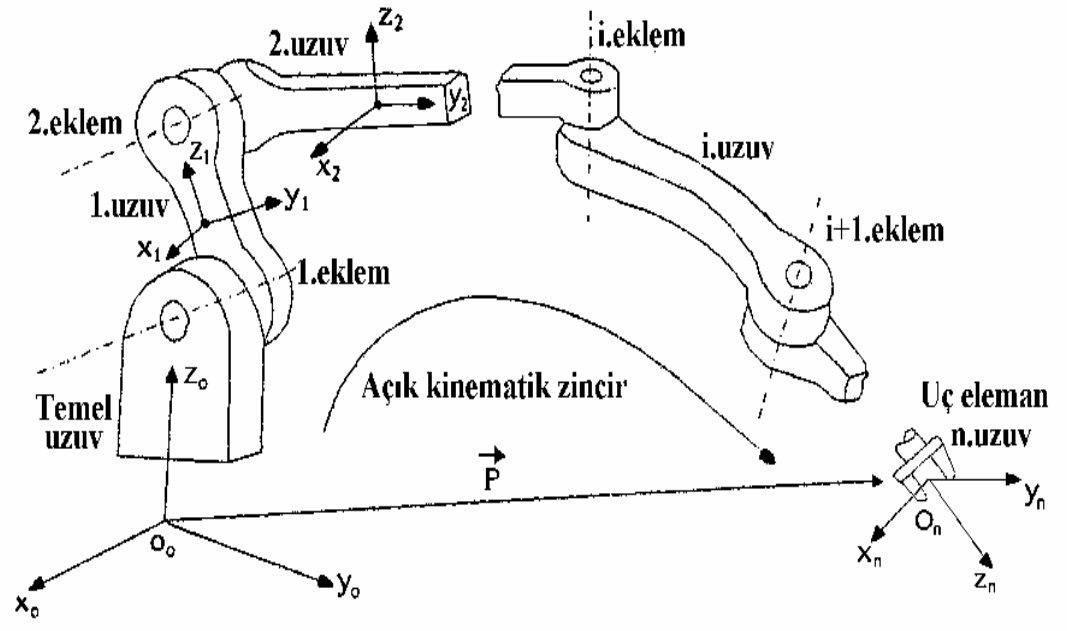
1. Gövde ve kol hareketleri
2. Bilek hareketleri

Bu kategorilerin her bir oynar hareketi o endüstriyel robotun serbestlik derecesini verir. Tipik bir endüstriyel robotun serbestlik derecesi 3 ile 6 arasında bulunur.

Robotların gövde, kol ve bilek hareketleriyle işlem yaptığı sahaya ise iş hacmi denir. Bir endüstriyel robotun iş hacmi onun fiziksel biçimi, boyutu, kol ve mafsal hareketlerinin sınırları gibi özeliklere göre belirlenir. Örneğin, bir kartezyen koordinat robotunun iş hacmi dikdörtgen prizması şeklindedir. Silindirik koordinat robotun iş hacmi silindiriktir. Küresel koordinatlı robot, bir küresel hacim oluşturur. Eklemli kol biçimine sahip bir robotun iş hacmi ise düzensiz bir yapıdadır, kısmen küresel bir hacme sahiptir.

### 2.1.2. Robot kolu kinematiği

Robot kolu, dönel veya kayar eklemlerle birbirine bağlanmış “uzuv” adı verilen eğilmez cisimlerden meydana gelen açık çevrimli bir zincirdir. Çevrimin bir ucu bir desteğe (temele) bağlanmış, diğer ucu ise serbesttir. Eklemler birbirlerine bağlı uzuvların izafi hareketine izin vermektedirler. Bir robot kolunun açık kinematik zinciri Şekil 2.2’de gösterilmiştir.




Şekil 2.2. Açık kinematik zincir

Her bir eklem-uzuv çifti bir serbestlik derecesi oluşturmaktadır. $n$ serbestlik dereceli bir robot kolu, $n$ adet uzuv ve $n$ adet eklemden meydana gelmektedir. Eklemler ve uzuvlar temelden başlayarak dışarıya doğru numaralandırılırlar. Robot kolunun bağlı olduğu temel $(0)$ uzvu, tutucusu (en uç uzuv) ise $(n)$ uzvu olarak isimlendirilir. $(i)$ eklemi $(i-1)$ uzvu ile $(i)$ uzvunu birleştirmektedir.

Kinematik problem, direkt ve ters kinematik problem olmak üzere iki şekilde ele alınmaktadır. Direkt kinematik problemde amaç, bir referans koordinat sistemine göre robot kolu tutucusunun pozisyon ve yönlendirmesini bulmaktır. Robot kolunun eklem değişkenleri vektörü $q = (q_1, q_2, \ldots, q_n)^T$ ve gerekli geometrik uzuv parametreleri bilinmektedir. Ters kinematik problemde ise amaç, robot kolu tutucusunu istenen pozisyon ve yönlendirmeye getirecek eklem değişkenleri vektörünü bulmaktır.

Bir robot kolun kinematik ve dinamik denklemlerini sistematik ve genelleştirilmiş bir şekilde çıkarabilmek için, her bir $(i)$ uzvuna cisim koordinat çerçevesi ( $\vec{x}_i, \vec{y}_i, \vec{z}_i$ ) yerleştirilir. Komşu koordinat çerçeveleri arasında Denavit ve Hartenberg [67]

tarafından geliştirilen dört parametre yardımı ile ilişki kurulmaktadır (Şekil 2.3). Bu parametreler aşağıdaki şekilde tanımlanmaktadır.

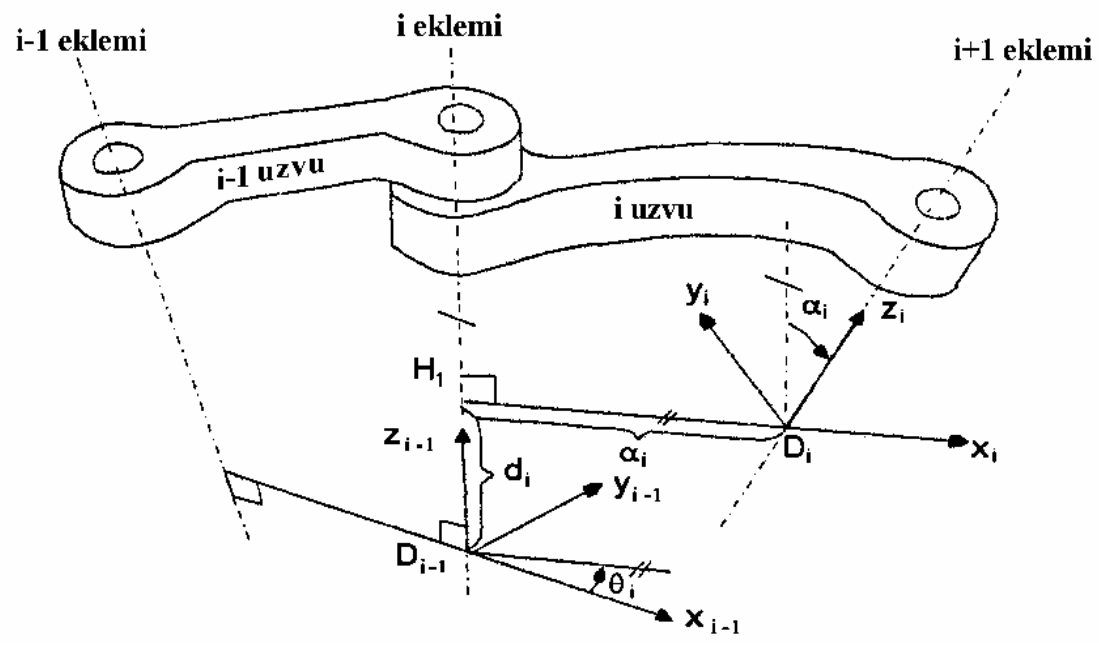
$\theta_i$ : $\vec{z}_{i-1}$ ekseni etrafında $\vec{x}_{i-1}$ ekseninden $\vec{x}_i$ eksenine kadar olan eklem açısı (sağ el kuralı kullanılarak),

$d_i$ : $\vec{z}_{i-1}$ ekseni boyunca $(i-1)$.koordinat çerçevesinin orijininden $\vec{x}_i$ ekseni ile $\vec{z}_{i-1}$ ekseninin kesişme yerine kadar olan mesafe,

$a_i$ : $\vec{x}_i$ ekseni boyunca $\vec{x}_i$ ekseni ile $\vec{z}_{i-1}$ ekseninin kesişme noktasından $(i)$.koordinat çerçevesinin orijinine kadar olan mesafe ($\vec{z}_{i-1}$ ve $\vec{z}_i$ eksenleri arasındaki en kısa mesafe),

$\alpha_i$ : $\vec{x}_i$ ekseni etrafında $\vec{z}_{i-1}$ ekseninden $\vec{z}_i$ eksenine kadar olan açı (sağ el kuralı kullanılarak)

Dönel eklem için $d_i$, $a_i$, $\alpha_i$ parametreleri sabit, $\theta_i$ parametresi değişkendir ve $(i)$ uzvu $(i-1)$ uzvuna göre döndüğünde değişmektedir. Kayar eklem için ise $\theta_i$, $a_i$, $\alpha_i$ parametreleri sabit, $d_i$ parametresi değişkendir.




Şekil 2.3. Denavit-Hartenberg notasyonu

Her bir koordinat çerçevesi aşağıdaki üç temel kurala göre yerleştirilmektedir [68].

1. $\vec{z}_{i-1}$ ekseni $(i)$.eklemin hareket ekseni boyunca yerleştirilir,
2. $\vec{x}_i$ ekseni $\vec{z}_{i-1}$ eksenine diktir (ucu dışarıya doğru olacak şekilde),
3. $\vec{y}_i$ ekseni sağ el kuralına göre koordinat sistemini tamamlayacak şekildedir.

Yukarıdaki kurallara uyularak $(\vec{x}_0, \vec{y}_0, \vec{z}_0)$ referans çerçevesi, $\vec{z}_0$ ekseni ilk eklemin hareket ekseni boyunca olacak şekilde temel uzuv üzerinde herhangi bir yere yerleştirilebilir. Son koordinat çerçevesi ($n$.çerçeve), $\vec{x}_n$ ekseni $\vec{z}_{n-1}$ eksenine dik olacak şekilde el üzerinde herhangi bir yere yerleştirilebilir.

$(i)$ uzvunun koordinat sisteminde tanımlanan herhangi bir vektörü, $(i-1)$ uzvunun koordinat sistemine taşımak için $A_{i-1}^{i}$ homojen dönüşüm matrisi kullanılmaktadır. Denavit-Hartenberg parametrelerini kullanarak $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$ koordinat sistemde tanımlamayı gerçekleştiren homojen dönüşüm matrisi aşağıdaki şekilde verilmektedir.

$$A_{i-1}^{i} = \begin{bmatrix} Cos\theta_i & -Cos\alpha_i\ Sin\theta_i & Sin\alpha_i\ Sin\theta_i & a_i\ Cos\theta_i \\ Sin\theta_i & Cos\alpha_i\ Cos\theta_i & -Sin\alpha_i\ Cos\theta_i & a_i\ Sin\theta_i \\ 0 & Sin\alpha_i & Cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

$A_{i-1}^{i}$ matrisinin tersi $([A_{i-1}^{i}]^{-1} = A_i^{i-1})$, herhangi bir vektörün $(i-1)$.çerçeveden $(i)$.çerçeveye dönüşümü için kullanılmaktadır. Herhangi bir vektörün koordinatlarını $(i)$ uzvunun koordinat sisteminden temel koordinat sistemine dönüştürecek homojen dönüşüm matrisi, dönüşüm matrislerinin ardışık olarak çarpılması ile elde edilir.

$$A_0^i = A_0^1\ A_1^2 \cdots A_{i-1}^i = \prod_{j=1}^{i} A_{j-1}^j \tag{2.2}$$

$A_{i-1}^{i}$ homojen dönüşüm matrisinin sol üst tarafında bulunan $3x3$ 'lük alt matris, rotasyon matrisini simgelemektedir. Rotasyon matrisi, bir vektörün koordinatlarını bir koordinat sisteminden diğerine orijinleri aynı kalacak fakat bir miktar dönecek şekilde dönüştürmektedir. $(i)$ uzvunun koordinatlarına dönüşümü sağlayan rotasyon matrisi,

$$R_{i-1}^{i} = \begin{bmatrix} Cos\theta_i & -Cos\alpha_i\ Sin\theta_i & Sin\alpha_i\ Sin\theta_i \\ Sin\theta_i & Cos\alpha_i\ Cos\theta_i & -Sin\alpha_i\ Cos\theta_i \\ 0 & Sin\alpha_i & Cos\alpha_i \end{bmatrix} \tag{2.3}$$

şeklinde verilmektedir. Bu dönüşüm ortonormal bir dönüşüm olduğundan rotasyon matrisinin tersi transpozesine eşittir ( $R_i^{i-1} = (R_{i-1}^{i})^T$ ).

( $\vec{x}_0, \vec{y}_0, \vec{z}_0$ ) referans çerçevesi ve homojen dönüşüm matrisi $A_0^i$ verilirse, rotasyon alt matrisinin sütun vektörleri referans çerçeveye göre ( $\vec{x}_i, \vec{y}_i, \vec{z}_i$ ) koordinat sisteminin asal eksenlerini göstermektedir. Homojen dönüşüm matrisinin 4.sütunu, referans çerçeveye göre $(i)$ .çerçevenin orijininin pozisyonunu göstermektedir.

$$A_0^i = \begin{bmatrix} \vec{x}_i & \vec{y}_i & \vec{z}_i & \vec{p}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

Denklem 2.4'de $(i)$ yerine $(n)$ konularak elde edilen matris, referans çerçeveye göre elin pozisyon ve yönlendirmesini tamamen belirleyen kol matrisi olarak isimlendirilir.

$$H = A_0^n = \begin{bmatrix} \vec{x}_n & \vec{y}_n & \vec{z}_n & \vec{p}_n \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \vec{n} & \vec{s} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

Burada;

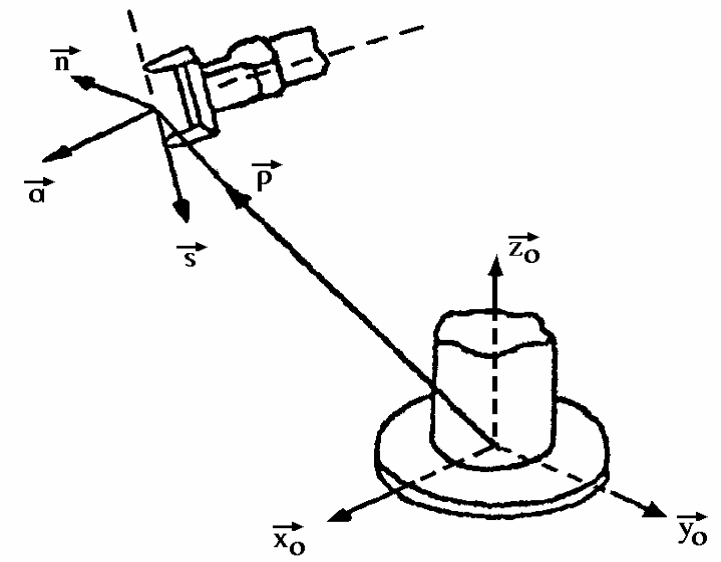
$\vec{n}$ : elin birim normal vektörü,

$\vec{s}$ : elin birim kayma vektörü,

$\vec{a}$ : elin birim yaklaşma vektörü,

$\vec{p}$ : elin pozisyon vektörüdür.

Bu vektörlerin hepsi, Şekil 2.4'de görüldüğü gibi temel koordinatlara göre tanımlanmıştır. (2.2) ifadesi kullanılarak, verilen eklem koordinatları ile robot kolu tutucusunun kartezyen pozisyon ve yönlendirmesini elde etmek mümkündür. Bu işlem "direkt kinematik çözüm" olarak adlandırılmaktadır.




Şekil 2.4. Elin pozisyon ve yönlendirme vektörleri

Robot kolunun kontrolü bazen ters kinematik çözüm gerektirebilir. Ters kinematik probleminin çözümü için elin istenen pozisyon ve yönlendirmesinin verilip buna karşılık gelen eklem değişkenleri vektörünün bulunması gerekmektedir. Düz kinematik için tek bir çözüm var iken, ters kinematik için birden fazla çözüm bulunabilir. Bu da ters kinematik çözüm problemini zorlaştırmaktadır [40].

### 2.1.3. Robot kolunun dinamik modeli

Robot kolu tasarımı ve kontrolü için atılacak ilk adım, robot kolunun dinamik modelinin çıkartılmasıdır. Üçten az serbestlik derecesine sahip robot kollarının dinamik denklemleri el ile çıkartılabilir. Üç veya daha fazla serbestlik derecesine sahip robot kollarının dinamik denklemlerini el ile çıkarmak oldukça zor olmaktadır.

Bu yüzden dinamik denklemleri, otomatik olarak oluşturacak bir algoritma gerekmektedir. Bu algoritma, hesaplamalar el ile yapıldığı zaman oluşan hataları da yok edecektir. Robot kolu modellemede etkili bir metot bulmak için aşağıdaki işlemler göz önüne alınmalıdır.

1. Hareket denklemlerini çıkarmak için kullanılacak algoritma çok karmaşık olmamalı, kolayca formüle edilebilmeli.

2. Model, gerçek sistem ile ilgili sonuçları doğru olarak vermeli.

3. Sistem denklemleri, çevrim içi kontrol ve hesapsal verimlilik açısından kısa zamanda çözülebilir olmalı.

4. Kullanılan metot, dinamiğin hem düz hem de ters problemlerini çözebilmeli. Yani, mekanizma parçalarının hareketi verildiğinde gerekli torklar, torklar verildiğinde ise ivmeler hesaplanabilmeli.

5. Kullanılan metot genel olmalı. Giriş olarak sadece sistem parametreleri verildiğinde, sistemin çalışması için gerekli tüm bilgiler elde edilmeli.

6. Dinamik model, sistemin tüm kısıtlarını göz önüne almalı.

Robot kolu dinamiğini formüle edebilmek için Lagrange-Euler (L-E) [68, 69], Recursive-Lagrange (R-L) [70], Newton-Euler (N-E) [71] ve Genelleştirilmiş D'Alambert Prensibi (G-D) [72] gibi yaklaşımlar önerilmiştir. Bu yaklaşımlar içinde L-E ve N-E yöntemleri en iyileridir. Bu tez çalışmasında, robot kolunun dinamik modelinin çıkarılmasında N-E yöntemine göre daha basit ve sistematik olan L-E yöntemi kullanılmıştır.

#### 2.1.3.1. Hareketin L-E denkleminin çıkarılması

Serbestlik derecesi $n$ olan bir robot kolunun hareket denklemlerinin çıkarılmasında aşağıdaki bilgiler temel alınacaktır.

1. 4x4'lük homojen koordinat dönüşüm matrisi ( $A_{i-1}^{i}$ )
2. Hareketin L-E denklemleri

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right)-\frac{\partial L}{\partial q_i}=\tau_i, \quad i=1, 2, \cdots, n \tag{2.6}$$

Burada;

L : Lagrangian fonksiyonu (L) = Kinetik Enerji (K) - Potansiyel Enerji (P),

K : Robot kolunun toplam kinetik enerjisi,

P : Robot kolunun toplam potansiyel enerjisi,

$q_i$ : Robot kolunun genelleştirilmiş koordinatları

(Dönel eklem durumunda $q_i = \theta_i$, kayar eklem durumunda $q_i = d_i$),

$\dot{q}_i$ : Genelleştirilmiş koordinat $q_i$'nin birinci türevi,

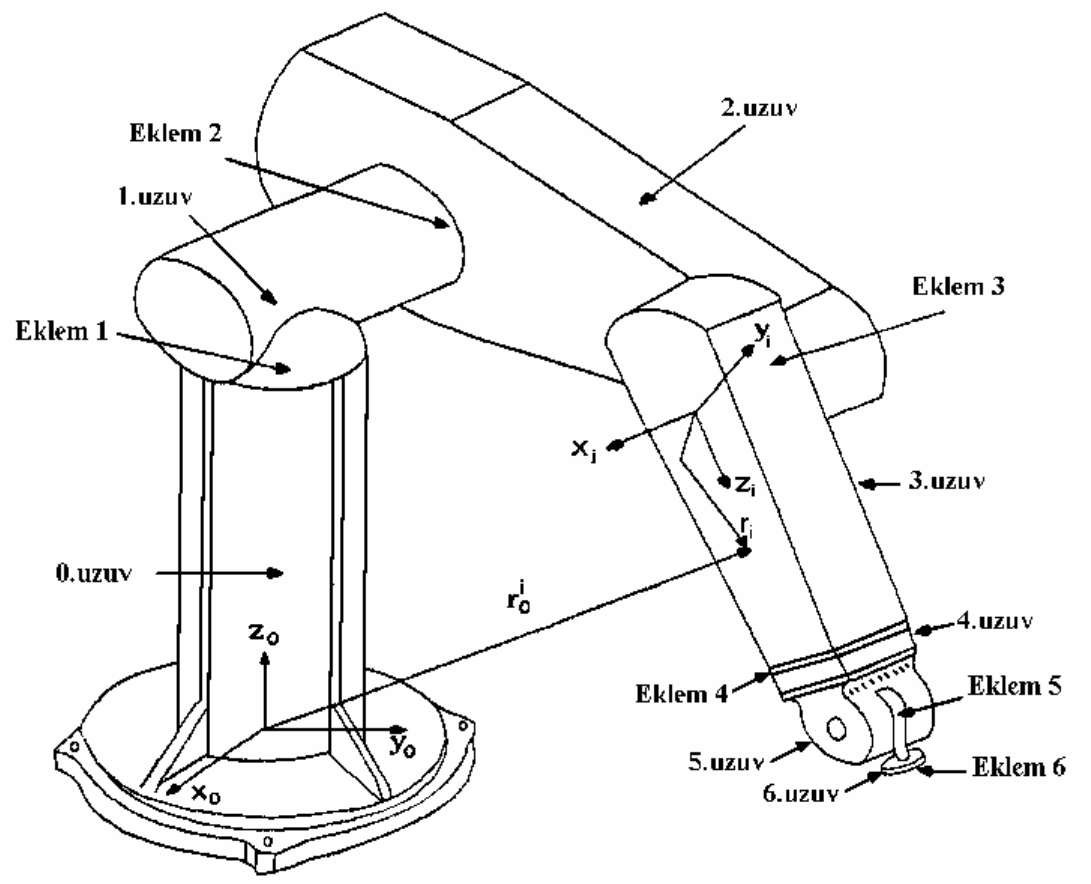
$\tau_i$ : $(i)$ uzvunu hareket ettiren $(i)$ ekleminden sisteme uygulanan tork.

### 2.1.3.2. Bir robot kolunun hızı

Hareketin L-E denklemlerini yazabilmek için sistemin toplam kinetik enerjisinin bilinmesi gerekir. Bu, her bir eklemin hızının bilinmesi demektir. Bu bölümde $(i)$ uzvundaki bir noktanın hızı ve bu uzuvdaki tüm noktalara diğer eklem hareketlerinin etkileri çıkartılacaktır.

$r_i$, $(i)$ koordinat çerçevesine göre homojen koordinatlarda tanımlanmış ve $(i)$ uzvunda sabit bir noktayı göstermek için kullanılmıştır (Şekil 2.5).

$$r_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = (x_i, y_i, z_i, 1)^T \tag{2.7}$$




Şekil 2.5. $(i)$ uzvundaki $r_i$ noktası

$(i)$.koordinat çerçevesine göre tanımlanan ve $(i)$ uzvunda bulunan diğer noktalar gibi $r_i$ noktası da $(i)$ koordinat çerçevesine göre sıfır hıza sahip olacaktır. Temel koordinat çerçevesinde $r_i$ 'nin hızı;

$$V_i = \frac{dr_0^i}{dt} \tag{2.8}$$

şeklinde hesaplanmaktadır. Burada $r_0^i$, temel koordinat çerçevesine göre tanımlanan aynı $r_i$ noktasıdır. Homojen dönüşüm matrisleri $A_{i-1}^i$ ve $A_0^i$ kullanılarak $r_0^i$ aşağıdaki şekilde ifade edilebilir.

$$r_0^i = A_0^i \, r_i \tag{2.9}$$

$A_0^i$ matrisinin elemanları $\theta_i$, $d_i$, $a_i$, $\alpha_i$'nin fonksiyonudur. $q_i$ dönel eklem için $\theta_i$, kayar eklem için ise $d_i$ olacaktır. Kayar ve döner eklemlerin her ikisi içinde uygulanabilecek hareket denklemlerini çıkarabilmek için ($i$) ekleminin genelleştirilmiş koordinatlarını ifade eden $q_i$ değişkeni kullanılacaktır.

Temel koordinat sistemi referans alınarak $r_i$ noktasının mutlak hızı aşağıdaki gibi yazılabilir.

$$V_0^i = V_i = \frac{d}{dt}(r_0^i) = \frac{d}{dt}(A_0^i r_i)$$

$$= \dot{A}_0^1 A_1^2 \cdots A_{i-1}^i r_i + A_0^1 \dot{A}_1^2 \cdots A_{i-1}^i r_i + \cdots\cdots + A_0^1 A_1^2 \cdots \dot{A}_{i-1}^i r_i + A_0^i \dot{r}_i \qquad (2.10)$$

$$= \left( \sum_{j=1}^{i} \frac{\partial A_0^i}{\partial q_j} \dot{q}_j \right) r_i \qquad i = 1,2,\cdots,n$$

$A_0^i$'nın $q_j$'ye göre kısmi türevi aşağıda tanımlanan $Q_i$ matrisi yardımıyla kolayca hesaplanabilir.

Dönel eklem için

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (2.11)$$

veya kayar bir eklem için

$$Q_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (2.12)$$

Böylece;

$$\frac{\partial A_{i-1}^{i}}{\partial q_i} = Q_i \, A_{i-1}^{i} \tag{2.13}$$

elde edilir. Buradan da;

$$\frac{\partial A_{0}^{i}}{\partial q_j} = \begin{cases} A_0^1 \, A_1^2 \cdots A_{j-2}^{j-1} \, Q_j \, A_{j-1}^{j} \cdots A_{i-1}^{i} & j \le i \\ 0 & j > i \end{cases} \qquad i = 1, 2, \cdots, n \tag{2.14}$$

yazılabilir. (2.14) ifadesi $(i)$ uzvundaki tüm noktalara $(j)$ eklem hareketinin etkisi olarak yorumlanabilir.

İfadeyi basitleştirmek için $U_{ij} \cong \partial A_0^i / \partial q_j$ tanımı kullanılırsa,

$$U_{ij} = \begin{cases} A_0^{j-1} \, Q_j \, A_{j-1}^{i} & j \le i \\ 0 & j > i \end{cases} \qquad i = 1, 2, \cdots, n \tag{2.15}$$

Bu ifade kullanılarak $\mathrm{V_i}$ ifadesi $i = 1, 2, \cdots, n$ için

$$V_i = \sum_{j=1}^{i} U_{ij} \, \dot{q}_j \, r_i \tag{2.16}$$

olarak elde edilir. Eklemler arasındaki etkileşim etkilerini bulmak için

$$\frac{\partial U_{ij}}{\partial q_k} \cong U_{ijk} = \begin{cases} A_0^{j-1} \, Q_j \, A_{j-1}^{k-1} \, Q_k \, A_{k-1}^{i} & i \ge k \ge j \\ A_0^{k-1} \, Q_k \, A_{k-1}^{j-1} \, Q_j \, A_{j-1}^{i} & i \ge j \ge k \\ 0 & i < j, \quad i < k \end{cases} \tag{2.17}$$

ifadesinden faydalanılacaktır. Örneğin $i = j = k = 1$ ve $q_1 = \theta_1$ olan PUMA robot kolu için,

$$U_{111} = \frac{\partial U_{11}}{\partial \theta_1} = \frac{\partial (A_0^0 Q_1 A_0^1)}{\partial \theta_1} = A_0^0 Q_1 Q_1 A_0^1 \tag{2.18}$$

olarak elde edilir. Buradaki $A_0^0$ 4x4'lük birim matristir. (2.17) ifadesi, $(i)$ uzvundaki tüm noktalara $(j).$ ve $(k).$ eklem hareketlerinden olan etkileri göstermektedir.

### 2.1.3.3. Bir robot kolun kinetik enerjisi

Her bir uzvun eklem hızları elde edildikten sonra, şimdi de $(i)$ uzvunun kinetik enerjisi bulunacaktır. $K_i$, $(i)$ uzvunun $(i = 1, 2, \cdots, n)$ temel koordinat sistemine göre tanımlanan kinetik enerjisi olsun. $m$ kütleli bir parçacığın kinetik enerjisi $\frac{1}{2} m V^2$ olduğundan, $(i)$ uzvundaki kütlesi $dm$ olan bir parçacığın kinetik enerjisi,

$$dK_i = \frac{1}{2} (\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2)\, dm = \frac{1}{2} Tr(V_i V_i^T)\, dm \tag{2.19}$$

Not : $Tr(A) = \sum_{i=1}^{n} a_{ii}$

(2.16)'da verilen $V_i$ ifadesi (2.19)'da yerine konursa,

$$\begin{aligned} dK_i &= \frac{1}{2} Tr\left( \sum_{p=1}^{i} U_{ip}\, \dot{q}_p\, r_i \left( \sum_{r=1}^{i} U_{ir}\, \dot{q}_r\, r_i \right)^T dm \right) \\ &= \frac{1}{2} Tr\left( \sum_{p=1}^{i} \sum_{r=1}^{i} U_{ip} \left( \int r_i\, r_i^T dm \right) U_{ir}^T\, \dot{q}_p\, \dot{q}_r \right) \end{aligned} \tag{2.20}$$

elde edilir. $U_{ij}$ ve $q_i$ 'ler $(i)$ uzvunun kütlesel dağılımından bağımsız olduklarından, $(i)$ uzvundaki tüm parçacıkların kinetik enerjilerinin toplamı, tümlevler parantez içine alınarak,

$$K_i = \int dK_i = \frac{1}{2} Tr\left( \sum_{p=1}^{i} \sum_{r=1}^{i} U_{ip} \left( \int r_i \, r_i^T \, dm \right) U_{ir}^T \, \dot{q}_p \, \dot{q}_r \right) \tag{2.21}$$

şeklinde elde edilir. Parantez içindeki tümlev terimi $(i)$ koordinat çerçevesine göre $(i)$ uzvundaki tüm parçacıkların ataletidir ve genel ifadesi aşağıdaki gibidir.

$$J_i = \int r_i \, r_i^T \, dm = \begin{bmatrix} \int x_i^2 \, dm & \int x_i \, y_i \, dm & \int x_i \, z_i \, dm & \int x_i \, dm \\ \int x_i \, y_i \, dm & \int y_i^2 \, dm & \int y_i \, z_i \, dm & \int y_i \, dm \\ \int x_i \, z_i \, dm & \int y_i \, z_i \, dm & \int z_i^2 \, dm & \int z_i \, dm \\ \int x_i \, dm & \int y_i \, dm & \int z_i \, dm & \int dm \end{bmatrix} \tag{2.22}$$

$I_{ij}$ atalet gerilimidir ve

$$I_{ij} = \int \left( \delta_{ij} \left[ \sum_k x_k^2 \right] - x_i x_j \right) dm \tag{2.23}$$

ifadesindeki $i, j, k$ değişkenleri $(i)$ koordinat çerçevesinin asal eksenlerini ve $\delta_{ij}$ ise Kronecker deltayı göstermektedir. ( $i = j$ için $\delta_{ij} = 1$ ve $i \neq j$ için $\delta_{ij} = 0$ 'dır ).

$$J_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \tag{2.24}$$

veya $(x_i, y_i, z_i)$ koordinat sisteminde $m_i$ eğilmez cisminin jirasyon yarıçapları kullanılarak $J_i$ atalet matrisi aşağıdaki şekilde ifade edilebilir. Burada $k_{ijk}$, $j-k$ ekseni etrafında $(i)$ uzvunun jirasyon yarıçapını göstermektedir. $\bar{r}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i, 1)^T$, $(i)$ koordinat çerçevesine göre $(i)$ uzvunun kütle merkezinin koordinatlarıdır.

$$J_i = m_i \begin{bmatrix} \frac{-k_{i11}^2 + k_{i22}^2 + k_{i33}^2}{2} & k_{i12}^2 & k_{i13}^2 & \bar{x}_i \\ k_{i12}^2 & \frac{k_{i11}^2 - k_{i22}^2 + k_{i33}^2}{2} & k_{i23}^2 & \bar{y}_i \\ k_{i13}^2 & k_{i23}^2 & \frac{k_{i11}^2 + k_{i22}^2 - k_{i33}^2}{2} & \bar{z}_i \\ \bar{x}_i & \bar{y}_i & \bar{z}_i & 1 \end{bmatrix} \tag{2.25}$$

$J_i$, $(i)$ uzvunun kütlesel dağılımına bağlıdır. Uzuvların pozisyonlarına ve hareket oranına bağlı değildir ve $(i)$.koordinat çerçevesine göre tanımlanmıştır. Bu yüzden $J_i$, herhangi bir robot kolunun kinetik enerjisinin belirlenmesinde sadece bir kez hesaplanmaktadır. Tüm uzuvlar için kinetik enerjilerin toplamı robot kolunun toplam kinetik enerjisi $K$'yı verecektir.

$$K = \sum_{i=1}^{n} K_i = \frac{1}{2} \sum_{i=1}^{n} Tr\left( \sum_{p=1}^{i} \sum_{r=1}^{i} U_{ip} J_i U_{ir}^T \dot{q}_p \dot{q}_r \right)$$

$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{p=1}^{i} \sum_{r=1}^{i} \left[ Tr\left( U_{ip} J_i U_{ir}^T \right) \dot{q}_p \dot{q}_r \right] \tag{2.26}$$

$$= \textit{Skalar bir büyüklük}$$

### 2.1.3.4. Bir robot kolun potansiyel enerjisi

Bir robot kolunun toplam potansiyel enerjisi $P$ ve uzuvların her birinin potansiyel enerjisi $P_i$ olarak kabul edilirse

$$P_i = -m_i\, g\, r_0^i = -m_i\, g\,(A_0^i\, \bar{r}_i)\,, \qquad i = 1,2,\ldots,n \tag{2.27}$$

Robot kolunun toplam potansiyel enerjisi;

$$P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} -m_i\, g\,(A_0^i\, \bar{r}_i) \tag{2.28}$$

dir. Burada $g = (g_x, g_y, g_z, 0)$ temel koordinat sisteminde tarif edilmiş yerçekimi ivme vektörüdür. $z_0$ ekseni dünya yüzeyine dik olacak şekilde yönü yukarı doğru alınırsa yerçekimi ivme vektörü $g = (0, 0, -|g|, 0)$ olur. Deniz seviyesindeki bir sistem için $|g| = 9.8062\ m/sn^2$.

### 2.1.3.5. Hareketin L-E denklemleri

(2.26) ve (2.28) ifadelerini kullanarak Lagrangian fonksiyonu $L = K - P$ aşağıdaki şekilde yazılır.

$$L = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=1}^{i}\left[Tr\left(U_{ij}\, J_i\, U_{ik}^T\right)\dot{q}_j\, \dot{q}_k\right] + \sum_{i=1}^{n} m_i\, g\,(A_0^i\, \bar{r}_i) \tag{2.29}$$

(2.6)'da verilen Lagrange-Euler ifadesi (2.29)'a uygulanarak bir robot kolun hareketine ait dinamik denklemler elde edilir.

$$\begin{aligned}\tau_i = &\sum_{j=1}^{n}\sum_{k=1}^{j} Tr\left(U_{jk}\, J_j\, U_{ji}^T\right)\ddot{q}_k + \sum_{j=1}^{n}\sum_{k=1}^{j}\sum_{m=1}^{j} Tr\left(U_{jkm}\, J_j\, U_{ji}^T\right)\dot{q}_k\, \dot{q}_m \\ &- \sum_{j=1}^{n} m_j\, g\, U_{ji}\, \bar{r}_j\,; \qquad\qquad i = 1,2,\ldots,n\end{aligned} \tag{2.30}$$

Yukarıdaki ifade daha basit olarak

$$\tau_i = \sum_{k=1}^{n} D_{ik}\, \ddot{q}_k + \sum_{k=1}^{n}\sum_{m=1}^{n} H_{ikm}\, \dot{q}_k\, \dot{q}_m + G_i \qquad i = 1,2,\ldots,n \tag{2.31}$$

şeklinde veya matris formunda

$$\tau(t) \;=\; D\big(q(t)\big)\,\ddot{q}(t) \;+\; H\big(q(t),\dot{q}(t)\big) \;+\; G\big(q(t)\big) \tag{2.32}$$

şeklinde gösterilir. Burada $D\big(q(t)\big)$, ivmelenme ile ilgili atalet matrisidir.

$$D_{ik} \;=\; \sum_{j=\max(i,k)}^{n} Tr\big(U_{jk}\, J_j\, U_{ji}^{T}\big) \qquad k = 1,2,\ldots,n \tag{2.33}$$

$H\big(q(t),\dot{q}(t)\big)$, $nx1$'lik doğrusal olmayan Coriolis ve merkezkaç kuvvet vektörüdür.

$$H\big(q(t),\dot{q}(t)\big) \;=\; \big(H_1, H_2, \ldots, H_n\big)^T \tag{2.34.a}$$

$$H_i \;=\; \sum_{k=1}^{n}\sum_{m=1}^{n} H_{ikm}\,\dot{q}_k\,\dot{q}_m \qquad i = 1,2,\ldots,n \tag{2.34.b}$$

$$H_{ikm} \;=\; \sum_{j=\max(i,k,m)}^{n} Tr\left(U_{jkm}\, J_j\, U_{ji}^{T}\right) \qquad i,k,m = 1,2,\ldots,n \tag{2.34.c}$$

$G\big(q(t)\big)$ $nx1$'lik yerçekimi kuvveti vektörüdür.

$$G\big(q(t)\big) \;=\; \big(G_1, G_2, \ldots, G_n\big)^T \tag{2.35.a}$$

$$G_i \;=\; \sum_{j=i}^{n} (-m_j\, g\, U_{ji}\, \bar{r}_j) \qquad i = 1,2,\ldots,n \tag{2.35.b}$$

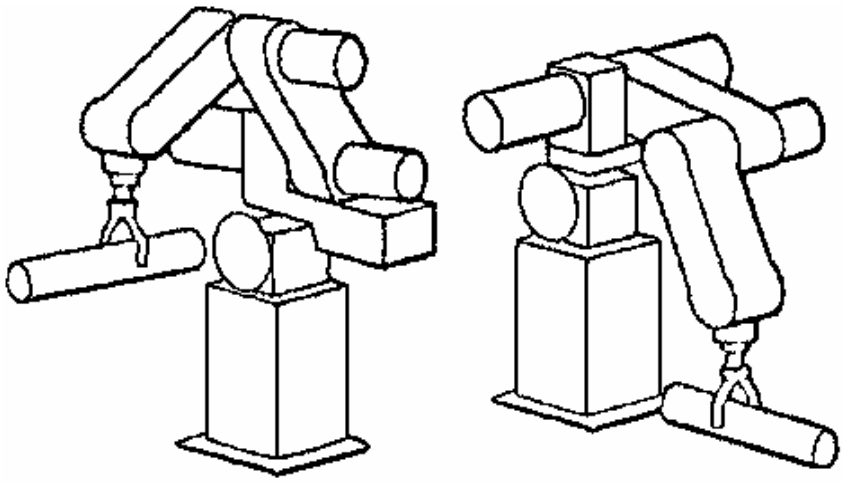
$\tau$, $nx1$'lik eklem aktuatörlerine uygulanan tork / kuvvet'i göstermektedir.

Bu tez çalışmasında yukarıda anlatılan Lagrange-Euler denklemleri kullanılarak robot kolunun dinamik modeli oluşturulmuştur. Dinamik modellemeye sürtünme, yük taşıma ve taşınan yükün taşıma esnasında düşmesi gibi durumlar da ayrıca ilave edilmiştir.

### 2.1.4. Yörünge Planlaması

Robot kolunun hareketi, istasyon (temel koordinat sistemi) eksenlerine bağlı takım eksenlerinin hareketleri toplamı olarak düşünülebilir. Bu şekilde tasarlanan yörünge planlaması hem kullanıcının düşünce tarzına daha uygun hem de bu belirlenen yol tasarımı önemli avantajlar sağlamaktadır [40]. İstasyon eksenlerine bağlı takım eksenlerinin hareketleri belirlendiğinde robot kolunun hareketi uç elemanın hareketi ile rahatlıkla birleştirilebilir. Bu durum robot koluna farklı uç elemanları ile kullanılabilme olanağı sağlar.

Yörünge planlamasında amaç, Şekil 2.6'da görüldüğü gibi robot kolunu bir başlangıç pozisyonundan istenen bir son pozisyonuna hareket ettirmektir. Robot kolunun yörünge planlaması, eksenlere göre veya eklem açılarına göre olmak üzere iki şekilde ele alınabilir. Eklemlere uygulanan torkların eklem açıları ile olan dinamiksel ilişkisi koordinat eksenleri ile olan dinamiksel ilişkisinden daha az karmaşıktır. Bu yüzden robot kolunun hareketini açılar cinsinden ifade etmek ve yörüngeleri açılara göre planlamak robot kolunun kontrolünü kolaylaştıracaktır.



Şekil 2.6. Robot kolunun başlangıç ve bitiş pozisyonu

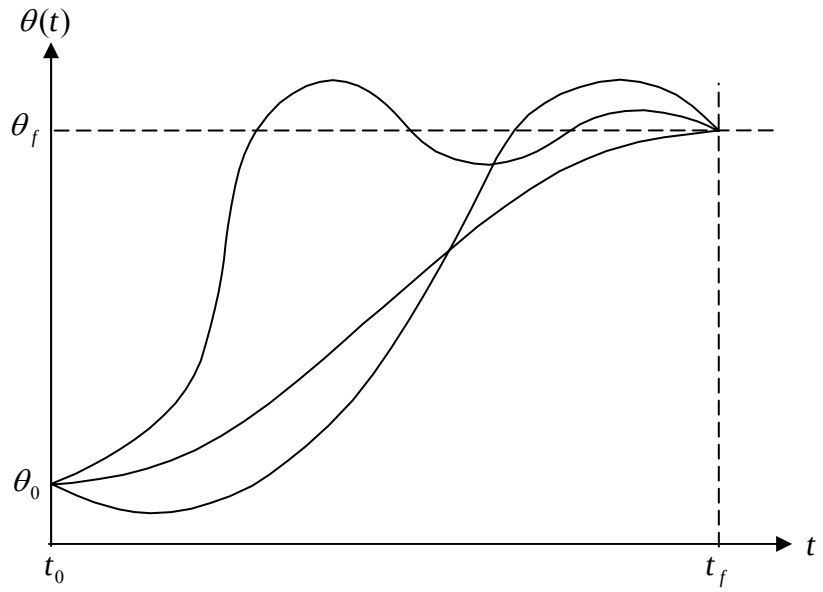
Hareketi eklem açıları cinsinden ifade edebilmek için her bir eklem üzerine bir koordinat çerçevesi yerleştirilir. Bu koordinat çerçeveleri birbirleri cinsinden kolayca ifade edilebilecek şekilde olmalıdır. Bu sayede her bir eklemin hareketi diğer

eklemlerden bağımsız olarak formüle edilebildiği gibi aynı zamanda bu hareketler birleştirilerek kolayca robot kolunun hareketi de çıkartılabilir.

### 2.1.4.1. Yörünge planlamada fonksiyon kullanımı

Yol tanımlamada detayları içine alan bir yöntem; noktalar yoluyla istenen hareket sırasını göstermek yani başlangıç noktasından bitiş noktasına kadar yörüngeyi ara noktalarla temsil etmektir. Hareketin sarsıntısız ve düzenli olabilmesi için bu ara noktalar uygun bir düzgünleştirme fonksiyonunu takip edecek şekilde seçilmelidir. Noktalar denmesine rağmen bunlar pozisyon ve yönelmeyi gösteren koordinatlardır.

Sabit bir zaman aralığında robot kolunu bir başlangıç konumdan bir hedef konuma getirme problemini göz önüne alalım. Sistem kinematiği kullanılarak başlangıç ve hedef konuma karşılık gelen eklem açıları hesaplanabilir. Burada istenen, sabit bir zaman aralığında eklem başlangıç konumu ile hedef konumu arasında her bir eklem için zamana bağlı bir $\theta(t)$ fonksiyonu elde etmektir. Bu amaçla kullanılabilecek bir çok düzgün fonksiyon mevcuttur (Şekil 2.7).




Şekil 2.7. Her bir eklem için uygun yörünge planları

Robot kolu düzgün bir hareket yaparken eklemlere ait $\theta(t)$ fonksiyonu üzerinde en az dört sınır değerin olması gerekmektedir. İki sınır değer, başlangıç ve bitiş pozisyonlarının belirlenmesinden gelmektedir.

$$\theta(0) = \theta_0, \qquad \theta(t_f) = \theta_f$$

İlave iki sınır değer ise fonksiyonun açısal hız bakımından sürekli olmasından gelir. Başlangıç ve bitiş pozisyonundaki açısal hızlar sıfırdır.

$$\dot{\theta}(0) = 0, \qquad \dot{\theta}(t_f) = 0$$

Robot koluna birbirini takip edecek şekilde peş peşe birden fazla hareket yaptırmak istenirse hareketlerin toplamı tek bir hareketmiş gibi düşünülebilir. Bu durumda hareketlerin birleşme noktalarına ara noktalar denir. Ara noktalarda açısal hızlar 0 olmak zorunda değildir. Fakat hareketin sarsıntısız olması için bu ara noktalarda hız ve ivme sürekliliği aranır.

### 2.1.4.2. Kübik yörünge

Kübik yörünge üçüncü dereceden bir fonksiyon olup denklemi

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{2.36}$$

biçimindedir. Bu durumda yörünge boyunca açısal hız ve ivme:

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2, \qquad \ddot{\theta}(t) = 2a_2 + 6a_3 t$$

olarak elde edilir. Kübik yörünge yukarıdaki dört şartı rahatlıkla sağlar. Yörünge denklemleri bu dört şart ile birleştirilip çözülürse katsayılar:

$$a_0 = \theta_0, \qquad a_1 = 0, \qquad a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0), \qquad a_3 = \frac{-2}{t_f^3}(\theta_f - \theta_0)$$

olarak elde edilir. Bu katsayılar yörünge denklemlerinde yerlerine konulursa

$$\theta(t) = \theta_0 + \frac{3}{t_f^2}(\theta_f - \theta_0)t^2 - \frac{2}{t_f^3}(\theta_f - \theta_0)t^3 \tag{2.37}$$

$$\dot{\theta}(t) = \frac{6}{t_f^2}(\theta_f - \theta_0)t - \frac{6}{t_f^3}(\theta_f - \theta_0)t^2 \tag{2.38}$$

denklemleri, sonra bu denklemler $n$ adımlı kesikli hale dönüştürülürse

$$\theta[i] = \theta_0 + (\theta_f - \theta_0)\left(\frac{i}{n}\right)^2\left(3 - \frac{2i}{n}\right) \quad , \quad i = 0 \cdots n \tag{2.39}$$

$$\dot{\theta}[i] = \frac{6(\theta_f - \theta_0)}{t_f}\frac{i}{n}\left(1 - \frac{i}{n}\right) \quad , \quad i = 0 \cdots n \tag{2.40}$$

denklemleri elde edilir.

### 2.1.4.3. Sinüzoidal yörünge

Sinüzoidal yörünge denklemi:

$$\theta(t) = a + bCos(wt) \tag{2.41}$$

biçimindedir. Bu durumda yörünge boyunca açısal hız ve ivme:

$$\dot{\theta}(t) = -b\,w\,Sin(wt), \qquad \ddot{\theta}(t) = -b\,w^2\,Cos(wt)$$

olarak elde edilir. Sinüzoidal yörünge yukarıdaki dört şartı rahatlıkla sağlar. Yörünge denklemleri bu dört şart ile birleştirilip çözülürse katsayılar

$$w = \frac{\pi}{t_f}, \qquad a = \frac{\theta_f + \theta_0}{2}, \qquad b = -(\frac{\theta_f - \theta_0}{2})$$

olarak elde edilir. Bu katsayılar yörünge denklemlerinde yerlerine konulursa

$$\theta(t) = (\frac{\theta_f + \theta_0}{2}) - (\frac{\theta_f - \theta_0}{2})Cos(\frac{\pi\, t}{t_f}) \tag{2.42}$$

$$\dot{\theta}(t) = (\frac{\theta_f - \theta_0}{2})(\frac{\pi}{t_f}) Sin(\frac{\pi\, t}{t_f}) \tag{2.43}$$

denklemleri, sonra bu denklemler $n$ adımlı kesikli hale dönüştürülürse

$$\theta[i] = (\frac{\theta_f + \theta_0}{2}) - (\frac{\theta_f - \theta_0}{2}) Cos(\frac{\pi\, i}{n}) \quad , \quad i = 0 \cdots n \tag{2.44}$$

$$\dot{\theta}[i] = (\frac{\theta_f - \theta_0}{2})(\frac{\pi}{t_f}) Sin(\frac{\pi\, i}{n}) \quad , \quad i = 0 \cdots n \tag{2.45}$$

denklemleri elde edilir.

### 2.1.4.4. Doğrusal yörünge

Doğrusal yörünge denklemi

$$\theta(t) = a + b\,t \tag{2.46}$$

biçimindedir. Bu durumda yörünge boyunca açısal hız ve ivme:

$$\dot{\theta}(t) = b\,, \qquad \ddot{\theta}(t) = 0$$

olarak elde edilir. Doğrusal yörüngede açısal hız sabittir. Bu yüzden sadece başlangıç ve bitiş açıları (2.46) denkleminde yerlerine konularak çözülürse katsayılar

$$a = \theta_0\,, \qquad b = \frac{\theta_f - \theta_0}{t_f}$$

olarak elde edilir. Bu katsayılar yörünge denklemlerinde yerlerine konulursa

$$\theta(t) = \theta_0 + (\frac{\theta_f - \theta_0}{t_f})\, t \tag{2.47}$$

$$\dot{\theta}(t) = \frac{\theta_f - \theta_0}{t_f} \tag{2.48}$$

denklemleri, sonra bu denklemler $n$ adımlı kesikli hale dönüştürülürse

$$\theta[i] = \theta_0 + (\theta_f - \theta_0)\left(\frac{i}{n}\right) \quad , \quad i = 0 \cdots n \tag{2.49}$$

$$\dot{\theta}[i] = \frac{\theta_f - \theta_0}{t_f} \quad , \quad i = 0 \cdots n \tag{2.50}$$

denklemleri elde edilir.

Bu tez çalışmasında, robot kolunun her bir eklemine ait takip etmesi istenen konum referans ve hız referans yörüngeleri bu bölümde anlatılan sinüzoidal yörünge esaslarına göre belirlenmiştir.

## 2.2. Altı Eklemli Puma 560 Robot Kolunun Dinamik Modeli

Bu bölümde, tez çalışmasında kullandığımız Unimate firmasının ürettiği 6 eklemli Puma 560 model robot kolunun dinamik modellenmesi anlatılacaktır.

Endüstriyel uygulamalarda robot kolları taşıma, kaldırma, tutma, döndürme v.s benzeri birçok görevde kullanılmaktadır. Robot kolunun yapacağı göreve bağlı olarak eklem sayısının yeterli sayıda olması istenir. Örneğin tutma görevi yapacak bir robot kolunun uç noktasında el yapısına benzer bir mekanizmaya ve bu mekanizmanın hareketini destekleyecek ilave eklemlere gereksinim duyulur. Bu görev için en az iki eklem daha tutma organına eklenmelidir. Görüldüğü gibi kolun icra edeceği göreve bağlı olarak eklem sayısı arttırılmaktadır.

Öte yandan hareket esnekliğini arttırmak için eklem sayısını arttırmak çözüm sağlamaktadır. Ancak eklem sayısının arttırılması, eklemler arasında yüksek oranda etkileşimler mevcut olduğundan robot kolunun kontrolünü güçleştirmektedir. Bu nedenle optimum tasarımlara ihtiyaç vardır.

Endüstriyel uygulamalara bakıldığında üç ve daha fazla ekleme sahip kolların kullanıldığı görülmektedir [40]. Özellikle beş ve altı ekleme sahip robot kolları optimum çözümler sunmaktadır. İnsan kolunu taklit ederek tasarlanan bu robot kolları omuz, dirsek, bilek ve el parçalarından oluşurlar. İnsan kolunun sahip olduğu hareket esnekliğine yakın bir hareket manevrasına haizdirler.

Altı eklemli robot kolları arasında en yaygın kullanılanı ve insan koluna benzerliği ile göze çarpan PUMA ailesine ait PUMA 560 robot koludur. Bu robot kolu sahip olduğu tasarım ve programlanabilir özelliği ile hassas kontrol gerektiren görevlerde kullanılmaktadır.

Bu tez çalışmasında, kontrol algoritmalarını test etmek için endüstriyel uygulamalarda yaygın bir şekilde kullanılan ve optimum eklem sayısına sahip Puma 560 model altı eklemli robot kolu tercih edilmiştir. Puma 560 robot kolu, otomotiv panel montajı, elektronik devre baskı yazımı, radyo-televizyon setlerinin montajı gibi yüksek hassasiyet gerektiren görevler ile eczacılık ve gıda sektöründe paketleme gibi elle tutma, taşıma ve kaldırma görevleri için kullanılan bir endüstriyel robot koludur [64]. İnsan koluna benzerliği ile dikkat çeken bu robot kolu, yüksek oranda hassasiyete ve hareket esnekliğine sahiptir. Ayrıca programlanabilirlik özelliğinden dolayı eğitimci ve araştırmacılarca da yoğun olarak tercih edilmektedir. Sahip olduğu bu özellikleri tez çalışmamızda onu kullanmayı tercih sebebi kılmıştır.

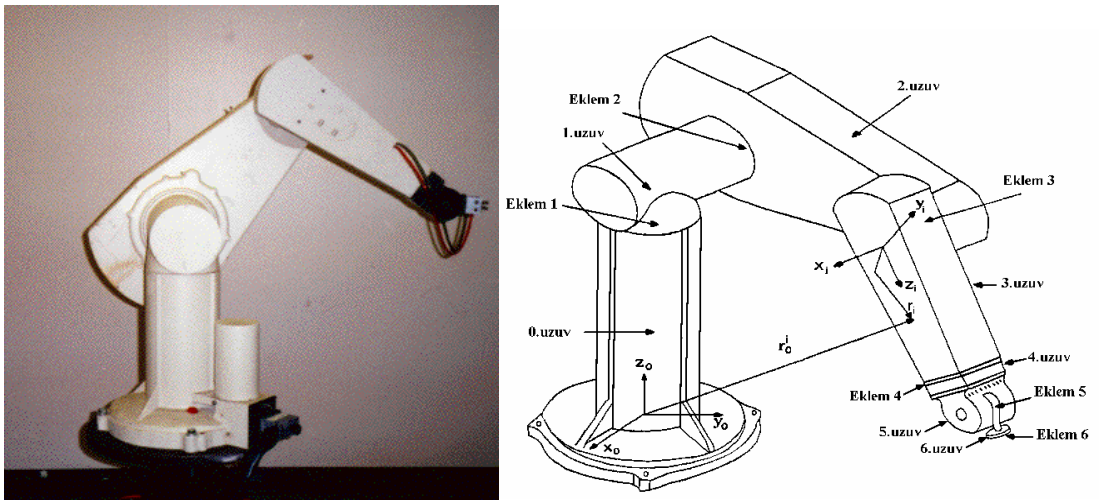
Robot kolunun dinamik modelinde Lagrange-Euler denklem sisteminden yararlanılmış, denklem çözümleri 4. mertebeden Runge-Kutta yöntemi ile gerçekleştirilmiştir. Sürtünme ve motor ataletlerini içeren ayrıca yük taşıma durumunun ilave edildiği en genel anlamda dinamik model oluşturulmuştur.

### 2.2.1. Dinamik modelleme

Bölüm 2.1‘de bahsedildiği gibi robot kollarının dinamik modellenmesinde büyük oranda eklemler arası etkileşimler mevcuttur. Dolayısıyla eklem sayısının artması dinamik modeli daha da karmaşık hale getireceğinden, robot kolunun kontrolünü

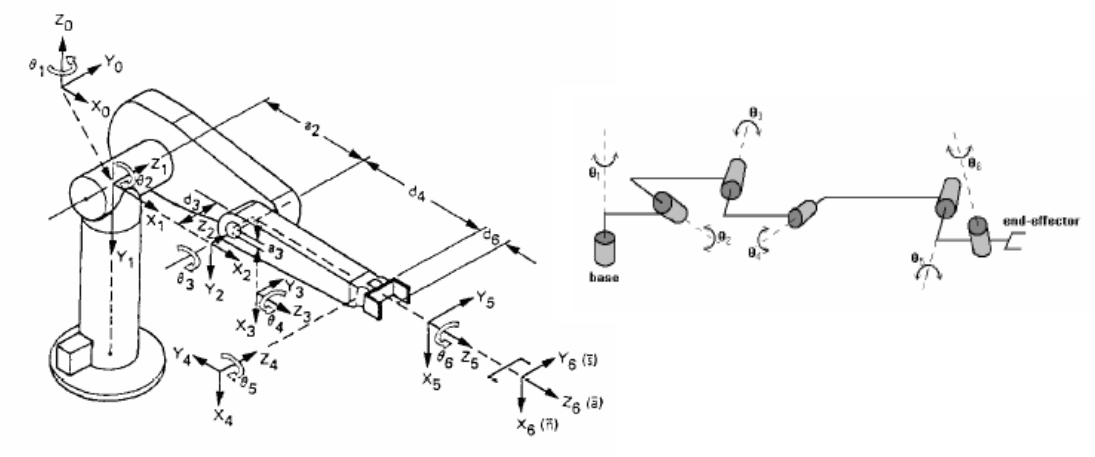
zorlaştıracaktır. Bu nedenle yapılacak göreve bağlı olarak eklem sayısının çok fazla artmaması istenir. Diğer bir deyişle optimum bir eklem sayısının olması gerekir.

Bu tezde, tasarlanan öngörülü kontrol algoritmalarının robot kolu uygulamalarındaki performanslarını incelemek için 6 ekleme sahip PUMA 560 robot kolunun dinamik modeli oluşturulmuştur. PUMA 560, insan koluna benzerliği ile göze çarpan bir koldur ve insan koluna yakın bir hareket esnekliğine sahiptir. PUMA 560 robot kolunun görünümü Şekil 2.8'de verilmiştir.




Şekil 2.8. Puma 560 robot kolu

Görüldüğü gibi robot kolu 7 uzuv 6 eklemden oluşmaktadır. İlk uzuv hareketsiz olup yere sabitleme için kullanılmaktadır "bel" (waist) olarak adlandırılır. 2.uzuv hareketli olup hareket manevrası en büyük olan uzuvdur, kolun omuz (shoulder) parçasıdır. Bunu 3. uzuv yani dirsek (elbow) takip eder. Diğer uzuvlar uç noktada yer almaktadırlar ve bilek (waist) olarak anılırlar. Yük tutma da oryantasyonu sağlayan uzuvlardır. Kolun açık kinematik modeli Şekil 2.9'da verilmiştir.




Şekil 2.9. Puma 560 robot kolunun açık kinematik modeli

Robot kolunun dinamik modellenmesinde daha önce yapılmış çalışmalardan yararlanılmıştır. Armstrong ve ekibinin yapmış olduğu bir çalışmada PUMA 560 robot kolunun dinamik model parametreleri hesaplanmıştır [49]. Manipülatör tasarımında dinamik model için bu parametre değerleri kullanılmıştır ve Ek A'da ayrıntılı bir şekilde anlatılmıştır. Dinamik model için gerekli parametreler şu şekildedir: Modifiye edilmiş Denavit-Hartenberg parametreleri, motor ve sürücü parametreleri, maximum motor tork değerleri.

Tablo 2.1. Puma 560 'a ait Denavit-Hartenberg parametreleri

| $i$ | $\alpha_{i-1}$ (derece) | $\theta_i$ | $a_{i-1}$ (metre) | $d_i$ (metre) |
|---|---|---|---|---|
| 1 | 0 | $\theta_1$ | 0 | 0 |
| 2 | -90 | $\theta_2$ | 0 | 0.2435 |
| 3 | 0 | $\theta_3$ | 0.4318 | -0.0934 |
| 4 | 90 | $\theta_4$ | -0.0203 | 0.4331 |
| 5 | -90 | $\theta_5$ | 0 | 0 |
| 6 | 90 | $\theta_6$ | 0 | 0 |

PUMA 560 robot koluna ait etiket özellikleri [73, 74]:

Eklem sayısı: 6

Servis uzunluğu: 0.95m radius

Maximum Hız: 1.0 nm/sn

Sürücüler: DC motor

Tablo 2.2. Altı eklemli robot kolunun parametre değerleri

| ***I*** | **1** | **2** | **3** | **4** | **5** | **6** | **Birim** |
|---|---|---|---|---|---|---|---|
| $m_i$ | 0 | 17.4 | 4.8 | 0.82 | 0.34 | 0.09 | kg |
| Hareket alanı | -160 ile 160 | -225 ile 45 | -45 ile 225 | -110 ile 170 | -100 ile 100 | -266 ile 266 | Derece |
| $d_i$ | 0 | 0.2435 | -0.0934 | 0.4331 | 0 | 0 | Metre |
| $a_i$ | 0 | 0.4318 | 0.0203 | 0 | 0 | 0 | Metre |
| $k_{i\,xx}^2$ | 0.1816 | 0.0596 | 0.0151 | 0.119 | 0.0009 | 0.0008 | $m^2$ |
| $k_{i\,yy}^2$ | 0.0152 | 0.1930 | 0.0155 | 0.029 | 0.0009 | 0.0008 | $m^2$ |
| $k_{i\,zz}^2$ | 0.1811 | 0.1514 | 0.0021 | 0.0118 | 0.0009 | 0.0008 | $m^2$ |

Burada;

$m_i$ = $i$ uzvunun kütlesi,

$l_i$ = $i$ uzvunun uzunluğu,

$a_i$ = Dönel $x_{i-1}$ ekseni boyunca $i$ .koordinat sisteminin kayma mesafesi,

$\alpha_i$ = $z_{i-1}$ eksenine göre $z_i$ ekseninin yönlendirmesi,

$\bar{x}_i , \bar{z}_i$ = $i$ .koordinat çerçevesine göre $i$ uzvunun kütle merkezinin koordinatları,

$k_{ijj}$ = Jirasyon yarıçapı ( $I = m_i\, k_{ijj}^2$ ).

Bölüm 2.1'de anlatılan Lagrange-Euler denklem sistemi ile robot kolunun dinamik modeli oluşturulmuştur. Dinamik modelde, L-E denklem sistemi ile elde edilen genel modele sürtünme, motor ataletleri ve uç noktada taşınan yükün etkisi de eklenerek en genel anlamda tork ifadesi elde edilmiştir.

Robot kolunun hareketlerini modellemede Lagrange-Euler yöntemi kullanılmıştır. Bu yönteme göre $n$ eklemli bir robot kolu için $n$x1 boyutlu eklem tork vektörü $\tau(t)$ ile $n$x1 boyutlu eklem açı vektörü $\theta(t)$ arasındaki ilişki aşağıdaki gibi doğrusal olmayan bir diferansiyel denklem takımı ile gösterilir.

$$\tau(t) = D(\theta)\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta) \qquad (2.51)$$

Tez çalışmaları sırasında, Lagrange-Euler denklem takımına eklemlerdeki sürtünmeler, eklemlerdeki motorlara ait ataletlerin dinamik modele etkisi, robot kolunun uç elemanında bir yük taşınması ve taşınan yükün taşıma esnasında düşmesi

durumları da ayrıca ilave edilerek simülasyon çalışmaları yapılmıştır. Bu durumda (2.51)'deki genel denklem aşağıdaki şekilde yeniden ifade edilir.

$$\tau(t) = D(\theta)\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta) + \tau_s(t) + \tau_y(t) \tag{2.52}$$

Burada $\tau(t)$; 6x1 'lik toplam tork vektörü, $D(\theta)$; 6x6 atalet matrisi, $H(\theta,\dot{\theta})$;6x1 Coriolis ve merkezkaç kuvvet vektörü, $G(\theta)$; 6x1 yer çekim kuvvet vektörü, $\tau_s(t)$;sürtünme vektörü ve $\tau_y(t)$ ise yükten oluşan tork vektörüdür.

Tez çalışmasında kullanılan bu dinamik model için, Puma 560 açık dinamik modelinin oluşturulduğu çalışmada atalet matrisi ($D(\theta)$), Coriolis ve merkezkaç kuvvet vektörü ($H(\theta,\dot{\theta})$) ve yer çekim kuvvet vektörüne ($G(\theta)$) ait hesaplamalar kullanılmıştır. Bu hesaplamalar Ek 1'de ayrıntılı bir şekilde anlatılmıştır [49].

### 2.2.2. Sürtünme etkisi

Sürtünme etkilerinin modellenmesi ve dengelenmesi ile robotların performansı arttırılabilmektedir. Dinamik modellemede robot kolunun sürtünmesi için statik, kinetik (Coulomb) ve akışkan sürtünmesini içeren bir sürtünme modellemesi yapılmıştır. Bu model aşağıda denklem ile gösterilmektedir.

$$\tau_{\text{friction}} = f_s \left( \frac{\operatorname{sgn}(\dot{q})}{1 + \left( \frac{\dot{q}}{x_s} \right)^2} \right) + f_k \tanh(\dot{q}) + k_{vn}(\dot{q}) \tag{2.53}$$

Burada $f_s$ statik sürtünme, $x_s$ Stribeck Etkisinden oluşan statik sürtünme sabiti, $f_k$ kinetik sürtünme ve $k_{vn}$ akışkan sürtünmesidir. Puma 560 robot koluna ait sürtünme parametreleri Tablo 2.3'de verilmiştir [75].

Tablo 2.3. Puma 560 sürtünme parametreleri

| Link | $f_s$ | $f_k$ | $k_{vn}$ | $x_s$ |
|---|---|---|---|---|
| 1 | 5 | 2 | 1 | 0.1 |
| 2 | 5 | 2 | 1 | 0.1 |
| 3 | 2.5 | 1 | 1 | 0.1 |
| 4 | 0.3 | 0.1 | 0.05 | 0.1 |
| 5 | 0.2 | 0.1 | 0.05 | 0.1 |
| 6 | 0.2 | 0.1 | 0.05 | 0.1 |

Sürtünme etkisinden oluşan tork değeri $\tau_s(t)$, eklemlere ait genel tork ifadesine eklenir. Bu durumda genel denklem aşağıdaki gibi elde edilir.

$$\tau(t) = D(\theta)\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta) + \tau_s(t) + \tau_y(t) \tag{2.54}$$

### 2.2.3. Yük etkisi

Robot kolunun yük taşıma esnasında kontrolünü gözlemlemek için manipülatöre yük durumu ile ilgili kontrol paneli eklenmiştir. Yük etkisi için yüklü ve yüksüz durumlar kullanıcı tercihine bırakılmıştır. Kullanıcı robot kolunun taşıyacağı yük ağırlığını arayüzden belirleyebilmektedir. Öte yandan kontrol esnasında yük düşmesi olayını incelemek için yükün düştüğü adım sayısı belirlenmelidir. Kullanıcı yükün düştüğü adım sayısını konsoldan girerek yük düşme esnasındaki kontrolün durumunu inceleyebilmektedir.

Taşınan yükten dolayı oluşan yük tork vektörü $\tau_y(t)$'yi bulmak için önce uç eleman vektörü $r(t)$ elde edilir. Uç eleman vektörü $r(t)$ ile eklem açıları vektörü $\theta(t)$ arasındaki ilişki aşağıda verilmektedir.

$$r(t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \Lambda(\theta) \tag{2.55}$$

$\Lambda(\theta)$, $3\ x\ 1$ boyutunda robot kolu dinamiğini içeren ve düz kinematiği simgeleyen bir vektördür.

Eklem uzayı $\{\theta\}$ ile kartezyen uzay $\{r\}$ arasındaki dönüşümü sağlayan bağıntılar

$$\begin{aligned} \dot{r}(t) &= J(\theta)\dot{\theta} \\ \ddot{r}(t) &= J(\theta)\ddot{\theta} + \dot{J}(\theta,\dot{\theta})\dot{\theta} \end{aligned} \tag{2.56}$$

şeklindedir. Burada, $J(\theta) = \partial\Lambda(\theta)/\partial\theta$, $3xn$ boyutlarında robot kolu Jacobian matrisidir. $\dot{J}(\theta,\dot{\theta})$ ise Jacobian matrisin birinci türevidir.

Robot kolu uç elemanının taşıdığı noktasal yükün kütlesi $m$ olsun. Bu durumunda yükün $\ddot{r}(t)$ ivmesi ile hareket edebilmesi için uç eleman yüke $6x1$'lik $f(t)$,

$$f(t) = m \cdot (\ddot{r}(t) - \vec{g}) \tag{2.57}$$

kuvvetini uygulayacaktır. $\vec{g}$, yerçekimi ivme vektörü, temel koordinat sistemi eksenleri cinsinden ifade edilmelidir. Temel koordinat sisteminde $\vec{z}_0$ ekseninin yönü yerçekimi ivmesinin zıt yönünde alınırsa bu durumda $\vec{g}$ vektörü,

$$\vec{g} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

şeklinde gösterilir.

Jacobian matrisi $J(\theta)$'nin transpozesi uç eleman kuvvet vektörü $f(t)$ ile çarpılırsa $m$ kütlesinden dolayı oluşan ilave eklem tork vektörü

$$\tau_y(t) = J^T(\theta)\, f(t) \tag{2.58}$$

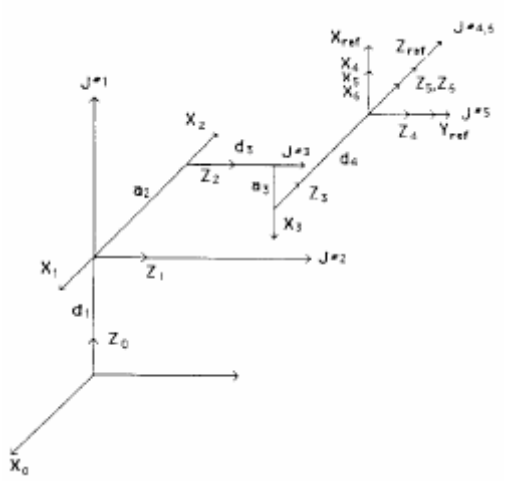
elde edilir. $f(t)$ yerine konulursa

$$\tau_y(t) = m\, J^T(\theta)[\, J(\theta)\,\ddot{\theta} + \dot{J}(\theta,\dot{\theta})\,\dot{\theta} - \vec{g}\,] \tag{2.59}$$

$$\tau_y(t) = m\, J^T J\,\ddot{\theta} + mJ^T\,\dot{J}\,\dot{\theta} - mJ^T\vec{g} = m\, L(\theta,\dot{\theta},\ddot{\theta}) \tag{2.60}$$

elde edilir. Burada $L(\theta,\dot{\theta},\ddot{\theta})$, birim yük başına düşen ilave tork vektörüdür.

Uç eleman koordinatları $(x,\, y,\, z)$'in daha kolay hesaplanabilmesi için robot modeli daha anlaşılır bir biçimde Şekil 2.10'da tekrardan verilmiştir [49].




Şekil 2.10. Altı eklemli robot modelinde $(x,\, y,\, z)$ koordinatlarının teşkili

Robot modeli kullanılarak uç elemanının $(x,\, y,\, z)$ koordinatları:

$$x = \begin{bmatrix} d_6\left(\cos(\theta_2+\theta_3)\cos\theta_4\sin\theta_5 + \sin(\theta_3+\theta_3)\cos\theta_5\right) \\ +\, d_4\sin(\theta_2+\theta_3) + a_3\cos(\theta_2+\theta_3) + a_2\cos\theta_2 \end{bmatrix}\cos\theta_1 - \left(d_2 + d_6\sin\theta_4\sin\theta_5\right)\sin\theta_1$$

$$y = \begin{bmatrix} d_6\left(\cos(\theta_2+\theta_3)\cos\theta_4\sin\theta_5 + \sin(\theta_3+\theta_3)\cos\theta_5\right) \\ + d_4\sin(\theta_2+\theta_3) + a_3\cos(\theta_2+\theta_3) + a_2\cos\theta_2 \end{bmatrix}\sin\theta_1 + \left(d_2 + d_6\sin\theta_4\sin\theta_5\right)\cos\theta_1$$

$$z = d_6\left(\cos(\theta_2+\theta_3)\cos\theta_5 - \sin(\theta_2+\theta_3)\cos\theta_4\sin\theta_5\right) + d_4\cos(\theta_2+\theta_3) - a_3\sin(\theta_2+\theta_3) - a_2\sin\theta_2$$

olarak hesaplanır. Elde edilen bu koordinatlar sayesinde yukarıda anlatıldığı gibi taşınan yükten dolayı oluşan yük tork vektörü $\tau_y(t)$ rahatlıkla hesaplanabilir.

Robot kolunun uç noktada taşıdığı varsayılan yükü temsil eden bu tork değeri denklem 2.54 'de verilen genel tork denklemine ilave edilir.

### 2.2.4. Motor ataletleri ve gerilim hesabı

Robot kollarında eklemlerin hareketi için küçük gerilim değişikliklerine bile duyarlı olan servo doğru akım (DC) motorları kullanılır. Puma 560 robot kolunda da 6 adet eklem motoru mevcuttur. Bu 6 adet servo motor değişik çaplarda dişlilerle aracılığı ile eklemleri hareket ettirmektedirler. Söz konusu motorların dinamik modele etkisi mevcuttur. Bu etkiyi hesaplayabilmek için kullanılan motorların dinamik model parametrelerine ihtiyaç vardır. Puma 560 robot kolunun açık dinamik modellemesinin yapıldığı Armstrong 'a [49] ait çalışmada eklemlere ait motor ataletleri belirlenmiştir. Tablo 2.4'de motor ataletleri verilmiştir [49]. Motor ataletleri denklem 2.4 de verilen tork ifadesinde atalet matrisi bünyesinde yer almaktadır ve toplam atalet ifadesine ilave edilmiştir. İlgili ifadeler Ek A 'da ayrıntılı bir şekilde verilmiştir.

Tablo 2.4. Puma 560 eklem motor ataletleri (kg-m$^2$)

| 1.motor | 2.motor | 3.motor | 4.motor | 5.motor | 6.motor |
|---|---|---|---|---|---|
| 1.14 | 4.71 | 0.83 | 0.2 | 0.179 | 0.193 |

Öte yandan eklemlere uygulanacak tork değerlerinin hesaplanması ile motorlara uygulanacak voltaj değerleri de hesaplanabilir. Bunun için motor iç dirençleri, dişli

dönüştürme oranı ve tork sabitlerinden oluşan değerleri kullanarak bu hesaplama yapılabilir. Puma 560'a ait motor parametre değerleri aşağıda verilmiştir [76].

Tablo 2.5. Puma 560 eklem motor parametreleri

| Motor $i$ | Armatür Direnci $R_i$ (ohm) | Geri EMF Sabiti $K_i^b$ (volt/(rad/saniye)) | Tork sabiti $K_i^t$ (Nm/amper) | Dişli oranı $N_i$ |
|---|---|---|---|---|
| 1 (Bel) | 1.6 | 0.19 | 0.2611 | 62-55 |
| 2 (Omuz) | 1.6 | 0.19 | 0.2611 | 107-81 |
| 3 (Dirsek) | 1.6 | 0.19 | 0.2611 | 53-17 |
| 4 (Bilek) | 3.9 | 0.12 | 0.0988 | 76-04 |
| 5 (Bilek) | 3.9 | 0.12 | 0.0988 | 71-92 |
| 6 (Bilek) | 3.9 | 0.12 | 0.0988 | 76-65 |

Eklemlere uygulanan tork değeri ile eklem motorunun gerilimi arasında bağıntı aşağıdaki denklemde tanımlanmıştır. Denklemde; $R_i$:armatür direnci, $N_i$:dişli oranı, $K_i^t$:tork sabiti $\tau_i$:eklem torku, $K_i^b$:geri EMF sabiti, $\dot{q}_i$ eklem açısal hızı ve $V_i$ ise eklem gerilimidir.

$$V_i = \left[ \frac{R_i}{N_i K_i^t} \right] \tau_i + \left( K_i^b N_i \right) \dot{q}_i \tag{2.61}$$

Sürtünme, motor ataletleri ve uç noktada taşınan yükün etkisini de içeren en genel anlamda bir dinamik model oluşturulmuştur. Lagrange-Euler denklem sistemi ile modellenen bu yapıda birbirleriyle yüksek oranda etkileşimli altı adet ikinci dereceden doğrusal olmayan diferansiyel denklem mevcuttur. Bu tür denklemlerin çözümü için klasik nümerik çözümler kullanılamaz. Özel metotların kullanılması gerekmektedir. Denklem takımının çözümü için Runge-Kutta integrasyon metodu kullanılmıştır. Bu metot Ek B'de ayrıntılı bir şekilde anlatılmıştır.

# BÖLÜM 3. TASARLANAN KONTROL ALGORİTMALARI

Bu bölümde, altı eklemli bir robot kolunun eklem esaslı kontrolü için tasarlanan öngörülü kontrol algoritmaları anlatılacaktır. İlk olarak D. W. Clarke [28-32] tarafından tanıtılan Genelleştirilmiş Öngörülü Kontrol (Generalized Predictive Control-GPC) algoritmasının yapısı incelenecek, daha sonra GPC algoritmasının optimizasyonunu geliştirmek için önerilen Basit Genetik Algoritma uyarlamalı GPC algoritması anlatılacaktır. Bölüm 3.3 'de ise D. Soloway [56, 63-65] tarafından tanıtılan, GPC performansını arttırmak için öngörü işleminin bir yapay sinir ağı üzerinden yapıldığı Nöro Genelleştirilmiş Öngörülü Kontrol (Neural Generalized Predictive Control-NGPC) algoritması incelenecektir. Son olarak bölüm 3.4 'de, NGPC algoritmasının performansını doğrudan etkileyen kullandığı yapay sinir ağ modelinde dinamik ağ kullanımını önerdiğimiz Yinelenen Elman Ağ uyarlamalı Nöro Genelleştirilmiş Öngörülü Kontrol (Recurrent Elman's implemented Neural Generalized Predictive Control-ENGPC) algoritması tanıtılacaktır.

## 3.1. Genelleştirilmiş Öngörülü Kontrol

Genelleştirilmiş Öngörülü Kontrol (Generalized Predictive Control - GPC), uzun menzilli öngörülü kontrol algoritmaları sınıfına ait olup ilk olarak D. W. Clarke ve arkadaşları tarafından 1987 yılında tanıtılmıştır [28-32]. GPC, esas olarak sistem çıkışı y(t)'nin sonlu bir ufuk üzerinden öngörüldüğü ve kontrol sinyalleri dizisinin kuadratik bir maliyet fonksiyonunu minimize edecek şekilde hesaplandığı bir algoritmadır. Uygun cevabı elde etmek için kontrol sinyallerinin beklenti büyüklüklerinde bazı tahminler yapılmaktadır, bu tahminler GPC yaklaşımının püf noktasını teşkil eder.

GPC, tek bir öngörü yerine sonlu bir öngörü ufuğu boyunca oluşturulan öngörü dizilerini kullanan daha genel bir maliyet fonksiyonuna sahip olması ile kendinden

önce yapılan öngörülü çalışmalardan esaslı bir şekilde farklıdır. Öngörülü kontrol algoritmasının esası temel olarak; bir öngörücü, bir maliyet fonksiyonu, bu maliyet fonksiyonunu minimize edecek bir kontrol dizisini seçme algoritması ve adaptif durumda parametre tanılama algoritmasının birleşmesinden meydana gelmektedir. Yapısı prensip olarak şu şekilde özetlenebilirler.

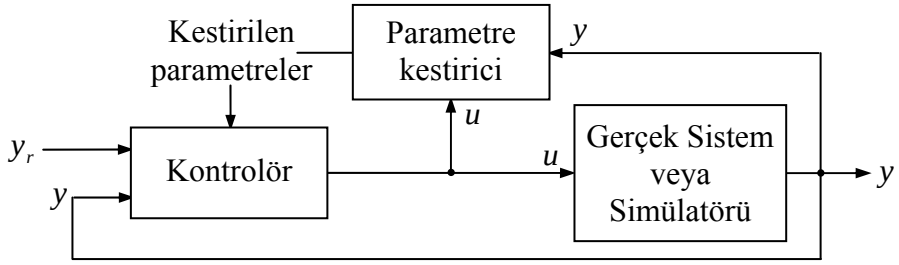
1. Her bir $(t)$ anında proses çıkışı $y(t+k)$, uzun menzilli bir ufuk $\{ k = 1, \cdots, N_2 \}$ üzerinden öngörülür. Öngörülen proses çıkış değerleri $\hat{y}(t+k)$ şeklinde gösterilir. Öngörü işlemi, proses dinamiği için seçilen bir modele göre yapılır. Öngörülen proses çıkış değerleri $\hat{y}(t+k)$ $\{ k = 1, \cdots, N_2 \}$, öngörülen proses giriş değerlerine $\hat{u}(t+k)$ $\{ k = 0, 1, \cdots, N_{u-1} \}$ bağlı olacaktır. Buradaki $\hat{u}(t+k)$ değerleri gelecekteki kontrol büyüklüklerinin gerçek değerleri olmayıp $(t)$ anında uzun menzilli bir ufuk üzerinde öngörülen kontrol büyüklükleridir.

2. Proses çıkışının istenen değişimini gösteren bir referans yörüngesi $y_r(t+k)$ $\{ k = 1, \cdots, N_2 \}$ belirlenir.

3. Gelecekteki öngörülen kontrol büyüklükleri $\hat{u}(t+k)$ $\{ k = 0, 1, \cdots, N_{u-1} \}$ öngörü hatalarına $( y_r(t+k) - \hat{y}(t+k) )$ $\{ k = 1, \cdots, N_2 \}$ bağlı bir maliyet fonksiyonunu minimize edecek şekilde hesaplanır.

4. Elde edilen öngörülen kontrol büyüklüklerinden sadece ilk elemanı $\hat{u}(t)$ alınır ve gerçek sisteme uygulanır. Diğer elemanlar sisteme uygulanmaz, bir sonraki örnekleme anında kullanılmak üzere tüm dizi elemanları geriye kaydırılır, yeni bir çıkış gözlemi $y(t+1)$ elde edilir ve proses bitene kadar tüm işlemler tekrarlanır.

GPC algoritmasının blok diyagramı Şekil 3.1'de verilmiş olup SISO için $u$ sistem girişi, $y$ sistem çıkışı, $y_r$ ise referans yörünge, MIMO için ise $u$ sistem giriş vektörü, $y$ sistem çıkış vektörü, $y_r$ ise referans yörünge vektörüdür. Parametre kestirimi uygun bir parametre kestirme algoritması ile yapılır.




Şekil 3.1. GPC algoritmasının blok diyagramı

GPC algoritması kontrol edilecek sisteme SISO (Single Input Single Output - Tek Giriş Tek Çıkış) ve MIMO (Multiple Inputs Multiple Outputs - Çok Giriş Çok Çıkış) olmak üzere iki şekilde uygulanmaktadır. Tek girişli tek çıkışlı bir sisteme SISO, çok girişli çok çıkışlı bir sisteme ise hem SISO hem de MIMO olarak uygulanabilir. Örneğin, $m$ eklemli bir robot kolu $m$ tane girişe ve $m$ tane çıkışa sahip olduğundan $m \times m$'lik bir sistemdir. Girişler eklemlere uygulanacak torklar / voltajlar, çıkışlar eklem hızlarıdır. Robot kolu tek bir sistem olmasına rağmen her bir eklem bir girişe ve bir çıkışa sahip olduğundan, her bir eklem ayrı bir sistem gibi düşünülebilir. Bu durumda, yapı itibarıyla aynı fakat birbirinden tamamen bağımsız çalışan $m$ tane ayrı SISO algoritması tasarlanıp kullanılır. Eğer kol tek bir sistem olarak düşünülürse bütün eklemleri kapsayacak ortak bir MIMO algoritması yeterlidir. Her bir eklem diğer eklemlerle dinamik olarak etkileşimli olması nedeniyle GPC algoritmasının MIMO mantığına göre tasarlanması daha kapsamlı bir yaklaşımmış gibi görülebilir. Fakat bu etkileşimler eklem hızlarına yansıdığından GPC algoritmasının SISO mantığına göre tasarlanmasında bir sakınca yoktur. Ayrıca SISO mantığına göre tasarlanan algoritmalar daha hızlıdır ve paralel programlama için uygundur.

### 3.1.1. Carima (Controlled Auto Regressive Integrated Moving Average) model

GPC algoritmasında CARIMA formunda (Controlled Auto Regressive Integrated Moving Average) doğrusal bir işlem modeli kullanılmaktadır. Denklemi (3.1)'de verilmiştir. GPC algoritması bu model sayesinde kararsız sistemler için bile doğru çıkış öngörüleri elde edebilmektedir.

$$A(q^{-1})\, y(t) \;=\; B(q^{-1})\, u(t-1) \;+\; C(q^{-1})\, \xi(t) \,/\, \Delta \qquad (3.1)$$

$A(q^{-1})$, $B(q^{-1})$, $C(q^{-1})$ parametreleri $m \times m$'lik matris düzeyinde polinom değişkenlerdir. SISO için ($m = 1$), MIMO için ise ($m =$ eklem sayısı) olarak alınır.

$$A(q^{-1}) = \mathrm{I}_m + \sum_{j=1}^{n_a} a_j \, q^{-j},$$

$$B(q^{-1}) = \sum_{j=0}^{n_b} b_j \, q^{-j},$$

$$C(q^{-1}) = \mathrm{I}_m + \sum_{j=1}^{n_c} c_j \, q^{-j}$$

$n_a$, $n_b$, $n_c$ polinom mertebeleridir. $\mathrm{I}_m$ $m \times m$'lik birim matristir. Hesaplamalarda kolaylık olması açısından $n_c = 0$ alınacak, yani $C(q^{-1})$ ihmal edilecektir. $\Delta = 1 - q^{-1}$ olup fark alma operatörüdür ($\Delta u_j(t) = u_j(t) - u_j(t-1)$). $1/\Delta$ integratör görevi görerek hatayı daha da azaltır. Bu durumda hata $e(t) = \xi(t) / \Delta$ olur.

$y(t) = \begin{pmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{pmatrix}$ sistem çıkış vektörü, $u(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_m(t) \end{pmatrix}$ sistem giriş vektörü,

$\xi(t) = \begin{pmatrix} \xi_1(t) \\ \vdots \\ \xi_m(t) \end{pmatrix}$ ise hata veya gürültü bileşenleri vektörüdür.

GPC algoritması aşırı parametrelendirmeye karşı duyarlı değildir. Örneğin, 7 parametre ($n_a = 3$, $n_b = 2$) ile elde edilen kapalı çevrim davranışı, 3 parametre ($n_a = 1$, $n_b = 1$) ile de aynı şekilde sağlanmaktadır [28-30]. Daha az parametre daha az hesapsal yük getireceğinden kendinden önceki diğer uzun menzilli öngörülü kontrol algoritmalarından daha hızlıdır ve gerçek zamanlı sistemlere uygulanması daha kolaydır. Yapısında doğal integratör ($1/\Delta$) bulunmasından dolayı kararsız ve titreşimli sistemler için bile çok iyi performans göstermektedir. Ayrıca ayar

parametrelerinin ve ufukların özel şekilde seçilmeleri ile kendinden önceki diğer uzun menzilli öngörülü kontrol algoritmalarını elde etmek mümkündür.

$A(q^{-1})$ ve $B(q^{-1})$ polinomları başlangıçta $A(q^{-1}) = B(q^{-1}) = \mathrm{I}_m$ olarak alınır ve kontrol bitene kadar her kontrol adımında kontrol edilen sisteme uygun kontrolü sağlattıracak şekilde devamlı değişir.

### 3.1.2. Öngörülerin çıkartılması

Denklem 3.1'de verilen CARIMA model $A(q^{-1})$ ve $B(q^{-1})$ polinomlarının açılımları kullanılarak aşağıdaki şekle dönüştürülebilir.

$$y(t+j) = F_j(q^{-1})\, y(t) + E_j(q^{-1})\, B(q^{-1})\, \Delta u(t+j-1) + E_j(q^{-1})\, \xi(t+j) \qquad (3.2)$$

Bu denklemde $j$ değişkeni öngörü ufuğu boyunca ( $j = N_1, \cdots, N_2$ ) değişmektedir. $N_1$ öngörü ufuğunun başlangıcı, $N_2$ ise bitimidir. $F_j(q^{-1})$ ve $E_j(q^{-1})$ parametreleri $A(q^{-1})$, $B(q^{-1})$ parametreleri gibi $m \times m$'lik polinom değişkenler olup mertebeleri sırası ile $n_a$, $j-1$'dir.

Denklem 3.2'deki $E_j(q^{-1})\, \xi(t+j)$ çok küçük olduğundan ihmal edilir. Bu durumda, denklem 3.2 aşağıdaki öngörüler denklemine dönüşür.

$$\hat{y}(t+j) = F_j(q^{-1})\, y(t) + E_j(q^{-1})\, B(q^{-1})\, \Delta u(t+j-1) \qquad (3.3)$$

Bu öngörüler denklemi kolaylık olsun diye kısaca, $\hat{y} = G\,\tilde{u} + f$ şeklinde yazılabilir. Burada $\hat{y}$, $\tilde{u}$, $f$, $G$ parametreleri sırası ile

$$\hat{y} = \begin{pmatrix} \hat{y}(t+N_1) \\ \vdots \\ \hat{y}(t+N_2) \end{pmatrix}, \qquad \tilde{u} = \begin{pmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N_u-1) \end{pmatrix},$$

$$f = \begin{pmatrix} f(t+N_1) \\ \vdots \\ f(t+N_2) \end{pmatrix}, \qquad G = \begin{pmatrix} g_{N_1} & \cdots & g_1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & & \ddots & g_1 \\ \vdots & & \ddots & & & \vdots \\ g_{N_2} & \cdots & g_{N_2+N_1-1} & \cdots & & g_{N_2+N_u-1} \end{pmatrix}$$

şeklindedir. $\hat{y}$, $\tilde{u}$, $f$ vektörleri içindeki her bir değişken $m \times 1$'lik vektördür. $G$ matrisi içindeki her bir değişken ise $m \times m$'lik bir matristir. $N_u$ kontrol ufuğudur. Bu ufuk dışındaki kontrol değişimleri 0 alınır ( $\Delta u(t+j-1) = 0,\ j > N_u$ ).

### 3.1.3. Maliyet fonksiyonu

Maliyet fonksiyonu kontrolün başından sonuna kadar her kontrol adımı sonunda sisteme uygulanacak uygun $\Delta u(t)$'yu bulmak için gerekli olup denklemi SISO için (3.4)'te MIMO için (3.5)'te verilmiştir. Denklem 3.5 'in farkı $\hat{y}(t+j)$, $y_r(t+j)$ ve $\Delta u(t+j-1)$ değişkenlerinin $m \times 1$'lik vektör değişkenler olmasıdır.

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \left( \hat{y}(t+j) - y_r(t+j) \right)^2 + \sum_{j=1}^{N_u} \lambda(j) \left( \Delta u(t+j-1) \right)^2 \tag{3.4}$$

$$\begin{aligned} J(N_1, N_2, N_u) &= \sum_{j=N_1}^{N_2} \left( \hat{y}(t+j) - y_r(t+j) \right)^T \left( \hat{y}(t+j) - y_r(t+j) \right) \\ &+ \sum_{j=1}^{N_u} \left( \Delta u(t+j-1) \right)^T \Lambda(j) \left( \Delta u(t+j-1) \right) \end{aligned} \tag{3.5}$$

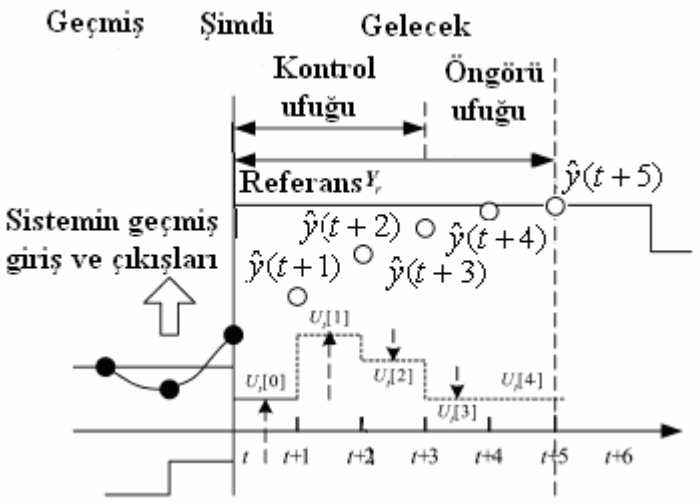
$$\Lambda(j) = \begin{pmatrix} \lambda_1(j) & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_m(j) \end{pmatrix} \qquad \Lambda = \begin{pmatrix} \Lambda(1) & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \Lambda(N_u) \end{pmatrix}$$

$\lambda(j)$ kontrol ağırlık faktörü olup oldukça küçük seçilir ($\lambda \approx 10^{-6}$ gibi).

Referans yörüngeler ne kadar yakın takip edilirse kontrol o kadar güçlü olur. Bu takibi sağlamak için önce maliyet fonksiyonu $J$ minimize edilir. Yani $\frac{\partial J}{\partial \tilde{u}} = 0$ yapılır. Sonra $\tilde{u}$ çekilerek aşağıdaki denklem elde edilir [28].

$$\tilde{u} = \left(G^T G + \Lambda\right)^{-1} G^T \left(y_r - f\right) \tag{3.6}$$

Bütün kontrol adımlarında bu denklem kullanılır. Kontrol dizisi elemanlarından ilki kullanılır, minimizasyon ve kontrol hesaplamaları her bir örnekleme anında tekrar edilir. Şekil 3.2'de örnek bir sistem için referans çıkışa yakınsama amacıyla GPC 'nin ürettiği öngörü cevapları gösterilmiştir.




Şekil 3.2. Öngörülen cevaplar

Kontrol bitene kadar her kontrol adımı sonunda $\tilde{u}$ 'nin ilk elemanı $\Delta u(t)$ alınır. Bir önceki kontrol girdisi ile toplanarak yeni kontrol girdisi ($u(t) = u(t-1) + \Delta u(t)$) elde edilir ve sisteme uygulanır. Fakat $G$ ve $f$ fonksiyonlarını belirleyen $A(q^{-1})$ ve $B(q^{-1})$ parametrelerinin her kontrol adımı sonunda tekrardan hesaplanması gerekir. Bu iş için bir parametre kestirim algoritmasına ihtiyaç vardır. Örneğin ardışık en küçük kareler algoritması kullanılarak parametreler kestirilebilir.

### 3.1.4. Ardışık en küçük kareler yöntemi

GPC algoritmasında kontrol parametreleri sistem çıkış/çıkışlarını referans yörünge/yörüngelere yakınsayacak şekilde her kontrol adımında sürekli güncellenir. Bu iş için bir parametre kestirme yöntemine ihtiyaç vardır. Parametre kestirim yöntemleri ölçülen giriş-çıkış bilgilerini kullanarak parametreleri günceller.

Ardışık en küçük kareler (Recursive Least Squares - RLS) [77], ardışık karekökler (Recursive Square Roots - RSQR) [78] ve ardışık U-D faktörizasyon [79] gibi pek çok parametre kestirme yöntemi mevcuttur. Bu yöntemler içinde en çok kullanılanı ardışık en küçük kareler yöntemi olup aşağıda anlatılmıştır.

Ardışık en küçük kareler yöntemini kullanmak için önce CARIMA model aşağıdaki denkleme dönüştürülür [40].

$$y(t) = -a_1 y(t-1) \cdots - a_{n_a} y(t-n_a) + b_0 u(t-1) + \cdots + b_{n_b} u(t-n_b-1) + \xi(t)/\Delta \qquad (3.7)$$

Bu denklem kısaca,

$$y(t) \;=\; \hat{\Theta}^T(t)\,\Phi(t) + e(t)$$

şeklinde yazılabilir. $N \;=\; (n_a + n_b + 1)\cdot m$ alınırsa $\hat{\Theta}^T(t)$ parametresi $m \times N$ 'lik, $\Phi(t)$ ve $e(t)$ parametreleri ise $N \times 1$'lik matris değişkenler olup açılımları aşağıda verilmiştir.

$$\hat{\Theta}^T(t) \;=\; \left(-a_1, \cdots, -a_{n_a}, b_0, \cdots, b_{n_b}\right)$$

$$\Phi(t) \;=\; \left( y(t-1)^T, \cdots, y(t-n_a)^T, u(t-1)^T, \cdots, u(t-n_b-1)^T \right)^T$$

$$e(t) \;=\; \xi(t)/\Delta$$

Sonra $A(q^{-1})$ ve $B(q^{-1})$ parametre bilgilerini içinde bulunduran $\hat{\Theta}^T(t)$ parametresi her kontrol adımı için aşağıdaki üç adım kullanılarak yeniden hesaplanır.

1. Kazanç denklemi,

$$K(t) = \frac{P(t-1)\,\Phi(t)}{\mu + \Phi^T(t)\,P(t-1)\,\Phi(t)} \tag{3.8}$$

kullanılarak kazanç hesaplanır. $\mu$ unutma faktörü olup 1'e yakın seçilir ($\mu = 0.95$ gibi ). $P(t)$ parametresi ise $N \times N$ 'lik bir matris olup ilk kontrol adımı için $P(0) = \mathrm{I}_N / \delta$ alınır. $\mathrm{I}_N$ $N \times N$ 'lik birim matris, $\delta$ ise sabit bir değer olup çok küçük seçilir ($\delta = 10^{-6}$ gibi). Diğer kontrol adımları için $P(t)$ 'nin hesaplanmasında aşağıdaki denklem kullanılır.

$$P(t) = (\,\mathrm{I}_N - K(t)\,\Phi^T(t)\,)\,\frac{P(t-1)}{\mu} \tag{3.9}$$

2. Hata denklemi,

$$e(t) = y(t) - \Theta^T(t-1)\,\Phi(t) \tag{3.10}$$

kullanılarak hata hesaplanır.

3. Aşağıdaki denklem kullanılarak $\hat{\Theta}^T(t)$ hesaplanır.

$$\hat{\Theta}^T(t) = \hat{\Theta}^T(t-1) + e(t)\,K(t)^T \tag{3.11}$$

Elde edilen $\hat{\Theta}^T(t)$ parametresinden rahatlıkla $A(q^{-1})$ ve $B(q^{-1})$ parametreleri çekilebilir.

### 3.2. Basit Genetik Algoritma Uyarlamalı Genelleştirilmiş Öngörülü Kontrol

Günümüzde optimizasyona olan talebin artması, problemlere hızlı ve kolay çözüm

veren yeni çözüm yöntemleri arayışına neden olmuştur. Özellikle sert (hard) optimizasyon teknikleri yerine, yumuşak hesaplama (soft computing) ve evrimsel algoritma (evolutionary algorithm) kullanımı ön plana çıkmıştır. Evrimsel yaklaşımlara dayanan genetik algoritmalar da, bu arayışlar içinde önemli bir yer tutmaya başlamıştır. Uygulama başarıları artan ve sürekli geliştirilmeye çalışılan genetik algoritmalar diğer yumuşak hesaplama yöntemleri ile birlikte kullanılarak hibrid (hybrid) çözümler geliştirilmesine çalışılmaktadır.

Genetik algoritmalar, doğal seçim ilkelerine dayanan bir arama ve optimizasyon yöntemidir. Temel ilkeleri John Holland tarafından ortaya atılmıştır [80]. Temel ilkelerinin ortaya atılmasından sonra, genetik algoritmalar hakkında birçok bilimsel çalışma yayınlanmıştır [51]. Genetik algoritmaların, fonksiyon optimizasyonu, çizelgeleme, mekanik öğrenme, tasarım, hücresel üretim gibi alanlarda başarılı uygulamaları bulunmaktadır. Geleneksel optimizasyon yöntemlerine göre farklılıkları olan genetik algoritmalar, parametre kümesini değil kodlanmış biçimlerini kullanırlar. Olasılık kurallarına göre çalışan genetik algoritmalar, yalnızca amaç fonksiyonuna gereksinim duyar. Çözüm uzayının tamamını değil belirli bir kısmını tararlar. Böylece, etkin arama yaparak çok daha kısa bir sürede çözüme ulaşırlar [51].

### 3.2.1. Basit genetik algoritma

Birçok alanda uygulama imkânı ve uygulamaları olan genetik algoritmaların işleme adımları şöyle açıklanabilir [81]:

1. Arama uzayındaki tüm mümkün çözümler dizi olarak kodlanır.
2. Genellikle rastsal bir çözüm kümesi seçilir ve başlangıç popülasyonu olarak kabul edilir.
3. Her bir dizi için bir uygunluk değeri hesaplanır, bulunan uygunluk değerleri dizilerin çözüm kalitesini gösterir.

4. Bir grup dizi belirli bir olasılık değerine göre rastsal olarak seçilip çoğalma işlemi gerçekleştirilir.

5. Yeni bireylerin uygunluk değerleri hesaplanarak, çaprazlama ve mutasyon işlemlerine tabi tutulur.

6. Önceden belirlenen kuşak sayısı boyunca yukarıdaki işlemler devam ettirilir.

7. İterasyon, belirlenen kuşak sayısına ulaşınca işlem sona erdirilir. Amaç fonksiyonuna göre en uygun olan dizi seçilir.

Genetik algoritmalar bir çözüm uzayındaki her noktayı, kromozom adı verilen ikili bit dizisi ile kodlar. Her noktanın bir uygunluk değeri vardır. Tek bir nokta yerine, genetik algoritmalar bir popülasyon olarak noktalar kümesini muhafaza eder. Her kuşakta, genetik algoritma, çaprazlama ve mutasyon gibi genetik operatörleri kullanarak yeni bir popülasyon oluşturur. Birkaç kuşak sonunda popülasyon daha iyi uygunluk değerine sahip üyeleri içerir. Bu, Darwin'in rastsal mutasyona ve doğal seçime dayanan evrim modellerine benzemektedir. Genetik algoritmalar, çözümlerin kodlanmasını, uygunlukların hesaplanmasını, çoğalma, çaprazlama ve mutasyon operatörlerinin uygulanmasını içerir [82].

#### 3.2.1.1. Çözümlerin kodlanması

Bir problemin çözümü için genetik algoritma geliştirmenin ilk adımı, tüm çözümlerin aynı boyutlara sahip bitler dizisi biçiminde gösterilmesidir. Dizilerden her biri, problemin olası çözümler uzayındaki rastsal bir noktayı simgeler [83]. Parametrelerin kodlanması, probleme özgü bilgilerin genetik algoritmanın kullanacağı şekle çevrilmesine olanak tanır [82].

#### 3.2.1.2. İlk popülasyonun oluşturulması

Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur. Çözüm grubu popülasyon,

çözümlerin kodları da kromozom olarak adlandırılır. İkili alfabenin kullanıldığı kromozomların gösteriminde, ilk popülasyonun oluşturulması için rastsal sayı üreticileri kullanılabilir. Rastsal sayı üreticisi çağrılır ve değer 0,5'den küçükse konum 0'a değilse 1 değerine ayarlanır [84]. Birey sayısının ve kromozom uzunluğunun az olduğu problemlerde yazı-tura ile de konum değerleri belirlenebilmektedir. Genetik algoritmalarda ikili kodlama yöntemi dışında, çözümü aranan probleme bağlı olarak farklı kodlama yöntemleri de kullanılmaktadır [51].

### 3.2.1.3. Uygunluk değerinin hesaplanması

Bir kuşak oluşturulduktan sonraki ilk adım, populasyondaki her üyenin uygunluk değerini hesaplama adımıdır. Örneğin, bir maksimizasyon problemi için i. üyenin uygunluk değeri f(i), genellikle o noktadaki amaç fonksiyonunun değeridir [82]. Çözümü aranan her problem için bir uygunluk fonksiyonu mevcuttur. Verilen belirli bir kromozom için uygunluk fonksiyonu, o kromozomun temsil ettiği çözümün kullanımıyla veya yeteneğiyle orantılı olan sayısal bir uygunluk değeri verir. Bu bilgi, her kuşakta daha uygun çözümlerin seçiminde yol göstermektedir. Bir çözümün uygunluk değeri ne kadar yüksekse, yaşama ve çoğalma şansı o kadar fazladır ve bir sonraki kuşakta temsil edilme oranı da o kadar yüksektir [83].

### 3.2.1.4. Çoğalma işleminin uygulanması

Çoğalma operatöründe diziler, amaç fonksiyonuna göre kopyalanır ve iyi kalıtsal özellikleri gelecek kuşağa daha iyi aktaracak bireyler seçilir. Üreme operatörü yapay bir seçimdir. Dizileri uygunluk değerlerine göre kopyalama, daha yüksek uygunluk değerine sahip dizilerin, bir sonraki kuşaktaki bir veya daha fazla yavruya daha yüksek bir olasılıkla katkıda bulunması anlamına gelmektedir. Çoğalma, bireyleri seçme işleminden, seçilmiş bireyleri bir eşleme havuzuna kopyalama işleminden ve havuzda bireyleri çiftler halinde gruplara ayırma işleminden oluşur [85].

Uygunluk değerinin hesaplanması adımından sonra mevcut kuşaktan yeni bir popülasyon yaratılmalıdır. Seçim işlemi, bir sonraki kuşak için yavru üretmek amacıyla hangi ailelerin yer alması gerektiğine karar vermektedir. Bu doğal

seçimdeki en uygunun yaşaması durumuna benzerdir. Bu yöntemin amacı, ortalama uygunluğun üzerindeki değerlere çoğalma fırsatı tanımaktır. Bir dizinin kopyalanma şansı, uygunluk fonksiyonuyla hesaplanan dizinin uygunluk değerine bağlıdır [82]. Seçim yöntemlerine rulet tekerleği seçimi, turnuva seçimi ve sıralama seçimi gibi seçim yöntemleri örnek verilebilir.

### 3.2.1.5. Çaprazlama işleminin uygulanması

Mevcut gen havuzunun potansiyelini araştırmak üzere, bir önceki kuşaktan daha iyi nitelikler içeren yeni kromozomlar yaratmak amacıyla çaprazlama operatörü kullanılmaktadır. Çaprazlama genellikle, verilen bir çaprazlama oranına eşit bir olasılıkla seçilen aile çeşitlerine uygulanmaktadır [82].

Genetik algoritmanın performansını etkileyen önemli parametrelerden biri olan çaprazlama operatörü doğal popülasyonlardaki çaprazlamaya karşılık gelmektedir. Çoğalma işlemi sonucunda elde edilen yeni popülasyondan rastsal olarak iki kromozom seçilmekte ve karşılıklı çaprazlama işlemine tabi tutulmaktadır. Çaprazlama işleminde dizi uzunluğu L olmak üzere, 1 < = k < = L-1 aralığında k tamsayısı seçilmektedir. Bu tamsayı değerine göre dizi çaprazlamaya uğratılır. En basit çaprazlama yöntemi tek noktalı çaprazlama yöntemidir. Tek noktalı çaprazlama yapılabilmesi için her iki kromozomun da aynı gen uzunluğunda olması gerekir. İki noktalı çaprazlamada ise kromozom iki noktadan kesilir ve karşılıklı olarak pozisyonlar yer değiştirilir [85].

### 3.2.1.6. Mutasyon işleminin uygulanması

Çaprazlama mevcut gen potansiyellerini araştırmak üzere kullanılır. Fakat popülasyon gerekli tüm kodlanmış bilgiyi içermez ise, çaprazlama tatmin edici bir çözüm üretemez. Bundan dolayı, mevcut kromozomlardan yeni kromozomlar üretme yeteneğine sahip bir operatör gerekmektedir. Bu görevi mutasyon gerçekleştirir. Yapay genetik sistemlerde mutasyon operatörü, bir daha elde edilemeyebilir iyi bir çözümün kaybına karşı koruma sağlamaktadır [51]. İkili kodlama sisteminin kullanıldığı problemlerde mutasyon, düşük bir olasılık değeri altında bir bit değerini

(0 veya 1 olabilir) diğer bit değerine dönüştürür. İkili kodlama sisteminin kullanılmadığı problemlerde ise sıra değiştirme, bit ters çevirme gibi farklı mutasyon yöntemleri kullanılmaktadır. Hangi yöntem kullanılırsa kullanılsın, mutasyonun genel amacı, genetik çeşitliliği sağlamak veya korumaktır [86].

#### 3.2.1.7. Yeni kuşağın oluşması ve döngünün durdurulması

Yeni kuşak çoğalma, çaprazlama ve mutasyon işlemlerinden sonra tanımlanmakta ve bir sonraki kuşağın ebeveynleri olmaktadırlar. Süreç yeni kuşakla çoğalma için belirlenen uygunluk ile devam eder. Bu süreç, önceden belirlenen kuşak sayısı kadar veya bir hedefe ulaşılıncaya kadar ya da başka bir durdurma kriteri sağlanana kadar devam eder [84]. İstenen hassasiyet derecesine göre de maksimum iterasyon sayısı belirlenebilmekte ve iterasyon bu sayıya ulaştığında döngü durdurulabilmektedir. Durdurma kriteri iterasyon sayısı olabileceği gibi hedeflenen uygunluk değeri de olabilmektedir [87].

### 3.2.2. Genetik algoritmalarda parametre seçimi

Parametreler, genetik algoritma performansı üzerinde önemli etkiye sahiptir. Optimal kontrol parametreleri bulmak için birçok çalışma yapılmıştır fakat tüm problemler için genel olarak kullanılabilecek parametreler bulunamamıştır [88]. Bu parametreler, kontrol parametreleri olarak adlandırılmaktadır. Kontrol parametreleri popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, kuşak aralığı, seçim stratejisi ve fonksiyon ölçeklemesi olarak sayılabilir. Bu parametreler aşağıda açıklanmıştır [83, 89]:

1. Popülasyon Büyüklüğü: Genetik algoritma kullanıcısı tarafından verilen en önemli kararlardan birisidir. Bu değer çok küçük olduğunda, genetik algoritma yerel bir optimuma takılabilmektedir. Popülasyonun çok büyük olması ise çözüme ulaşma zamanını arttırmaktadır. Bu konuda Goldberg 1989'da, yalnızca kromozom uzunluğuna bağlı bir popülasyon büyüklüğü hesaplama yöntemi önermiştir [82]. Ayrıca 20–30 arası bir popülasyon büyüklüğünün iyi sonuçlar verdiğini belirtmiştir.

2. Çaprazlama Olasılığı: Çaprazlamanın amacı, mevcut iyi kromozomların özelliklerini birleştirerek daha uygun kromozomlar yaratmaktır. Kromozom çiftleri P(c) olasılığı ile çaprazlamaya uğramak üzere seçilirler. Bu parametre çaprazlamanın ne kadar sıklıkla yapılacağını belirtir. Eğer herhangi bir çaprazlama yoksa yavrular ataların aynısı olacaktır. Eğer bir çaprazlama yapılırsa yavrular ataların parçalarından oluşur. Eğer çaprazlama olasılığı %100 ise yavrular tamamen çaprazlama ile yapılır. Eğer %0 ise yavrular ataların kromozomlarının aynısına sahip olurlar. Çaprazlamanın artması, yapı bloklarının artmasına neden olmakta fakat aynı zamanda bazı iyi kromozomların da bozulma olasılığını arttırmaktadır.
3. Mutasyon Olasılığı: Mutasyonun amacı popülasyondaki genetik çeşitliliği korumaktır. Sınırlı bir popülasyon üzerinde popülasyondaki birkaç genetik bilginin erkenden kaybolma ihtimali vardır. Örneğin bir kromozomun genlerinin tamamı 0 ya da 1 olabilir. Böyle bir kromozomu çaprazlama operatörü ile değiştirmek mümkün değildir. Ya da uygunluk değeri yüksek olan kromozomları bozma ihtimali vardır. Bu sebeple çaprazlama ile üretilemeyen kromozomları mutasyon ile üretmek mümkündür.

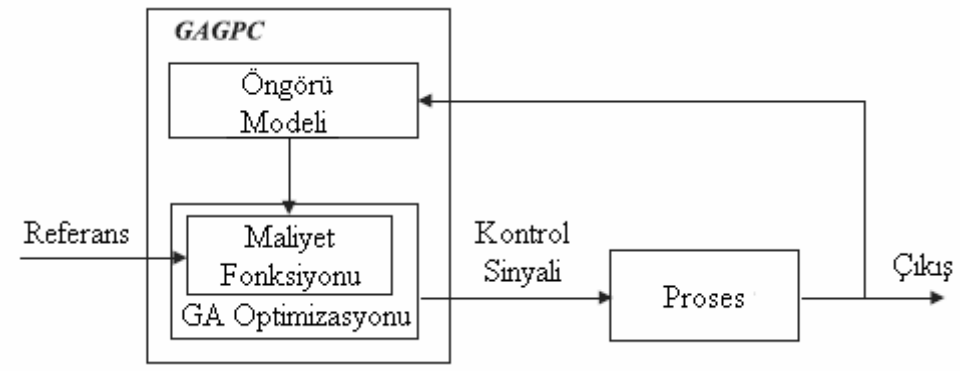
   Mutasyon P(m) olasılığı ile bir kromozomdaki her bitte meydana gelebilir. Kromozom parçalarının ne kadar sıklıkla mutasyon geçireceğini belirtir. Eğer mutasyon yoksa yavrular çaprazlamadan hemen sonra değiştirilmeden üretilir (veya doğrudan kopyalanır). Eğer mutasyon varsa, yavruların kromozomlarının bir veya daha fazla parçası değişir. Eğer mutasyon olasılığı %100 ise tüm kromozom değişecektir. %0 ise hiçbir şey değişmez. Mutasyon genellikle GA'nın yerel aşırılıklara düşmesini engeller. Mutasyonlar çok sık oluşmamalıdır, çünkü GA rasgele aramaya dönüşebilir. Eğer mutasyon olasılığı artarsa, genetik arama rastsal bir aramaya dönüşür.
4. Kuşak Aralığı: Her kuşaktaki yeni kromozom oranına kuşak aralığı denmektedir. Genetik operatörler için kaç tane kromozomun seçildiğini gösterir. Yüksek bir değer birçok kromozomun yer değiştirdiği anlamına gelmektedir.
5. Seçim Stratejisi: Eski kuşağı yenilemenin çeşitli yöntemleri mevcuttur. Kuşaksal stratejide, mevcut popülasyondaki kromozomlar tamamen yavrular ile yer değiştirir. Popülasyonun en iyi kromozomu da yenilendiğinden dolayı

bir sonraki kuşağa aktarılamaz ve bu yüzden bu strateji en uygun (elitist) stratejisiyle beraber kullanılmaktadır. En uygun stratejisinde, popülasyondaki en iyi kromozomlar hiçbir zaman yenilenmemektedir, bundan dolayı çoğalma için en iyi çözüm her zaman elverişlidir. Denge durumu stratejisinde ise, her kuşakta yalnızca birkaç kromozom yenilenmektedir. Genellikle, yeni kromozomlar popülasyona katıldığında en kötü kromozomlar yenilenir.

### 3.2.3. Basit genetik algoritma uyarlamalı genelleştirilmiş öngörülü kontrol

Bölüm 3.1 ‘de GPC algoritmasının yapısında bulunan maliyet fonksiyonu ayrıntılı bir şekilde anlatılmıştır. GPC ‘nin temel yapı taşı olan bu fonksiyonun minimize edilmesi kontrolün başarılı olması için son derece önemlidir.

Öte yandan optimizasyon tekniği olarak geniş bir uygulama alanı bulan genetik algoritmalar, fonksiyon minimizasyon işlemlerinde de kullanılabilmektedir. Bu bölümde anlatılan SGA-GPC (Basit Genetik Algoritma uyarlamalı GPC) algoritmasında, genetik algoritma GPC ‘ye uyarlanarak maliyet fonksiyonunun minimize edilmesinde kullanılmıştır. Basit Genetik Algoritma uyarlamalı GPC ‘nin yapısı Şekil 3.3‘de gösterilmiştir.




Şekil 3.3. Genetik algoritma uyarlamalı GPC ’nin yapısı

GPC ‘ye ait maliyet fonksiyonu SISO ve MIMO yapılar için sırasıyla aşağıdaki gibidir:

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \left( \hat{y}(t+j) - y_r(t+j) \right)^2 + \sum_{j=1}^{N_u} \lambda(j) \left( \Delta u(t+j-1) \right)^2 \quad (3.12)$$

$$\begin{aligned} J(N_1, N_2, N_u) &= \sum_{j=N_1}^{N_2} \left( \hat{y}(t+j) - y_r(t+j) \right)^T \left( \hat{y}(t+j) - y_r(t+j) \right) \\ &+ \sum_{j=1}^{N_u} \left( \Delta u(t+j-1) \right)^T \Lambda(j) \left( \Delta u(t+j-1) \right) \end{aligned} \quad (3.13)$$

GPC algoritmasında olduğu gibi buradaki amaç kontrolün başından sonuna kadar her kontrol adımı sonunda sisteme uygulanacak uygun $\Delta u(t)$'yi bulmaktır. Güçlü bir kontrol için referans yörüngenin çok yakın takip edilmesi istenir. Bunun için maliyet fonksiyonu $J$ minimize edilir. Yani maliyet fonksiyonu $J$ 'yi sıfıra yapan ya da sıfıra yakınsatan uygun tork değişimi $\Delta u(t)$ hesaplanır. SGA-GPC algoritması istenilen tork değişimini hesaplayabilmek için genetik operatörleri kullanır ve genetik hesaplama sürecindeki işlemleri gerçekler. Uygulama için yapılan bu genetik işlemler süreci aşağıda anlatılmıştır.

### 3.2.3.1. Başlangıç popülasyonu

Basit Genetik Algoritma (SGA) işlem basamaklarındaki işlemler sırası ile gerçeklenerek en uygun tork değişimi hesaplanır. Bunun için ilk olarak başlangıç popülasyonu oluşturulur. Başlangıç popülasyonu için eklemlere uygulanacak tork değerleri göz önünde bulundurulmuştur. Buna göre eklemlere maximum -186 ile + 186 Nm değerleri arasında tork uygulanabilmektedir. Dolayısıyla kontrol giriş sinyali için bu aralıkta arama yapılacaktır.

### 3.2.3.2. Kodlama

Bir kromozom temsil ettiği çözüm hakkında bir şekilde bilgi içermelidir. En çok kullanılan kodlama ikili karakter dizisidir. Her kromozom ikili karakter dizisi şeklinde temsil edilmektedir. Karakter dizisindeki her bit çözümün bir özelliğini temsil eder. Bu uygulamada popülasyonundaki her birey için 21-bit uzunluğunda kromozomlar kullanılmıştır.

- İkili kodlama 21-bit ($2^{22}$-1=4.194.302)
Tork sınır değerleri için;
-186,0000: 00 0000 0000 0000 0000 0000
0,0000: 01 1100 0110 0001 1010 0000
+186,0000:11 1100 1100 0011 0100 0000
kodlama sistemi kullanılmıştır.

### 3.2.3.3. Uygunluk değeri

Uygunluk değeri, her bir bireyin amaç fonksiyonuna ne kadar yaklaştığını gösteren bir parametre olup, popülasyon kümesinde sonraki nesillere aktarılacak bireylerin seçimi için kullanılır. Genetik algoritmanın en önemli yapı taşıdır.

Burada genetik algoritma GPC algoritmasında maliyet fonksiyonu optimizasyonu için kullanıldığından bireylerin yani eklemlere uygulanacak tork için uygun tork değişim değerlerinin bu fonksiyona göre belirlenmesi gerekir. Bireye denk gelen tork değişimi maliyet fonksiyonunda yazılarak elde edilen sonuç değeri o bireyin uygunluğunu belirleyecektir. Ancak optimizasyondaki amaç maliyet fonksiyonunu minimize etmek olduğundan, uygunluk değeri daha küçük değere sahip olan yani maliyet fonksiyonundaki değeri düşük bireylerin sonraki nesillere aktarılması gerekir. SISO ve MIMO yapılar için uygunluk fonksiyonları denklem 3.14 ve 3.15 ‘de tanımlanmıştır.

$$f(x)_{uygunluk-siso} = J_{uygunluk} = \sum_{j=N_1}^{N_2} \left( \hat{y}(t+j) - y_r(t+j) \right)^2 + \sum_{j=1}^{N_u} \lambda(j) \left( \Delta u(t+j-1) \right)^2 \quad (3.14)$$

$$\begin{aligned} f(x)_{uygunluk-mimo} = J_{uygunluk} &= \sum_{j=N_1}^{N_2} \left( \hat{y}(t+j) - y_r(t+j) \right)^T \left( \hat{y}(t+j) - y_r(t+j) \right) \\ &+ \sum_{j=1}^{N_u} \left( \Delta u(t+j-1) \right)^T \Lambda(j) \left( \Delta u(t+j-1) \right) \end{aligned} \quad (3.15)$$

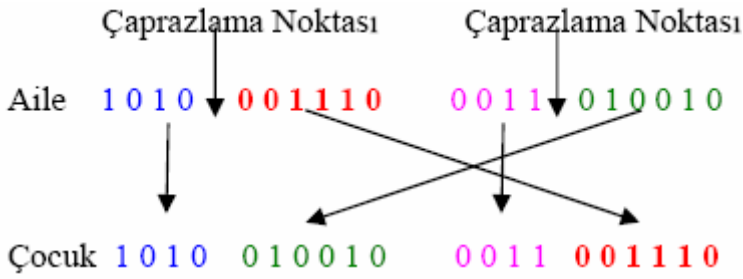
Uygunluk değerlerine göre bireyler küçükten büyüğe doğru sıralanarak uygunluk değeri büyük bireyler yani başarısı iyi olmayan bireyler elenecektir. Elenenlerin yerine uygunluk değeri daha küçük olan bireylerin birkaç kopyası yapılır.

### 3.2.3.4. Seçme ve yeniden oluşturma

Popülasyon kümesindeki her bir kromozomun uygunluk fonksiyonu hesaplandıktan sonra uygunluk değerlerine göre küçükten büyüğe sıralanarak en kötü bireyler popülasyondan atılır. Seçme işlemi, çaprazlama ve mutasyon işlemlerine uğratılacak bireyleri belirleme aşamasıdır. Seçme operatörü olarak uygunluk değeri oranı kullanılmıştır. Buna göre uygunluk değerleri yüksek olan yani maliyet fonksiyonu değeri yüksek olan bireyler popülasyondan atılırlar ve bu bireylerin yerine yavru bireyler alınır.

### 3.2.3.5. Çaprazlama

Çaprazlama, atalardaki seçili genler üzerinde işlem yapar ve yeni yavrular oluşturur. Bunun en basit şekli, rasgele bir kesme noktası (çaprazlama noktası) seçip, bu noktadan önceki her biti ilk atadan, sonraki her biti ikinci atadan alıp birleştirerek yavruyu oluşturmaktır. Çaprazlama aşağıdaki şekilde gösterilebilir:

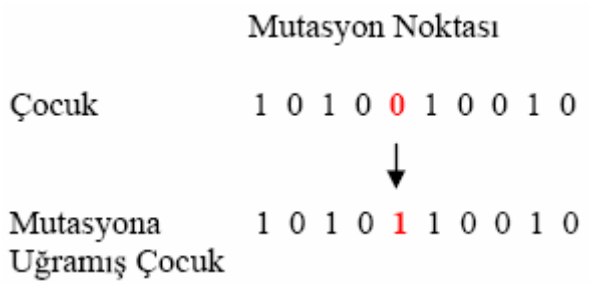



Şekil 3.4. Tek nokta çaprazlama

### 3.2.3.6. Mutasyon

Çaprazlama işlemi gerçekleştirildikten sonra, mutasyon işlemi yapılır. Mutasyonun amacı, toplumdaki tüm çözümlerin çözülen problemlerin bir yerel uygun değerine düşmesinin önüne geçmektir. Mutasyon işlemi çaprazlama sonucu oluşan yavruyu rasgele değiştirmektedir. Kromozomdaki genlerden rasgele birini değiştirme ile

gerçeklenebilir. Şekil 3.5'de gen takısı 5 olan bir mutasyon örneği verilmiştir. Seçme, çaprazlama ve mutasyon işlemlerine tabii tutulan bireyler yeni popülasyona atanırlar. Artık yeni neslin bireyleridir.




Şekil 3.5. Mutasyon işlemi

### 3.2.3.7. Döngünün durdurulması

Yeni neslin çoğalması, çaprazlama ve mutasyon süreci önceden belirlenen kuşak sayısı kadar veya bir hedefe ulaşılıncaya kadar ya da başka bir durdurma kriteri sağlanana kadar devam eder. İstenen hassasiyet derecesine göre de maksimum iterasyon sayısı belirlenebilmekte ve iterasyon bu sayıya ulaştığında döngü durdurulabilmektedir. Durdurma kriteri iterasyon sayısı olabileceği gibi hedeflenen uygunluk değeri de olabilmektedir. Bu uygulamada durdurma kriteri 100 iterasyon olarak seçilmiştir. 100 iterasyon tamamlandığında o ana kadar hesaplanmış en iyi kromozom sonuç olarak alınır.

Sonuç olarak, belirlenen en iyi kromozoma karşılık gelen tork değeri, tork değişimi $\Delta u(t)$ olarak atanır ve değeri bir önceki kontrol girdisi ile toplanarak yeni kontrol girdisi ($u(t) = u(t-1) + \Delta u(t)$) elde edilir ve sisteme uygulanır.

## 3.3. Nöro Genelleştirilmiş Öngörülü Kontrol

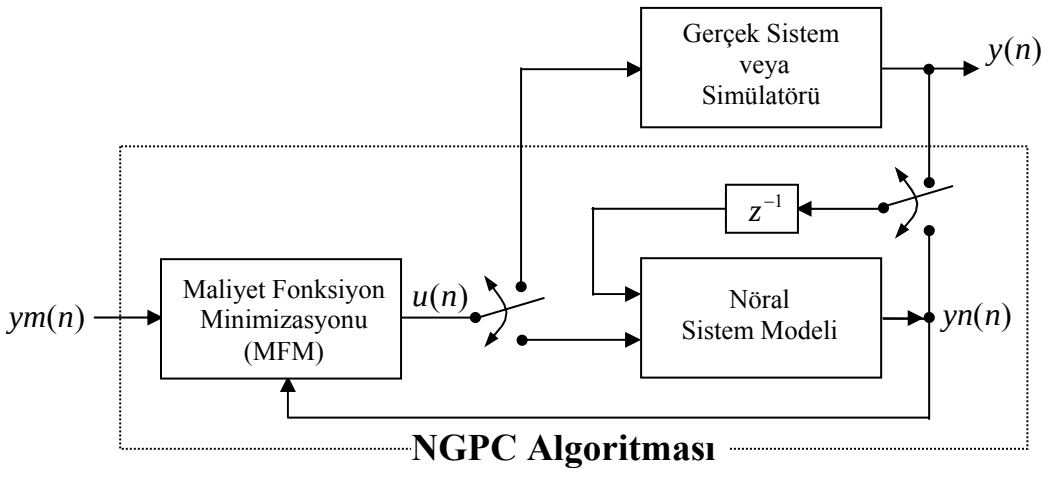
GPC algoritması, doğrusal öngörülü sistem modeline sahip bir yapıda geliştirilmiştir. Bu model, yapı itibariyle kontrol edilecek sistemden tamamen bağımsız olup adaptif kontrol kanunu gereği sistemle beraber eşzamanlı çalışmaktadır. Bu model yerine

uygun bir doğrusal olmayan model kullanılırsa, GPC algoritmasının doğru tahmin etmekteki kabiliyeti artırılabilir.

Şimdiye kadar kullanılan doğrusal olmayan sistem modelleri içerisinde en uygun olanları yapısında yapay sinir ağ, diğer bir deyişle nöral ağ bulunduran modeller olmuştur [90]. Bu model yapısına sahip GPC algoritmalarına Nöro Genelleştirilmiş Öngörülü Kontrol ( Neural Generalized Predictive Control - NGPC ) algoritmaları denir. NGPC algoritmaları yüksek oranda doğrusal olmayan özellikler gösteren sistemler için bile rahatlıkla kullanılabilmektedir.

Bu tezde anlatılacak olan NGPC algoritması Newton-Raphson güncelleme metodu uyarlamalı bir algoritma olup ilk olarak D. Soloway ve P. J. Haley tarafından 1996 yılında tanıtılmıştır [56, 63-65].

Newton-Raphson uyarlamalı NGPC algoritması istenen referans yörüngeye yakınsama bakımından GPC algoritmasına kıyasla daha hassastır. Newton-Raphson uyarlamalı NGPC algoritması bir nöral sistem modeli ile maliyet fonksiyonunu minimize eden bir algoritmanın birleştirilmesinden meydana gelmiş olup blok diyagramı Şekil 3.6'da verilmiştir.




Şekil 3.6. NGPC algoritmasının blok diyagramı

NGPC algoritmasında kullanılan MFM bloğu, her kontrol adımında maliyet fonksiyonunu tekrardan minimize ederek sistem için uygun kontrol girdilerini üretir.

Bu kontrol girdileri kontrolün başından sonuna kadar sistem çıkışını istenen çıkış yörüngesine yakınsayacak şekilde devamlı değişir.

### 3.3.1. Maliyet fonksiyonu

Her kontrol adımında minimize edilerek sisteme uygun girdi sağlamakta kullanılan maliyet fonksiyonu, NGPC algoritmasının temelini teşkil eder. Denklemi,

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \left( yn(n+j) - y_r(n+j) \right)^2 + \sum_{j=1}^{N_u} \lambda(j) \left( \Delta u(n+j) \right)^2 + \sum_{j=1}^{N_u} \left( \frac{s}{u(n+j) - u_{\min} + \varepsilon} + \frac{s}{u_{\max} - u(n+j) + \varepsilon} \right) \tag{3.16}$$

şeklindedir.

Burada,

$N_1$ : Öngörü ufuğunun başlangıcı

$N_2$ : Öngörü ufuğunun bitimi

$N_u$ : Kontrol ufuğu

$\lambda$ : Kontrol giriş ağırlık faktörü

$\Delta$ : Fark alma operatörü

$y_r(n+j)$ : Öngörü ufuğu boyunca ( $j = N_1, \cdots, N_2$ ) referans yörünge

$yn(n+j)$ : Öngörü ufuğu boyunca ( $j = N_1, \cdots, N_2$ ) öngörülen nöral ağ çıkışları

$u(n+j)$ : Kontrol ufuğu boyunca ( $j = 1, \cdots, N_u$ ) öngörülen kontrol girişleri

$u_{\min}$ ve $u_{\max}$ : Kontrol girişinin alabileceği minimum ve maksimum değerler

Maliyet fonksiyonunun $N_1$, $N_2$, $N_u$ ve $\lambda$ olmak üzere dört adet ayar parametresi mevcuttur. $s$ ve $\varepsilon$ parametreleri ise yapay sinir ağının kilitlenip çalışamamasını önlemek içindir, çok küçük seçilirler. Bu tez çalışmasında $s = 10^{-21}$, $\varepsilon = 10^{-7}$ değerleri kullanılmıştır.

Gerçek zamanlı ve hassas bir kontrol için maliyet fonksiyonunu minimize etmekte kullanılacak olan iyileştirme yönteminin uygun seçilmesi oldukça önemlidir. Bu seçimde üç önemli kriter vardır. Bu kriterler;

1. Minimize için tekrar sayısının azlığı
2. Hesapsal işlemlerin azlığı
3. İstenen sonuca yakınsama

Non-gradient [91], Simplex [92], Successive Quadratic Programming [93, 94] ve Newton-Raphson [63-65] gibi pek çok yöntem maliyet fonksiyonunu minimize etmekte kullanılmaktadır. Bu yöntemler, yakınsama için çok tekrar gerektirdiğinden gerçek zamanlı kontrolü güçleştirmektedir. Newton-Raphson iyileştirme yönteminde yakınsamadaki tekrar sayısı bu yöntemlere göre daha azdır. Ayrıca yakınsaması çok daha hassas olduğundan güçlü bir kontrol sağlar. Yakınsama için iki veya üç tekrar yeterlidir. Newton-Raphson iyileştirme yönteminde hesapsal yükü en çok Hessian kullanır. Fakat tekrar sayısının azlığı bu sorunu önemsiz kılar [63, 64].

Newton-Raphson uyarlamalı NGPC algoritmasında altı adet işlem basamağı vardır. Bu adımlar şunlardır:

1. Referans yörünge oluşturulur.
2. Önceki hesaplanan kontrol giriş vektörü ile başlanır ve nöral sistem modeli kullanılarak sistemin performansı öngörülür.
3. Maliyet fonksiyonunu minimize eden yeni kontrol giriş vektörü hesaplanır.
4. İstenen minimize sağlanana kadar adım 2 ve 3 tekrarlanır.
5. Kontrol giriş vektöründeki ilk kontrol girişi sisteme uygulanır.
6. Proses bitene kadar bütün adımlar tekrarlanır.

### 3.3.2. Maliyet fonksiyonunun minimize edilmesi

Bu tez çalışmasında, maliyet fonksiyonunun minimize edilmesinde iyileştirme yöntemi olarak yakınsamadaki tekrar sayısının azlığı nedeniyle Newton-Raphson tercih edilmiştir. Newton-Raphson yakınsama denklemi aşağıda gibi tanımlanmaktadır:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{3.17}$$

Denklem 3.16 'da verilen maliyet fonksiyonu ( $J$ ) ifadesinin güncellenmesi için Newton-Raphson güncelleme denklemi aşağıdaki gibi uyarlanır:

Kontrol giriş vektörü,

$$U(k) = \begin{pmatrix} u(n+1) \\ u(n+2) \\ \vdots \\ u(n+N_u) \end{pmatrix}, \quad k = 1, \cdots, \#tekrar$$

$U(k+1)$ için Newton-Raphson güncelleme denklemi,

$$U(k+1) = U(k) - \left( \frac{\partial^2 J}{\partial U^2}(k) \right)^{-1} \left( \frac{\partial J}{\partial U}(k) \right) \tag{3.18}$$

olur. Burada Jacobian;

$$\frac{\partial J}{\partial U}(k) = \begin{pmatrix} \frac{\partial J}{\partial u(n+1)} \\ \vdots \\ \frac{\partial J}{\partial u(n+N_u)} \end{pmatrix}$$

olup $N_u$ tane elemanı olan bir sütun matristir. Jacobian sütun matrisinin her bir elemanının hesabı için aşağıdaki denklem kullanılır. ( $h = 1, ..., N_u$ )

$$\begin{aligned} \frac{\partial J}{\partial u(n+h)} &= 2 \sum_{j=N_1}^{N_2} \left( yn(n+j) - ym(n+j) \right) \frac{\partial yn(n+j)}{\partial u(n+h)} \\ &+ 2 \sum_{j=1}^{N_u} \lambda(j) \left( u(n+j) - u(n+j-1) \right) \left( \frac{\partial u(n+j)}{\partial u(n+h)} - \frac{\partial u(n+j-1)}{\partial u(n+h)} \right) \\ &+ \sum_{j=1}^{N_u} \frac{\partial u(n+j)}{\partial u(n+h)} \left( \frac{-s}{\left( u(n+j) - u_{\min} + \varepsilon \right)^2} + \frac{s}{\left( u_{\max} - u(n+j) + \varepsilon \right)^2} \right) \end{aligned} \tag{3.19}$$

Hessian ise,

$$\frac{\partial^2 J}{\partial U^2}(k) = \begin{pmatrix} \frac{\partial^2 J}{\partial u(n+1)^2} & \cdots & \frac{\partial^2 J}{\partial u(n+1)\partial u(n+N_u)} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial u(n+N_u)\partial u(n+1)} & \cdots & \frac{\partial^2 J}{\partial u(n+N_u)^2} \end{pmatrix}$$

olup $N_u \times N_u$ tane elemanı olan simetrik bir kare matristir. Hessian'nin her bir elemanının hesabı için aşağıdaki denklem kullanılır. ( $h = 1, ..., N_u$ ve $m = 1, ..., N_u$ )

$$\frac{\partial^2 J}{\partial u(n+m)\,\partial u(n+h)} =$$

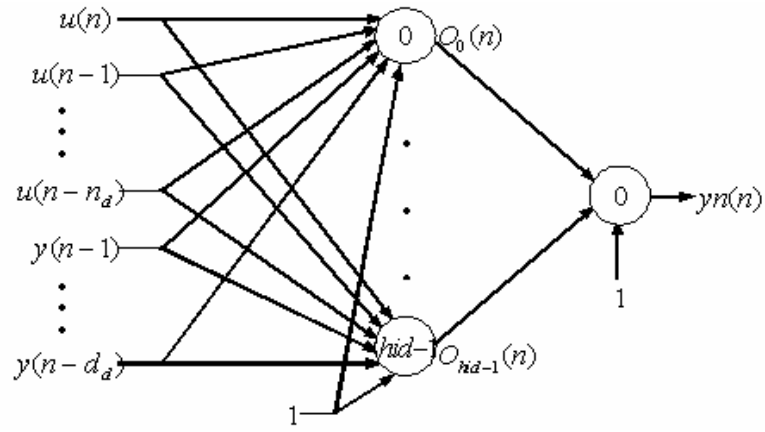
$$2\sum_{j=N_1}^{N_2}\left(\frac{\partial yn(n+j)}{\partial u(n+m)}\frac{\partial yn(n+j)}{\partial u(n+h)} + \big(yn(n+j) - ym(n+j)\big)\frac{\partial^2 yn(n+j)}{\partial u(n+m)\partial u(n+h)}\right)$$

$$+2\sum_{j=1}^{N_u}\lambda(j)\left(\frac{\partial u(n+j)}{\partial u(n+m)} - \frac{\partial u(n+j-1)}{\partial u(n+m)}\right)\left(\frac{\partial u(n+j)}{\partial u(n+h)} - \frac{\partial u(n+j-1)}{\partial u(n+h)}\right) \tag{3.20}$$

$$+2\sum_{j=1}^{N_u}\frac{\partial u(n+j)}{\partial u(n+m)}\frac{\partial u(n+j)}{\partial u(n+h)}\left(\frac{s}{\big(u(n+j) - u_{\min} + \varepsilon\big)^3} + \frac{s}{\big(u_{\max} - u(n+j) + \varepsilon\big)^3}\right)$$

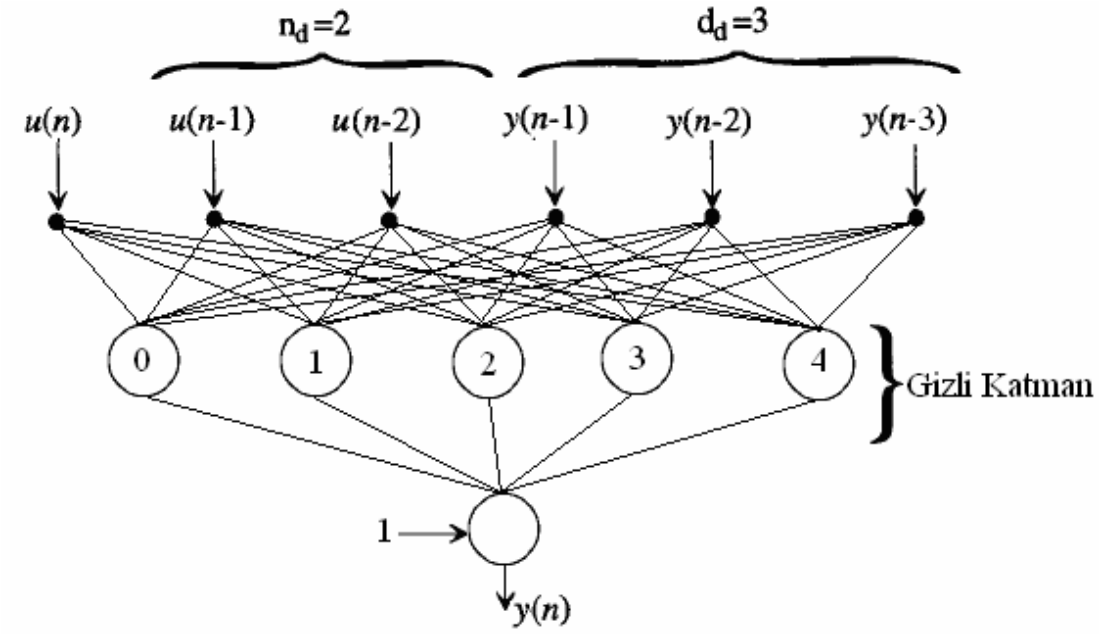
### 3.3.3. Kullanılan yapay sinir ağ modeli

NGPC algoritmasında öngörülerin yapılması için bir yapay sinir ağı kullanılır. Bu ağ, sisteme uygulanan geçmiş zaman dilimlerindeki girdileri ( $n_d + 1$ tane) ve sistemden alınan geçmiş zaman dilimlerindeki çıktıları ( $d_d$ tane) kullanarak sistemin o anki ürettiği çıktıya yakınsama yapar. $n_d$ ve $d_d$ değerlerinin büyük seçilmesinin fazla bir faydası yoktur. Bununla beraber, işlem yükünü artıracağından gerçek zamanlı kontrolü zorlaştıracaktır. Kullanılan YSA modeli Şekil 3.7'de gösterilmiştir.




Şekil 3.7. NGPC algoritmasında kullanılan yapay sinir ağ modeli

Bu tez çalışmasında NGPC algoritması için Şekil 3.8'de gösterilen üç katmanlı bir yapay sinir ağı kullanılmıştır. Giriş katmanında altı giriş, çıkışta ise bir çıkış mevcuttur. Ara katmanda beş adet nöron kullanılmıştır.




Şekil 3.8. Tasarlanan YSA modeli

Sistem modelinde kullanılan ağ denklemleri (3.21), (3.22) ve (3.23)'de verilmiştir. Her biri bir öncekinin sonucunu kullanır. Denklemlerden anlaşıldığı gibi ara katmanda tan-sigmoid transfer fonksiyonu, çıkış katmanında ise lineer transfer fonksiyonu kullanılmaktadır. Çıkış katmanında lineer transfer fonksiyonu

kullanılmasının sebebi işlemleri azaltmaktır. Ayrıca, işlem yükünü azaltmak için ara katmandaki nöron sayısı öngörüye zarar vermeyecek şekilde küçük tutulabilir.

$$net_j(n) = b_j + \sum_{i=0}^{n_d} w_{j,i} u(n-i) + \sum_{i=1}^{d_d} w_{j,i+n_d} y(n-i) \tag{3.21}$$

$$O_j(n) = f\left(net_j(n)\right) = \frac{e^{net_j(n)} - e^{-net_j(n)}}{e^{net_j(n)} + e^{-net_j(n)}} = 1 - \frac{2}{1 + e^{2 \cdot net_j(n)}} \tag{3.22}$$

$$yn(n) = b + \sum_{j=0}^{hid-1} w_j O_j(n) \tag{3.23}$$

Hata fonksiyon denklemi (3.24)'de verilmiş olup hatayı minimize edecek şekilde ağırlıkları güncellenmede kullanılacak denklemlerin çıkarılması için gereklidir.

$$E = \frac{1}{2}\left( y(n) - yn(n) \right)^2 \tag{3.24}$$

Ağırlık güncellemede işlem kolaylığı olsun diye ağ gidileri

$$x(n) = \left( u(n), u(n-1), \cdots, u(n-n_d), y(n-1), \cdots, y(n-d_d) \right)$$

şeklinde satır matrisi olarak alınır. Her kontrol adımında ağ tekrardan eğitilir ve eğitim hata istenen seviyeye düşene kadar tekrarlanır. Eğitim tek data için olduğundan çok hızlıdır. Yani eğitim için geçen zaman ihmal edilebilir. Ağırlık değişimleri her eğitim döngüsü için (3.25), (3.26), (3.27) ve (3.28)'deki denklemler kullanılarak hesaplanır.

$$\Delta w_j(t+1) = -\eta \frac{\partial E}{\partial w_j} = \eta \left( y(n) - yn(n) \right) O_j(n) \tag{3.25}$$

$$\Delta b(t+1) = -\eta \frac{\partial E}{\partial b} = \eta \left( y(n) - yn(n) \right) \tag{3.26}$$

$$\Delta w_{j,i}(t+1) = -\eta \frac{\partial E}{\partial w_{j,i}} = \eta \left( y(n) - yn(n) \right) \left( 1 - O_j(n)^2 \right) w_j \; x_i(n) \tag{3.27}$$

$$\Delta b_j(t+1) = -\eta \frac{\partial E}{\partial b_j} = \eta \left( y(n) - yn(n) \right) \left( 1 - O_j(n)^2 \right) w_j \tag{3.28}$$

Elde edilen ağırlık değişim değerleri (3.29), (3.30), (3.31) ve (3.32)'deki denklemlerde yerlerine konularak ağırlıklar güncellenir.

$$w_j(t+1) = w_j(t) + \Delta w_j(t+1) \tag{3.29}$$

$$b(t+1) = b(t) + \Delta b(t+1) \tag{3.30}$$

$$w_{j,i}(t+1) = w_{j,i}(t) + \Delta w_{j,i}(t+1) \tag{3.31}$$

$$b_j(t+1) = b_j(t) + \Delta b_j(t+1) \tag{3.32}$$

### 3.3.4. Yapay sinir ağ modelinde öngörme

NGPC algoritması sistemin davranışını taklit edebilmek ve bu taklidi kullanarak öngörüler yapabilmek için yukarıda da bahsedildiği gibi nöral sistem modeli kullanır. Bu nöral sistem modeli sisteme uygulanan geçmiş kontrol adımlarındaki girdileri ve bu girdilere karşılık sistemin verdiği çıktıları kullanarak önce sistemi kendi içinde modeller. Sonra bu modeli kullanarak gelecek zaman dilimleri için öngörüler yapar. Bütün bu işlemler her kontrol adımda tekrarlanır. Öngörü denklemeleri aşağıda verilmiştir.

$$\begin{aligned} net_j(n+k) = b_j + \sum_{i=0}^{n_d} w_{j,i} \begin{cases} u(n+k-i) & i > k - N_u \\ u(n+N_u) & i \le k - N_u \end{cases} \\ + \sum_{i=1}^{\min(k-1,d_d)} w_{j,i+n_d}\, yn(n+k-i) + \sum_{i=k}^{d_d} w_{j,i+n_d}\, y(n+k-i) \end{aligned} \tag{3.33}$$

$$O_j(n+k) = f\left(net_j(n+k)\right) = 1 - \frac{2}{1+e^{2\cdot net_j(n+k)}} \tag{3.34}$$

$$yn(n+k) = b + \sum_{j=0}^{hid-1} w_j O_j(n+k) \tag{3.35}$$

Bu üç denklem kullanılarak aşağıdaki türev denklemleri ( Denklem 3.36'dan Denklem 3.43'e kadar ) elde edilir. Elde edilen bu denklemler Newton-Raphson güncelleme denkleminde yerlerine konulursa uygun kontrol giriş vektörü kolaylıkla hesaplanır. Bu hesaplama işlemi bir kaç kere tekrarlanırsa kontrol giriş vektörü daha hassas olur. Kontrol giriş vektöründeki ilk kontrol girişi sisteme uygulanır.

$$\frac{\partial f\left(net_j(n+k)\right)}{\partial net_j(n+k)} = 1 - O_j(n+k)^2 \tag{3.36}$$

$$\frac{\partial^2 f\left(net_j(n+k)\right)}{\partial net_j(n+k)^2} = -2\,O_j(n+k)\left(1 - O_j(n+k)^2\right) \tag{3.37}$$

$$\frac{\partial net_j(n+k)}{\partial u(n+h)} = \sum_{i=0}^{n_d} w_{j,i} \begin{cases} \dfrac{\partial u(n+k-i)}{\partial u(n+h)} & i > k - N_u \\ \dfrac{\partial u(n+N_u)}{\partial u(n+h)} & i \le k - N_u \end{cases} + \sum_{i=1}^{\min(k-1,d_d)} w_{j,i+n_d} \frac{\partial yn(n+k-i)}{\partial u(n+h)} \tag{3.38}$$

$$\frac{\partial^2 net_j(n+k)}{\partial u(n+m)\,\partial u(n+h)} = \sum_{i=1}^{\min(k-1,d_d)} w_{j,i+n_d} \frac{\partial^2 yn(n+k-i)}{\partial u(n+m)\,\partial u(n+h)} \tag{3.39}$$

$$\frac{\partial yn(n+k)}{\partial u(n+h)} = \sum_{j=0}^{hid-1} w_j \frac{\partial f\left(net_j(n+k)\right)}{\partial u(n+h)} \tag{3.40}$$

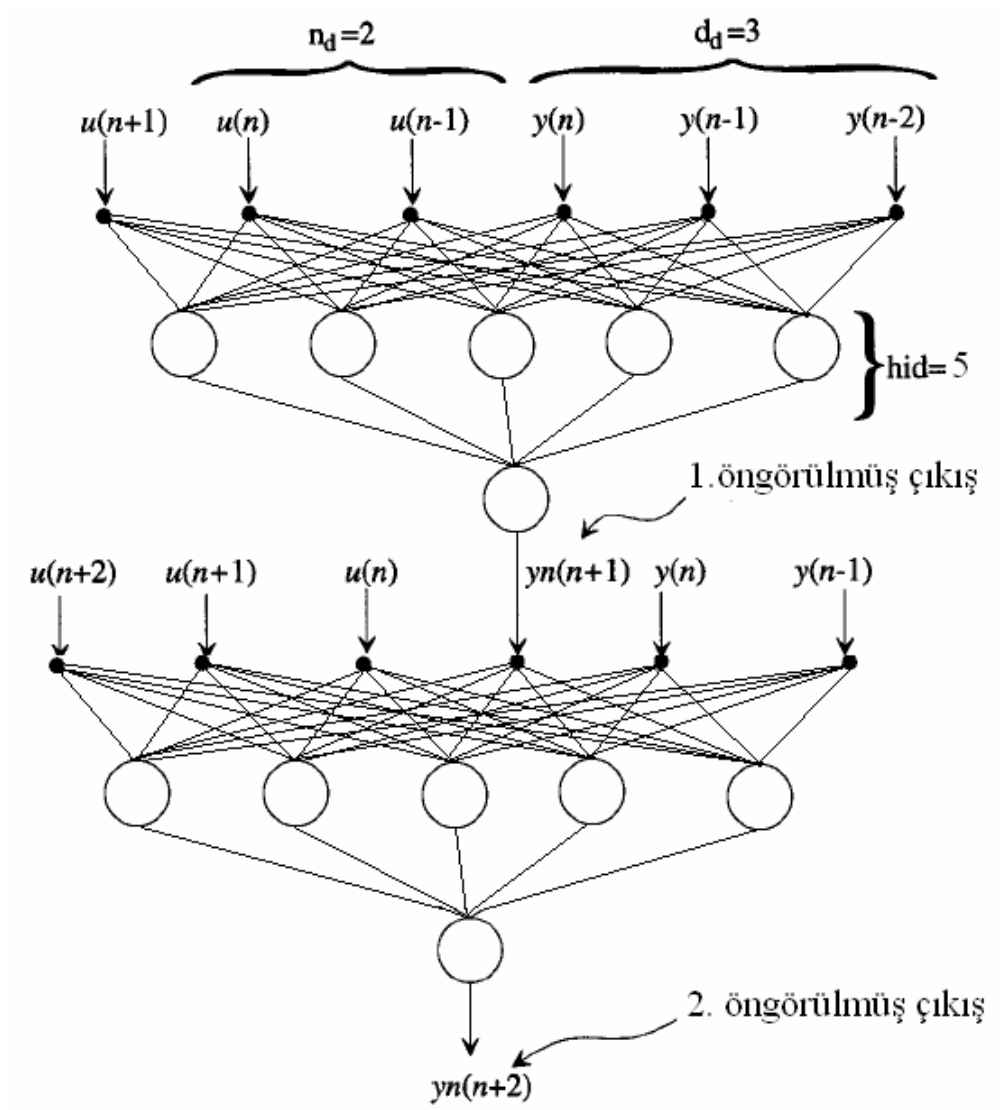
$$\frac{\partial^2 yn(n+k)}{\partial u(n+m)\,\partial u(n+h)} = \sum_{j=0}^{hid-1} w_j \frac{\partial^2 f\left(net_j(n+k)\right)}{\partial u(n+m)\,\partial u(n+h)} \tag{3.41}$$

$$\frac{\partial f\left(net_j(n+k)\right)}{\partial u(n+h)} = \frac{\partial f\left(net_j(n+k)\right)}{\partial net_j(n+k)} \frac{\partial net_j(n+k)}{\partial u(n+h)} \tag{3.42}$$

$$\frac{\partial^2 f\left(net_j(n+k)\right)}{\partial u(n+m)\,\partial u(n+h)} = \frac{\partial f\left(net_j(n+k)\right)}{\partial net_j(n+k)} \frac{\partial^2 net_j(n+k)}{\partial u(n+m)\,\partial u(n+h)} \tag{3.43}$$

$$+ \frac{\partial^2 f\left(net_j(n+k)\right)}{\partial net_j(n+k)^2} \frac{\partial net_j(n+k)}{\partial u(n+m)} \frac{\partial net_j(n+k)}{\partial u(n+h)}$$

İki adım sonrasını öngören, altı girişe ve bir çıkışa sahip örnek bir öngörme işlemini aşağıdaki gibi organize edebiliriz.




Şekil 3.9. YSA ile örnek bir öngörme işlemi

$y_n(n+2)$ çıkışını elde etmek için $u(n+1)$ ve $u(n+2)$ girişlerine ihtiyaç vardır. Öngörme işlemi $n$ anında $[u(n)\ u(n-1)]$ ve $[y(n)\ y(n-1)\ y(n-2)]$ başlangıç şartları ve tahmin edilmiş $u(n+1)$ girişi ile başlatılır. Bu işlemin çıkışı $y_n(n+1)$'dır ve $y_n(n+2)$ çıkışını elde etmek için yeniden YSA ya geri iletilir.

### 3.4. Yinelenen Elman Ağ Uyarlamalı Nöro Genelleştirilmiş Öngörülü Kontrol

Doğrusal olmayan sistemlerin öngörülü kontrolü için doğrusal olmayan öngörü modeli kullanmanın kontrolün başarımını arttırdığı literatürdeki birçok çalışmada vurgulanmıştır [40, 57, 63]. Son yirmi yıldır yapay sinir ağları doğrusal olmayan yapıları ile bu alandaki çözümün odak noktası olmuştur. Yapay sinir ağ uyarlamalı genelleştirilmiş öngörülü kontrol (NGPC) algoritması, öngörü modelinin kabiliyetini geliştirmiş, doğru tahmin edebilirliğini arttırmış ve dolayısıyla istenilen sistem çıkışları ile gerçeklenen çıkışlar arasındaki hataları azaltmıştır [63]. Ayrıca kontrol süresince oluşabilecek istenmeyen durumlar karşısında daha dengeleyici davranışlar sergilemiş ve kontrol algoritmalarının adaptif özelliklerini arttırmıştır [40].

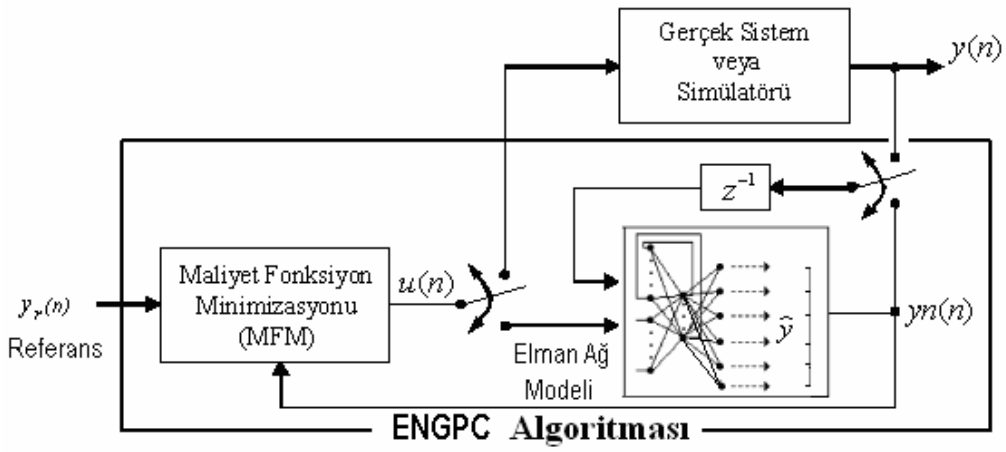
Öte yandan öngörülü kontrol algoritmasında kullanılan yapay sinir ağ modelinin yapısı algoritmanın performansını doğrudan etkileyen bir unsurdur. Ağın öğrenme süresi ve öngörü işlemlerinin başarısı bu yapının seçimine bağlıdır. Uzun bir öğrenme süresi ve hatalı öngörü işlemi gerçek-zaman uygulamaları için bir dezavantajdır [58]. Yapay sinir ağ uyarlamalı GPC (NGPC) algoritmalarında şimdiye kadar kullanılan ağ modelleri ileri beslemeli ağ gibi statik ağ modelleridir. Bu ağlar, öngörülü kontrolün başarımını arttırmalarına karşın ağırlıkların güncellenmesi ve öğrenme süreleri nedeniyle gerçek-zaman uygulamaları için problem teşkil etmektedirler.

Bu tezde, Bölüm 3.3'de anlatılan Nöro Genelleştirilmiş Öngörülü Kontrol (NGPC) algoritmasının performansını geliştirmek için Yinelenen Elman Ağı (Recurrent Elman's Network-REN) önerilmiştir.

Elman ağı ilk olarak 1990 yılında zaman serileri için önerilmiştir [59]. Kremer elman ağını diğer ağlar ile karşılaştırmış ve hesaplama gücünü ortaya koymuştur [60]. Sonraki çalışmalarda bu ağ yapısı zaman serilerinin tahmininde kullanılmıştır [61, 62]. Ayrıca dinamik sistemler için Elman ağ tabanlı kontrol algoritmaları tasarlanmış ve ağın öğrenme süresini kısaltarak algoritmaların performansını geliştirdiği vurgulanmıştır [58].

Elman ağı diğer statik ağların aksine sahip olduğu gizli bağlam katmanı (context hidden layer) ile zaman geciktirilmiş (delaying ) giriş ve çıkışları olmaksızın geçmiş bilgileri hatırlayabilmektedir. Dolayısıyla katmanlar arasındaki ağırlık bileşenlerinin ayarlanması diğer ağlara nispeten daha hızlıdır. Bu durum öğrenme kabiliyetini ve ağın performansını doğrudan arttırır. Öğrenme ve öngörme işlemlerinde harcanan süre dinamik sistemlerin gerçek-zaman yapay sinir ağ tabanlı kontrolünde problemlere yol açacağından, daha hızlı ağ modellerine ihtiyaç duyulur. Dolayısıyla elman ağının öğrenme hızı işlem zamanını kısaltacağından, gerçek-zaman uygulamaları için daha etkindir.

Bu tezde, NGPC algoritmasının performansını geliştirmek için Bölüm 3.3'de anlatılan ileri beslemeli ağa sahip NGPC yerine Yinelenen Elman Ağ uyarlamalı NGPC (Elman Recurrent Network implemented-ENGPC) algoritması önerilmiştir. Önerilen bu kontrol algoritması istenilen referans yörüngeye yakınsama bakımından ileri besleme ağına sahip NGPC algoritmasına kıyasla daha hızlıdır. ENGPC algoritmasının blok diyagramı Şekil 3.10'da verilmiştir.




Şekil 3.10. Yinelenen elman ağ uyarlamalı NGPC (ENGPC)

Görüldüğü gibi ENGPC algoritması NGPC algoritması ile yapısal olarak aynıdır. Algoritma öngörü ve öğrenme işlemleri için kullanılan bir elman ağı ve maliyet fonksiyonunu minimize eden bir işlem bloğundan oluşmaktadır.

### 3.4.1. Maliyet fonksiyonu

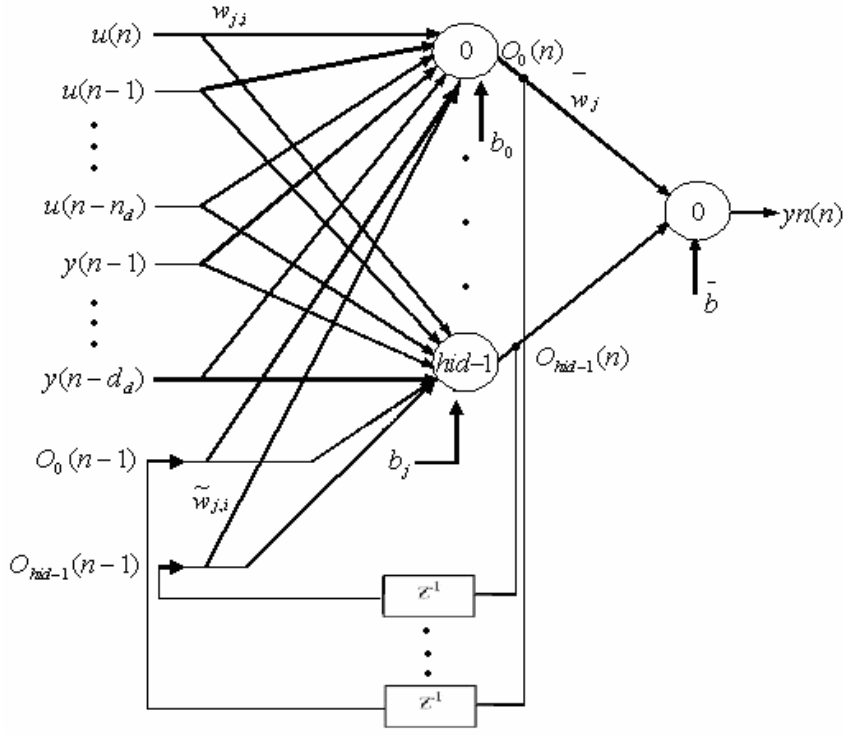
ENGPC algoritmasında kullanılan MFM bloğu, her kontrol adımında maliyet fonksiyonunu tekrardan minimize ederek sistem için uygun kontrol girdilerini üretir. Bu kontrol girdileri kontrolün başından sonuna kadar sistem çıkışını istenen çıkış yörüngesine yakınsayacak şekilde devamlı değişir.

Maliyet fonksiyonu minimizasyonunda güncelleme algoritması olarak hem iterasyon sayısının azlığı nedeniyle hem de ileri besleme ağ kullanan NGPC ile kıyaslama yapmak için Newton-Raphson metodu tercih edilmiştir.

ENGPC algoritmasında kullanılan maliyet fonksiyonu ve güncelleme metoduna ait denklemler NGPC algoritmasında kullanılanlarla aynıdır ve bir önceki bölümde (Bölüm 3.3) ayrıntılı bir şekilde anlatılmıştır.

### 3.4.2. Kullanılan elman yapay sinir ağ modeli

ENGPC algoritmasında öngörülerin yapılması için yinelenen elman yapay sinir ağı kullanılmıştır. Bu ağ, sisteme uygulanan geçmiş zaman dilimlerindeki girdileri ($n_d + 1$ tane) ve sistemden alınan geçmiş zaman dilimlerindeki çıktıları ($d_d$ tane) kullanarak sistemin o anki ürettiği çıktıya yakınsama yapar. $n_d$ ve $d_d$ değerlerinin büyük seçilmesinin fazla bir faydası yoktur. Bununla beraber, işlem yükünü artıracağından gerçek zamanlı kontrolü zorlaştıracaktır. Ağın yapısı Şekil 3.11'de gösterilmiştir.




Şekil 3.11. ENGPC algoritmasında kullanılan yapay sinir ağ modeli

Bu tez çalışmasında ENGPC algoritması için üç katmanlı bir yapay sinir ağı kullanılmıştır. Giriş katmanında altı giriş, çıkışta ise bir çıkış mevcuttur. Ara katmanda beş adet nöron kullanılmıştır.

Sistem modelinde kullanılan ağ denklemleri (3.44), (3.45) ve (3.46)'da verilmiştir. Her biri bir öncekinin sonucunu kullanır. Denklemlerden anlaşıldığı gibi ara katmanda tan-sigmoid transfer fonksiyonu, çıkış katmanında ise lineer transfer fonksiyonu kullanılmaktadır. Çıkış katmanında lineer transfer fonksiyonu kullanılmasının sebebi işlemleri azaltmaktır. Ayrıca, işlem yükünü azaltmak için ara katmandaki nöron sayısı öngörüye zarar vermeyecek şekilde küçük tutulabilir.

$$net_j(n) = b_j + \sum_{i=0}^{n_d} w_{j,i}\, u(n-i) + \sum_{i=1}^{d_d} w_{j,i+n_d}\, y(n-i) + \sum_{i=0}^{hid-1} \tilde{w}_{j,i}\, O_i(n-1) \tag{3.44}$$

$$O_j(n) = f\left(net_j(n)\right) = \frac{e^{net_j(n)} - e^{-net_j(n)}}{e^{net_j(n)} + e^{-net_j(n)}} = 1 - \frac{2}{1 + e^{2 \cdot net_j(n)}} \tag{3.45}$$

$$yn(n) = b + \sum_{j=0}^{hid-1} \bar{w}_j O_j(n) \tag{3.46}$$

Hata fonksiyon denklemi (3.47)'de verilmiş olup hatayı minimize edecek şekilde ağırlıkları güncellenmede kullanılacak denklemlerin çıkarılması için gereklidir.

$$E_p = \frac{1}{2}\left( y(n) - yn(n) \right)^2 \tag{3.47}$$

Ağırlık güncellemede işlem kolaylığı olsun diye ağ girdileri

$$x(n) = \left( u(n), u(n-1), \cdots, u(n-n_d), y(n-1), \cdots, y(n-d_d) \right)$$

şeklinde satır matrisi olarak alınır. Her kontrol adımında ağ tekrardan eğitilir ve eğitim hata istenen seviyeye düşene kadar tekrarlanır. Eğitim tek data için olduğundan çok hızlıdır. Yani eğitim için geçen zaman ihmal edilebilir. Ağırlık değişimleri her eğitim döngüsü için (3.48), (3.49), (3.50), (3.51) ve (3.52)'deki denklemler kullanılarak hesaplanır.

$$\Delta \bar{w}_j(t+1) = -\eta \frac{\partial E}{\partial w_j} = \eta \left( y(n) - yn(n) \right) O_j(n) \tag{3.48}$$

$$\Delta \bar{b}(t+1) = -\eta \frac{\partial E}{\partial b} = \eta \left( y(n) - yn(n) \right) \tag{3.49}$$

$$\Delta w_{j,i}(t+1) = -\eta \frac{\partial E}{\partial w_{j,i}} = \eta \left( y(n) - yn(n) \right) \left( 1 - O_j(n)^2 \right) \bar{w}_j \; x_i(n) \tag{3.50}$$

$$\Delta b_j(t+1) = -\eta \frac{\partial E}{\partial b_j} = \eta \left( y(n) - yn(n) \right) \left( 1 - O_j(n)^2 \right) \bar{w}_j \tag{3.51}$$

$$\Delta \tilde{w}_{j,i}(t+1) = -\eta \frac{\partial E}{\partial \tilde{w}_{j,i}} = \eta \left( y(n) - yn(n) \right) \left( 1 - O_j(n)^2 \right) \bar{w}_j \; O_i(n-1) \tag{3.52}$$

Elde edilen ağırlık değişim değerleri (3.53), (3.54), (3.55), (3.56) ve (3.57)'deki denklemlerde yerlerine konularak ağırlıklar güncellenir.

$$\bar{w}_j(t+1) = \bar{w}_j(t) + \Delta \bar{w}_j(t+1) \tag{3.53}$$

$$\bar{b}(t+1) = \bar{b}(t) + \Delta \bar{b}(t+1) \tag{3.54}$$

$$w_{j,i}(t+1) = w_{j,i}(t) + \Delta w_{j,i}(t+1) \tag{3.55}$$

$$b_j(t+1) = b_j(t) + \Delta b_j(t+1) \tag{3.56}$$

$$\tilde{w}_{j,i}(t+1) = \bar{w}_{j,i} + \Delta \tilde{w}_{j,i}(t+1) \tag{3.57}$$

### 3.4.3. Yapay sinir ağ modelinde öngörme

NGPC algoritması sistemin davranışını taklit edebilmek ve bu taklidi kullanarak öngörüler yapabilmek için yukarıda da bahsedildiği gibi nöral sistem modeli kullanır. Bu nöral sistem modeli sisteme uygulanan geçmiş kontrol adımlarındaki girdileri ve bu girdilere karşılık sistemin verdiği çıktıları kullanarak önce sistemi kendi içinde modeller. Sonra bu modeli kullanarak gelecek zaman dilimleri için öngörüler yapar. Bütün bu işlemler her kontrol adımda tekrarlanır. Öngörü denklemeleri aşağıda verilmiştir.

$$net_j(n+k) = b_j + \sum_{i=0}^{n_d} w_{j,i} \begin{cases} u(n+k-i) & i > k - N_u \\ u(n+N_u) & i \le k - N_u \end{cases}$$

$$+ \sum_{i=1}^{\min(k-1,d_d)} w_{j,i+n_d}\, yn(n+k-i) + \sum_{i=k}^{d_d} w_{j,i+n_d}\, y(n+k-i) + \sum_{i=k}^{hid-1} \tilde{w}_{j,i}\, O_i(n+k-1) \tag{3.58}$$

$$O_j(n+k) = f\left(net_j(n+k)\right) = 1 - \frac{2}{1+e^{2\cdot net_j(n+k)}} \tag{3.59}$$

$$yn(n+k) = \bar{b} + \sum_{j=0}^{hid-1} \bar{w}_j\, O_j(n+k) \tag{3.60}$$

Bu üç denklem kullanılarak aşağıdaki türev denklemleri ( Denklem 3.61'den Denklem 3.68'e kadar ) elde edilir.

$$\frac{\partial f\left(net_j(n+k)\right)}{\partial net_j(n+k)} = 1 - O_j(n+k)^2 \tag{3.61}$$

$$\frac{\partial^2 f\left(net_j(n+k)\right)}{\partial net_j(n+k)^2} = -2\,O_j(n+k)\left(1 - O_j(n+k)^2\right) \tag{3.62}$$

$$\frac{\partial net_j(n+k)}{\partial u(n+h)} = \sum_{i=0}^{n_d} w_{j,i} \begin{cases} \dfrac{\partial u(n+k-i)}{\partial u(n+h)} & i > k - N_u \\ \\ \dfrac{\partial u(n+N_u)}{\partial u(n+h)} & i \le k - N_u \end{cases} + \sum_{i=1}^{\min(k-1,d_d)} w_{j,i+n_d} \frac{\partial yn(n+k-i)}{\partial u(n+h)} \tag{3.63}$$

$$\frac{\partial^2 net_j(n+k)}{\partial u(n+m)\,\partial u(n+h)} = \sum_{i=1}^{\min(k-1,d_d)} w_{j,i+n_d} \frac{\partial^2 yn(n+k-i)}{\partial u(n+m)\,\partial u(n+h)} \tag{3.64}$$

$$\frac{\partial yn(n+k)}{\partial u(n+h)} = \sum_{j=0}^{hid-1} w_j \frac{\partial f\left(net_j(n+k)\right)}{\partial u(n+h)} \tag{3.65}$$

$$\frac{\partial^2 yn(n+k)}{\partial u(n+m)\,\partial u(n+h)} = \sum_{j=0}^{hid-1} w_j \frac{\partial^2 f\left(net_j(n+k)\right)}{\partial u(n+m)\,\partial u(n+h)} \tag{3.66}$$

$$\frac{\partial f\left(net_j(n+k)\right)}{\partial u(n+h)} = \frac{\partial f\left(net_j(n+k)\right)}{\partial net_j(n+k)} \frac{\partial net_j(n+k)}{\partial u(n+h)} \tag{3.67}$$

$$\begin{aligned} \frac{\partial^2 f\left(net_j(n+k)\right)}{\partial u(n+m)\,\partial u(n+h)} &= \frac{\partial f\left(net_j(n+k)\right)}{\partial net_j(n+k)} \frac{\partial^2 net_j(n+k)}{\partial u(n+m)\,\partial u(n+h)} \\ &+ \frac{\partial^2 f\left(net_j(n+k)\right)}{\partial net_j(n+k)^2} \frac{\partial net_j(n+k)}{\partial u(n+m)} \frac{\partial net_j(n+k)}{\partial u(n+h)} \end{aligned} \tag{3.68}$$

Elde edilen bu denklemler Newton-Raphson güncelleme denkleminde yerlerine konulursa uygun kontrol giriş vektörü kolaylıkla hesaplanır. Bu hesaplama işlemi bir kaç kere tekrarlanırsa kontrol giriş vektörü daha hassas olur. Kontrol giriş vektöründeki ilk eleman kontrol girişi olarak sisteme uygulanır.

# BÖLÜM 4. SONUÇLAR

Bu bölümde, ilk olarak tasarlanan altı eklemli robot manipülatörü tanıtılacak, ardından gerçeklenen kontrol algoritmaları ile elde edilen simülasyon sonuçları aktarılacaktır.

## 4.1. Robot Kolu Simülasyonu

Bu bölümde, GPC, SGA-GPC, NGPC ve ENGPC algoritmalarının her birinin SISO ve MIMO olmak üzere altı eklemli bir robot koluna eklem esaslı yörünge kontrolü için uygulanmalarıyla ilgili yapılan çalışmalar anlatılacaktır. Ayrıca gerçeklenen paket program, kullanılan robot model, algoritmaların bu robot modele uygulanış blok ve akış diyagramları ve kontrol deneyleri anlatılacaktır.

GPC algoritmalarında parametre kestirimi için ardışık en küçük kareler yöntemi, NGPC ve ENGPC algoritmalarında güncelleme için ise Newton-Raphson yöntemi tercih edilmiştir. SGA-GPC algoritmasında ise maliyet fonksiyonu minimizasyonu için genetik algoritma kullanılmıştır.

Robot kolunun her bir eklemine ait takip etmesi istenen konum referans ve hız referans yörüngeleri kübik ve sinüzoidal yörünge esaslarına göre belirlenmiştir.

Robot kolunun hareketlerini modellemede Lagrange-Euler yöntemi kullanılmıştır. Bu yönteme göre $n$ eklemli bir robot kolu için $n$x1 boyutlu eklem tork vektörü $\tau(t)$ ile $n$x1 boyutlu eklem açı vektörü $\theta(t)$ arasındaki ilişki aşağıdaki gibi doğrusal olmayan bir diferansiyel denklem takımı ile gösterilir.

$$\tau(t) = D(\theta)\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta) \tag{4.1}$$

Tez çalışmaları sırasında, Lagrange-Euler denklem takımına eklemlerdeki sürtünmeler, eklemlerdeki motorlara ait ataletlerin dinamik modele etkisi, robot kolunun uç elemanında bir yük taşınması ve taşınan yükün taşıma esnasında düşmesi durumları da ayrıca ilave edilerek simülasyon çalışmaları yapılmıştır. Bu durumda (4.1)'deki genel denklem aşağıdaki şekilde yeniden ifade edilir.

$$\tau(t) = D(\theta)\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta) + \tau_s(t) + \tau_y(t) \qquad (4.2)$$

Dinamik modelleme sonucunda birbirleriyle yüksek oranda etkileşimli altı adet ikinci dereceden doğrusal olmayan diferansiyel denklem takımı elde edilir. Bu altı diferansiyel denklem altı eklemli robot kolunu sürtünme ve taşınan yük etkilerini de içerecek şekilde dinamik olarak formüle eder. Bu denklemler Borland Delphi 7.0 programlama dili kullanılarak bilgisayar ortamında robot kolu simülatörüne dönüştürülmüştür. Robot kolu simülatörünün yazılımında diferansiyel denklem çözümleri için en uygun yöntem olarak gözüken dördüncü mertebeden Runge-Kutta bütünleştirme yöntemi tercih edilmiştir.

### 4.1.1. Kontrol algoritmalarının altı eklemli robot koluna uygulanışı

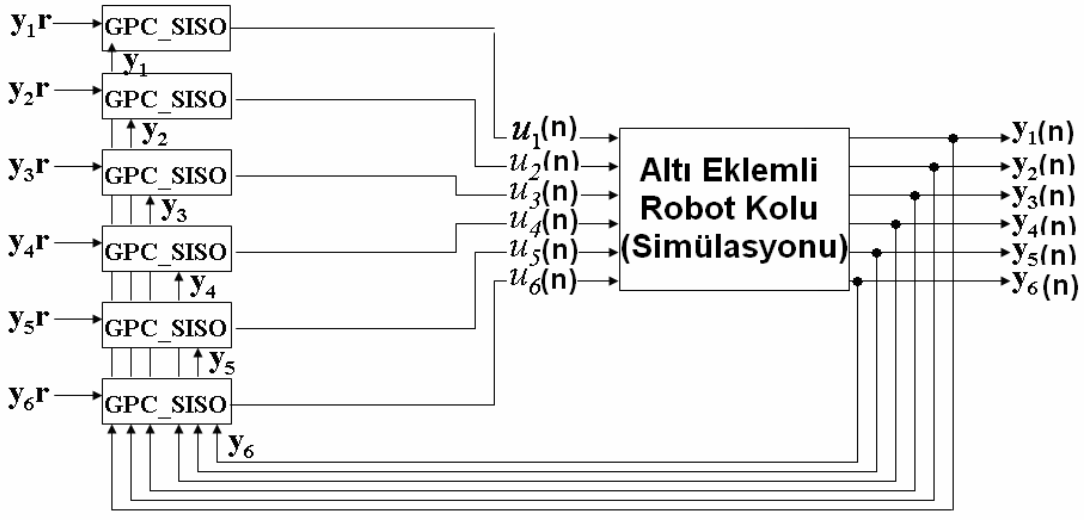
Kontrol edilecek sistemler genellikle çok değişkenlidir. Bu değişkenler giriş ve çıkış olmak üzere iki kısımdan oluşmaktadırlar. Her bir giriş ve çıkış değişkeni kendileri de dahil olmak üzere diğer giriş ve çıkış değişkenleri ile değişik düzeylerde de olsa etkileşimlidir. Bu yüzden, robot kolu kontrolü için kullanılabilecek kontrol algoritmalarının seçimi oldukça önemlidir.

Bu tez çalışmasında kullanılan altı eklemli robot kolu, altı girişli altı çıkışlı bir sistemdir. Girişler eklemlere uygulanan torklar / voltajlar, çıkışlar ise eklemlerin açısal hızları olarak düşünülebilir. Robot kolunun dinamik davranışı, altı adet ikinci dereceden doğrusal olmayan diferansiyel denklem ile karakterize edilmektedir. Bu altı denklemin her biri hem kendi içinde hem de diğerleriyle yüksek oranda etkileşimlidir. Robot kol tek bir sistem olmasına rağmen her bir eklem ayrı bir diferansiyel denkleme sahip olması nedeniyle ayrı bir sistem gibi düşünülebilir. Diğer bir deyişle her bir eklemin bağımsız olarak bir giriş bir çıkışa sahip olması,

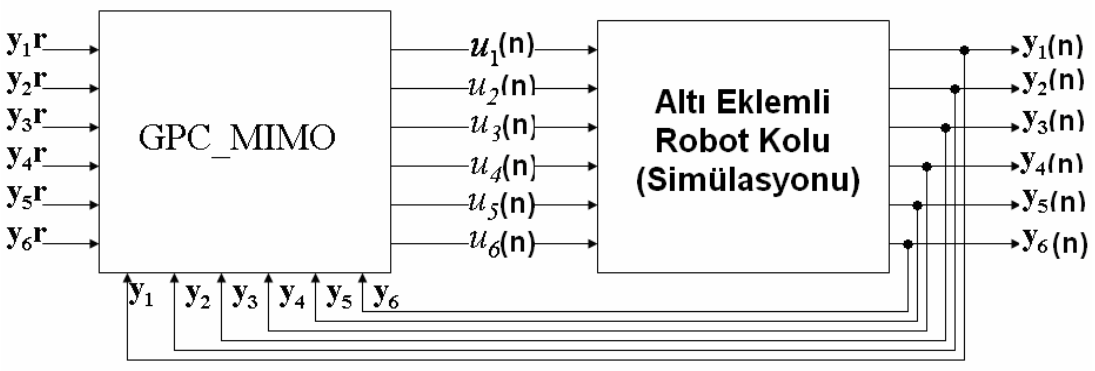
diğer eklemlerle etkileşimli olmasına rağmen, her bir eklemin ayrı bir SISO algoritması ile kontrol edilebilmesine olanak sağlar. Bu durumda, yapı itibarıyla aynı fakat birbirinden tamamen bağımsız çalışan altı tane SISO algoritması kullanılır. Eğer kol tek bir sistem olarak düşünülürse bütün eklemleri kapsayacak ortak bir MIMO algoritması yeterlidir.

Eklemler birbirleriyle dinamik olarak etkileşimli olduklarından MIMO mantığına göre tasarım daha kapsamlı bir yaklaşım gibi görülebilir. Fakat, bu etkileşimler eklem hızlarına yansıdığından algoritmaların SISO mantığına göre tasarlanmasında bir sakınca yoktur. SISO mantığına göre tasarlanan algoritmalarda hesapsal yük MIMO algoritmalara kıyasla çok daha azdır. Bu yüzden daha hızlıdırlar. Ayrıca, her bir eklem ayrı bir algoritma ile kontrol edildiğinden paralel programlama için uygundurlar. Bunlara ilaveten, yapılan simülasyon çalışmaları SISO tasarımın robot kolu kontrolünde daha uygun ve kaliteli sonuçlar verdiğini göstermiştir.

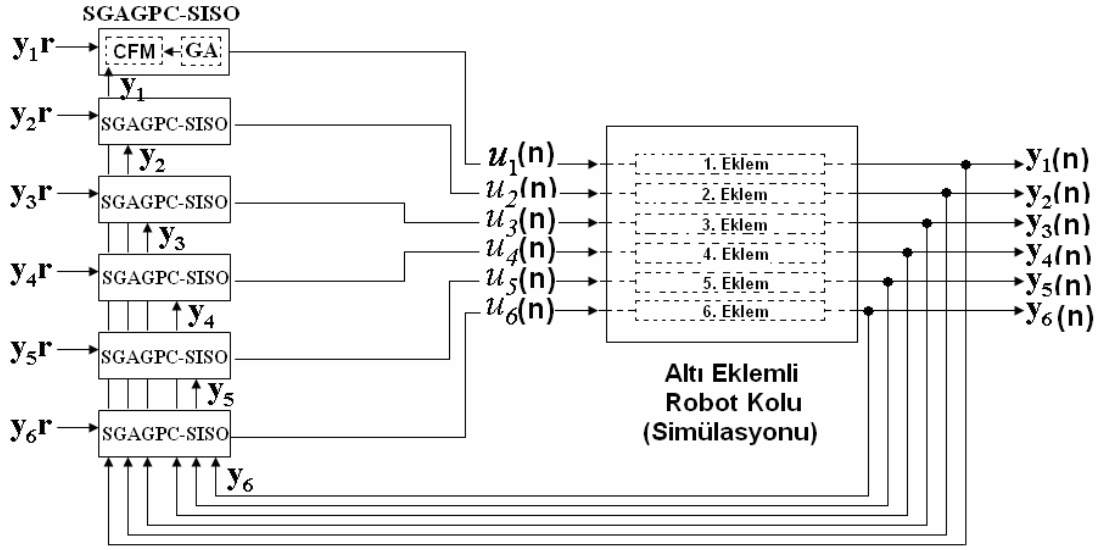
GPC, SGA-GPC, NGPC ve ENGPC algoritmalarının altı eklemli robot koluna SISO ve MIMO olarak uygulanış blok diyagramları aşağıda Şekil 4.1'den Şekil 4.8'e kadar verilmiştir.



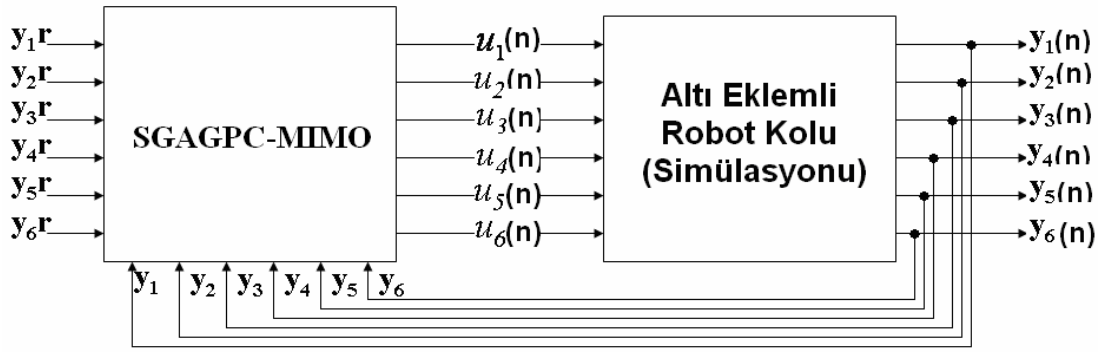

Şekil 4.1. GPC algoritmasının altı eklemli bir robot koluna SISO olarak uygulanış blok diyagramı



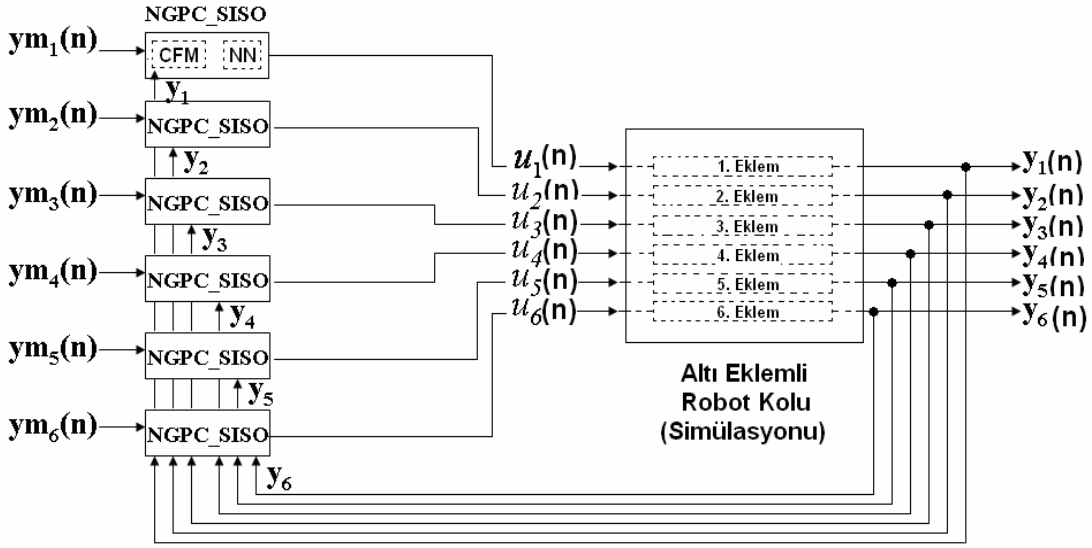

Şekil 4.2. GPC algoritmasının altı eklemli bir robot koluna MIMO olarak uygulanış blok diyagramı



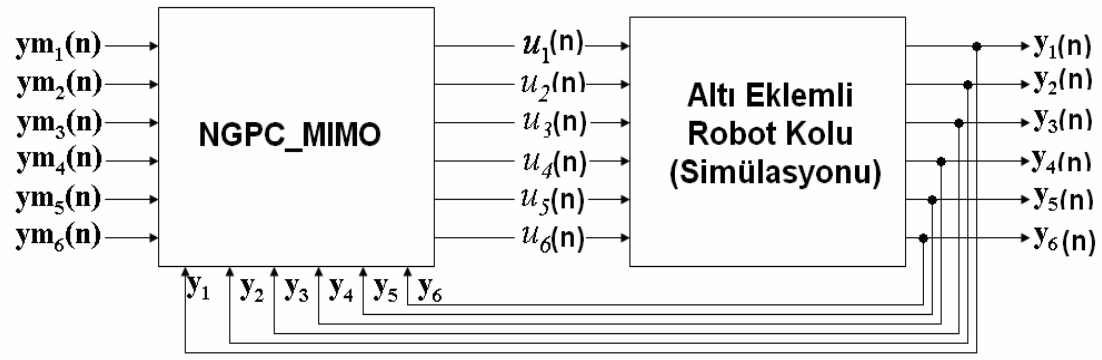

Şekil 4.3. SGA-GPC algoritmasının altı eklemli bir robot koluna SISO olarak uygulanış blok diyagramı



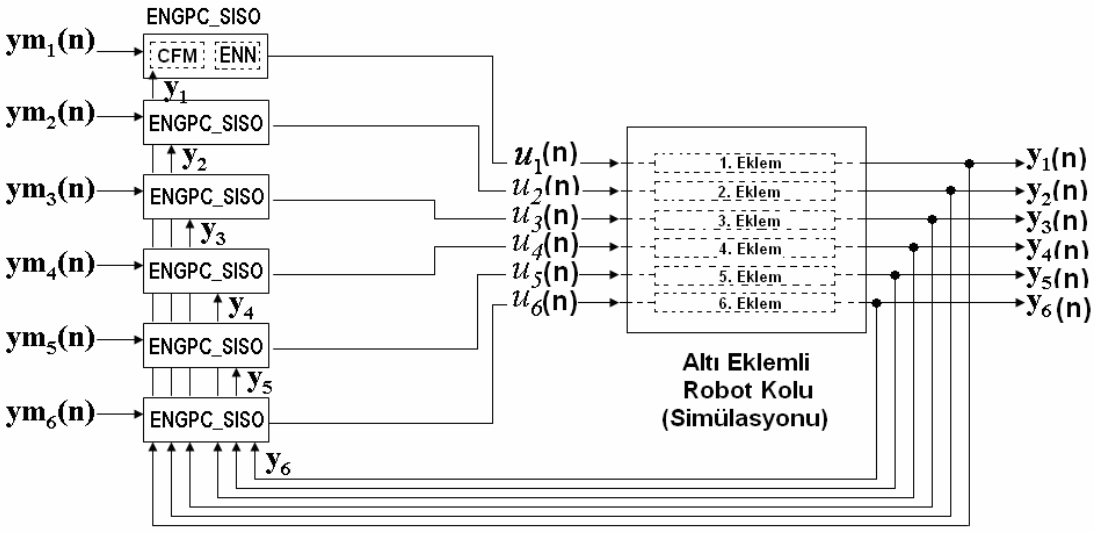

Şekil 4.4. SGA-GPC algoritmasının altı eklemli bir robot koluna MIMO olarak uygulanış blok diyagramı



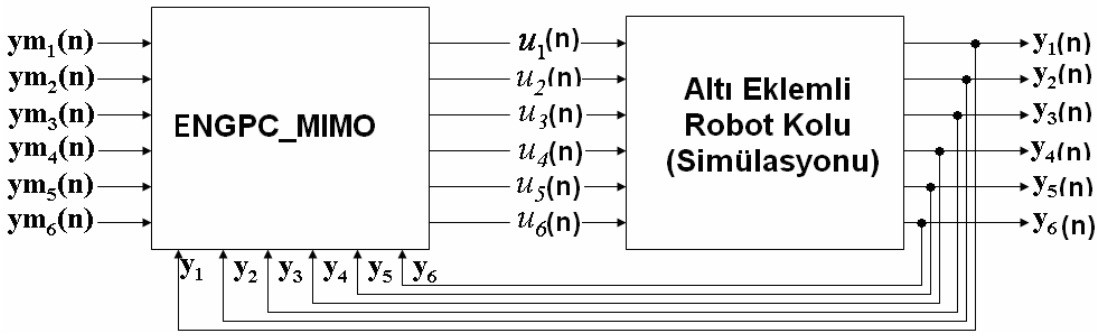

Şekil 4.5. NGPC algoritmasının altı eklemli bir robot koluna SISO olarak uygulanış blok diyagramı




Şekil 4.6. NGPC algoritmasının altı eklemli bir robot koluna MIMO olarak uygulanış blok diyagramı




Şekil 4.7. ENGPC algoritmasının altı eklemli bir robot koluna SISO olarak uygulanış blok diyagramı




Şekil 4.8. ENGPC algoritmasının altı eklemli bir robot koluna MIMO olarak uygulanış blok diyagramı

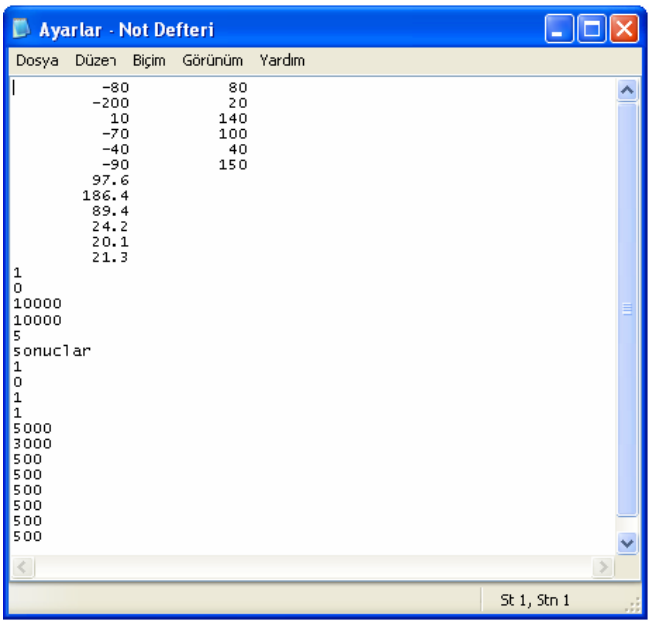
### 4.1.2. Gerçeklenen programın tanıtımı

Bu tez çalışmasında, altı eklemli bir robot kolu ve değişik şartlar altında kontrolü ile ilgili bir simülasyon programı tek bir paket program halinde Borland Delphi 7.0 programlama dili kullanılarak gerçekleştirilmiş ve kullanılmıştır. Paket program 6 ana kısımdan oluşmaktadır. Bu kısımlar; kontrol algoritmaları, robot kolu simülatörü, yörünge planlaması, dosyalama, grafiksel gösterim ve grafik ara yüzdür.

Kontrol algoritmaları GPC SISO, GPC MIMO, SGA-GPC SISO, SGA-GPC MIMO, NGPC SISO, NGPC MIMO, ENGPC SISO ve ENGPC MIMO olmak üzere toplam 8 tanedir. Bu kontrol algoritmalarının görevi, simülasyon boyunca robot kolu eklemlerini istenen yörüngeler doğrultusunda en az hata ile hareket ettirmektir. Eklem yörüngelerinin çıkartılması işini yörünge planlaması kısmı yapar. Dosyalama kısmı simülasyon sonuçlarının dosyalara kaydedilmesini, grafiksel gösterim kısmı ise bu sonuçların grafiksel olarak gösterilmesini sağlar. Bu sayede simülasyon sonuçlarının yorumlanması daha kolay yapılabilir. Simülasyon sonunda kullanıcı hangi grafiği görmek istiyorsa kolayca seçebilmekte, aynı zamanda istediği dosya ismiyle grafik formatında kaydedebilmektedir. Grafik ara yüz, yazılımın kullanımını kolaylaştırmak ve sonuçları sadece dosyalar halinde değil aynı zamanda grafiksel olarak görebilmek için tercih edilmiştir.

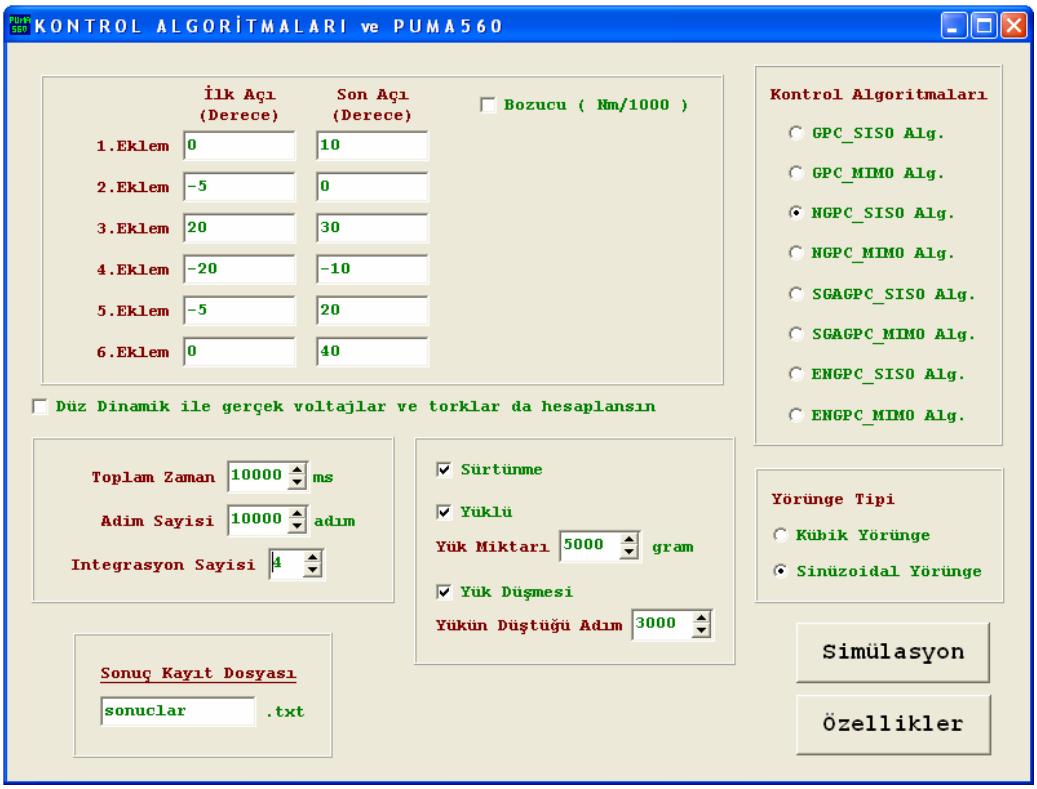
Geliştirilen paket program oldukça kullanışlı ve kullanımı gayet kolaydır. Her bir eklem için başlangıç ve bitiş pozisyonu rahatlıkla girilebilmektedir. İstenilen kontrol algoritması ve yörünge planlaması kolayca seçilebilmektedir. Minimum ve maksimum tork aralığı, toplam simülasyon zamanı, toplam kontrol adım sayısı ve her

bir kontrol adımı başına düşen Runge-Kutta adım sayısı kullanıcı tarafından kolayca belirlenebilmektedir. Sürtünme etkisi, bozucu ilavesi, yük taşıma ve taşınan yükün düşmesi gibi durumlar seçilip simülasyona dahil edilebilmektedir. Bozucu aralığı, yük miktarı ve yük düşme zamanı kullanıcı tarafından istenildiği gibi belirlenebilmektedir. Bütün bu değişiklikler başlangıç ayarları dosyasına kaydedilmekte (Şekil 4.9) ve program her yeniden çalıştırıldığında en son ayarlar korunmaktadır. Bu durum kullanıcıya kolaylık sağlamaktadır.

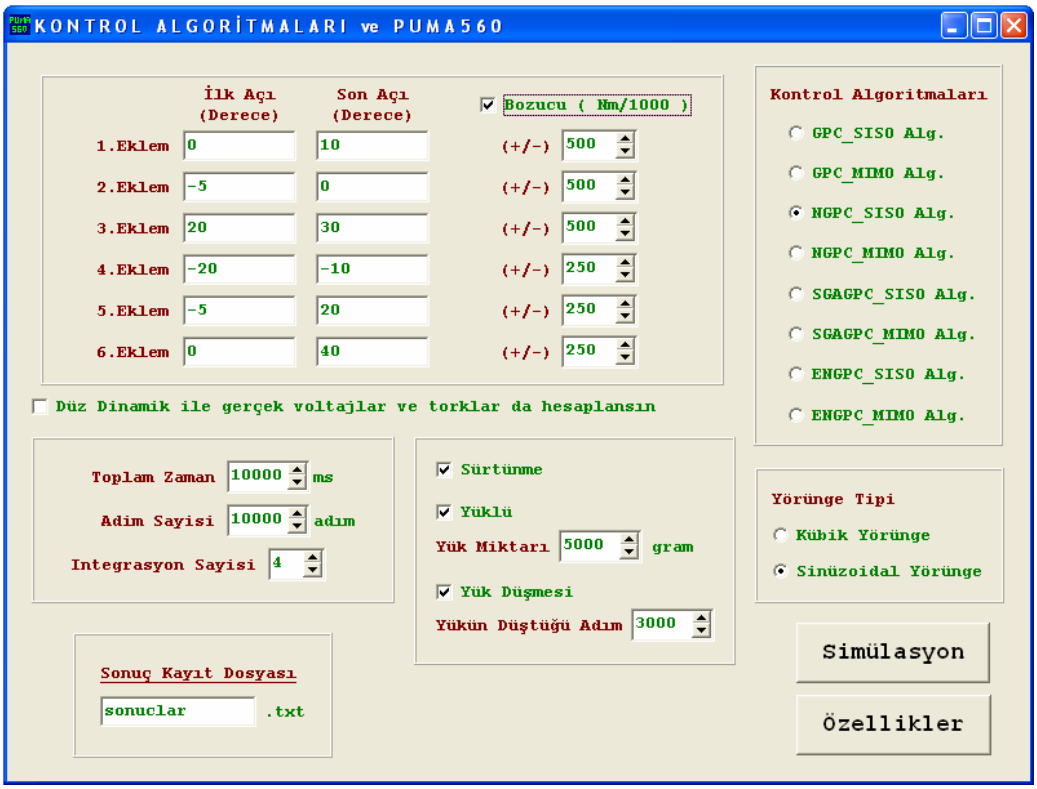



Şekil 4.9. Paket programın başlangıç ayarları dosyasının görünümü

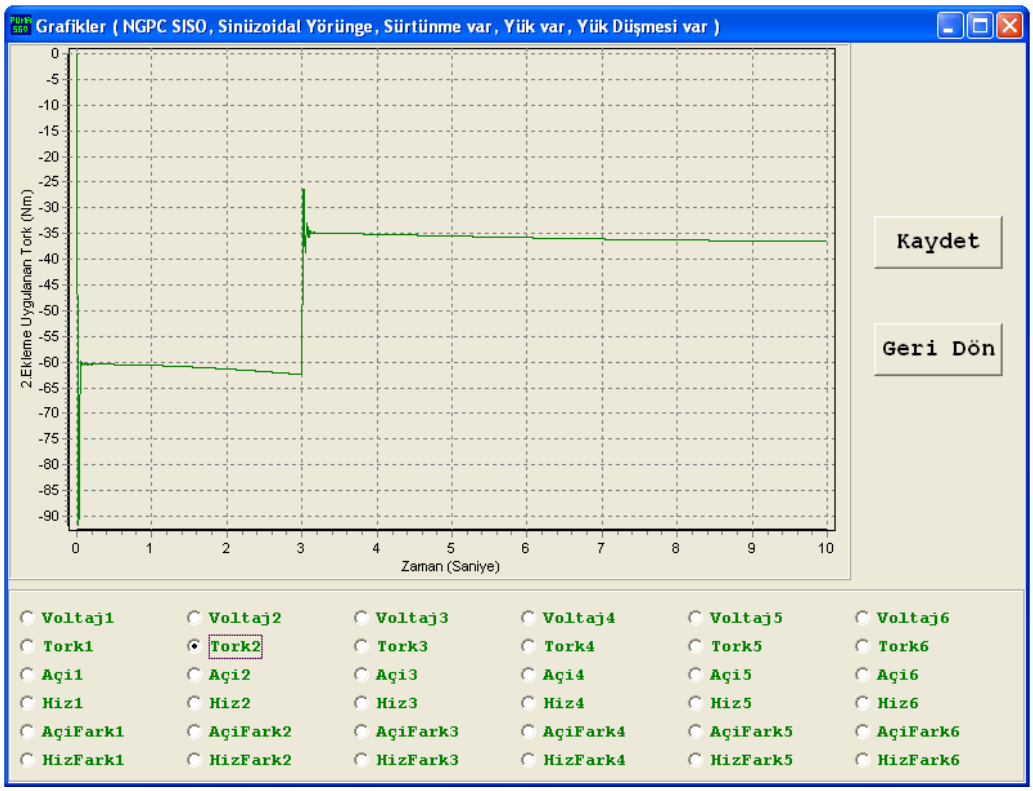
Paket programın grafik ara yüzünden bazı görünümleri Şekil 4.10'dan Şekil 4.14'e kadar, elde edilen sonuçların dosyasal görünümleri Şekil 4.15'de, paket programın ve algoritmaların akış diyagramları Şekil 4.16'dan Şekil 4.20'ye kadar aşağıda verilmiştir.



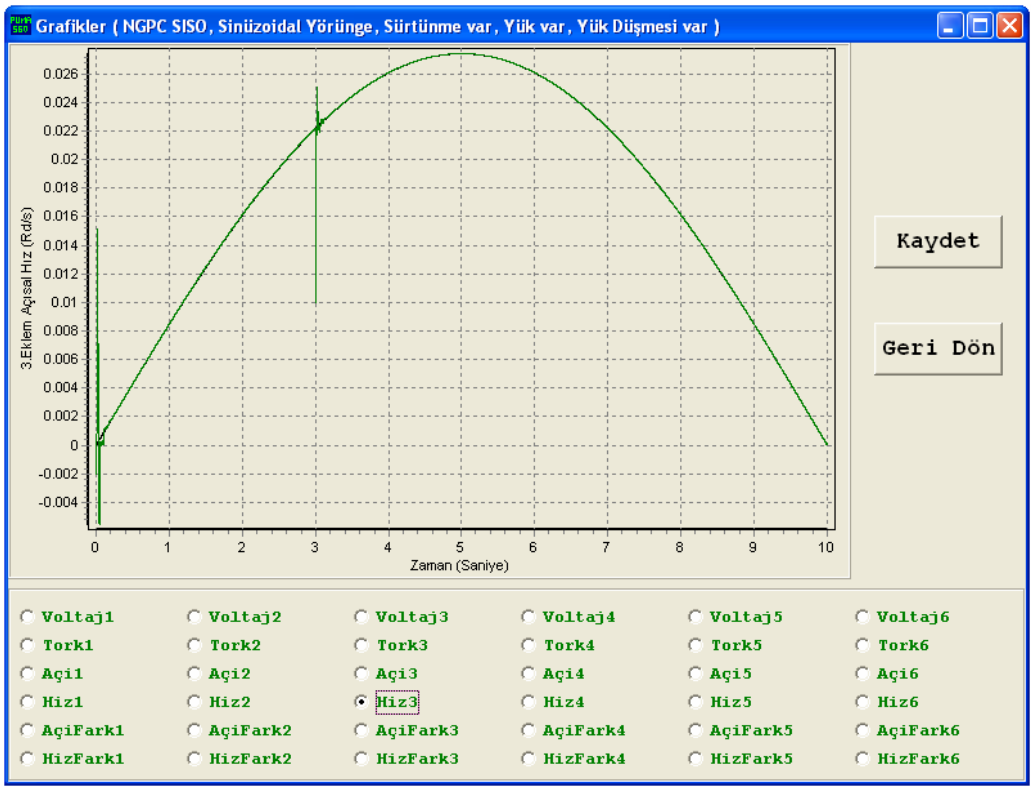

Şekil 4.10. Paket programın grafik ara yüzünden bir kesit (1)



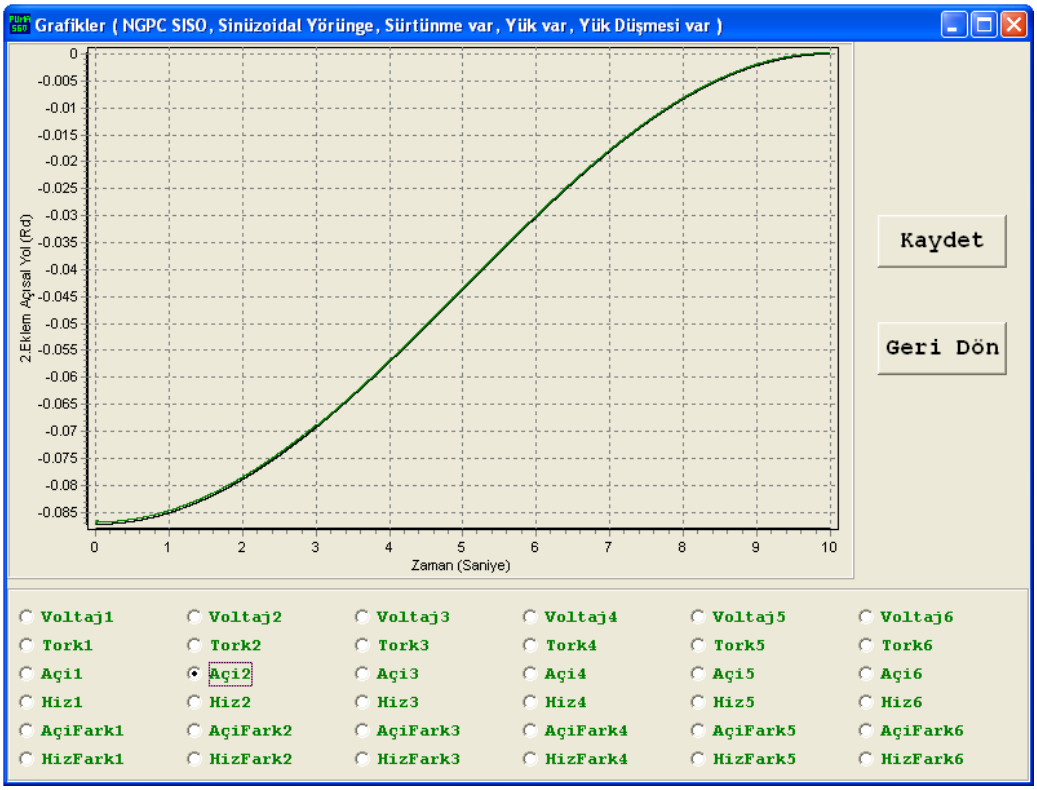

Şekil 4.11. Paket programın grafik ara yüzünden bir kesit (2)



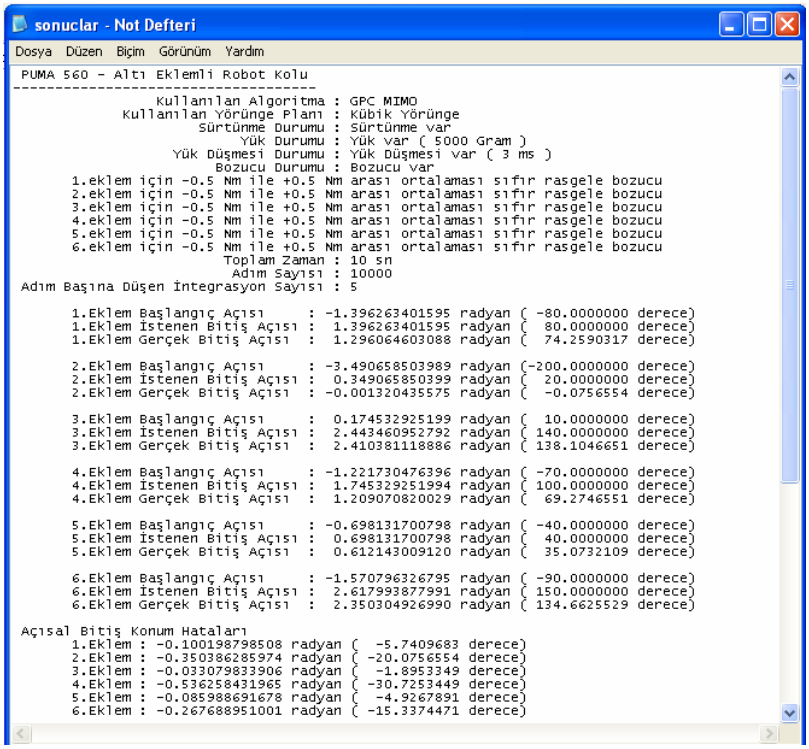

Şekil 4.12. Paket programın grafik ara yüzünden bir kesit (3)



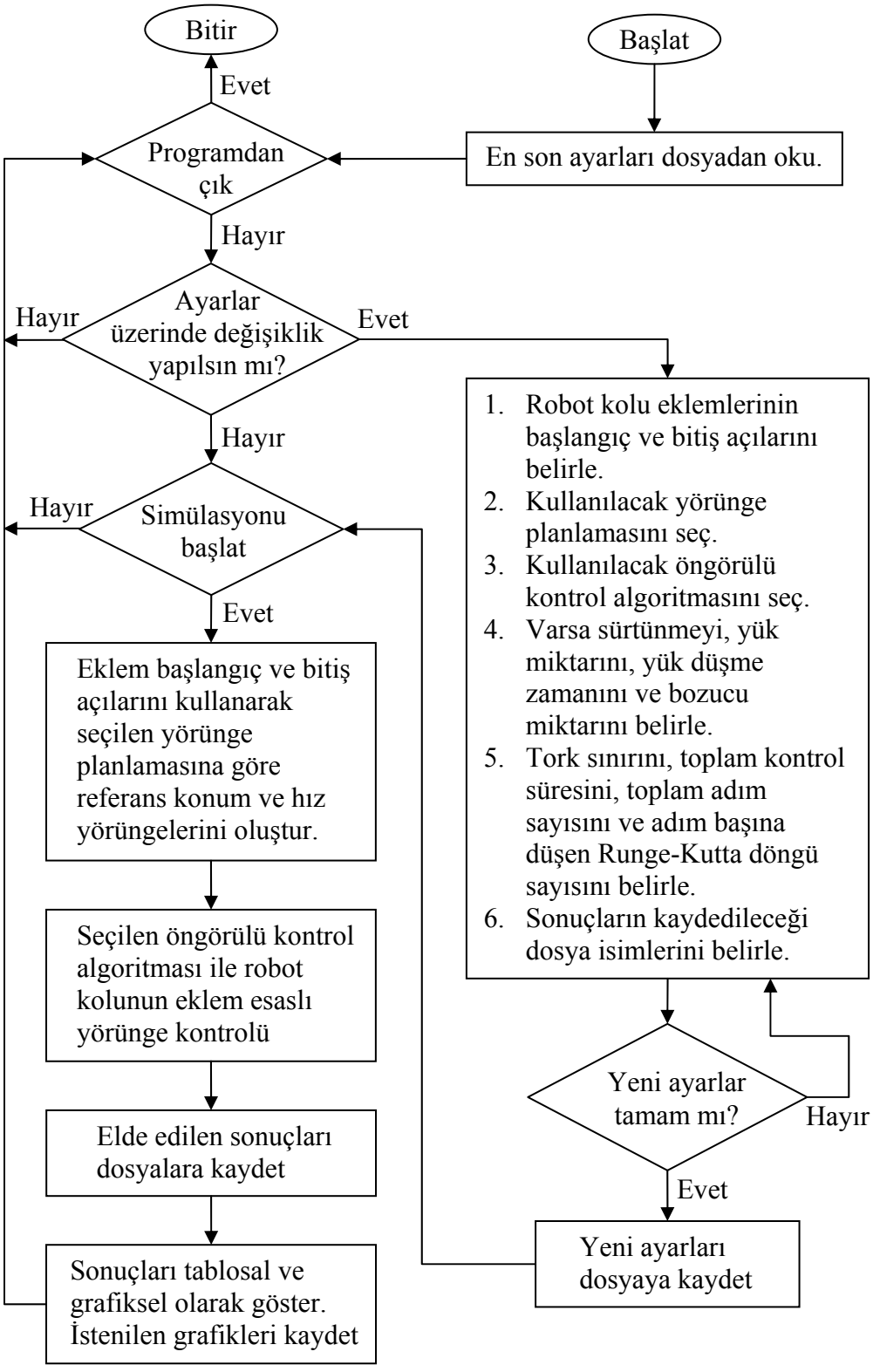

Şekil 4.13. Paket programın grafik ara yüzünden bir kesit (4)




Şekil 4.14. Paket programın grafik ara yüzünden bir kesit (5)



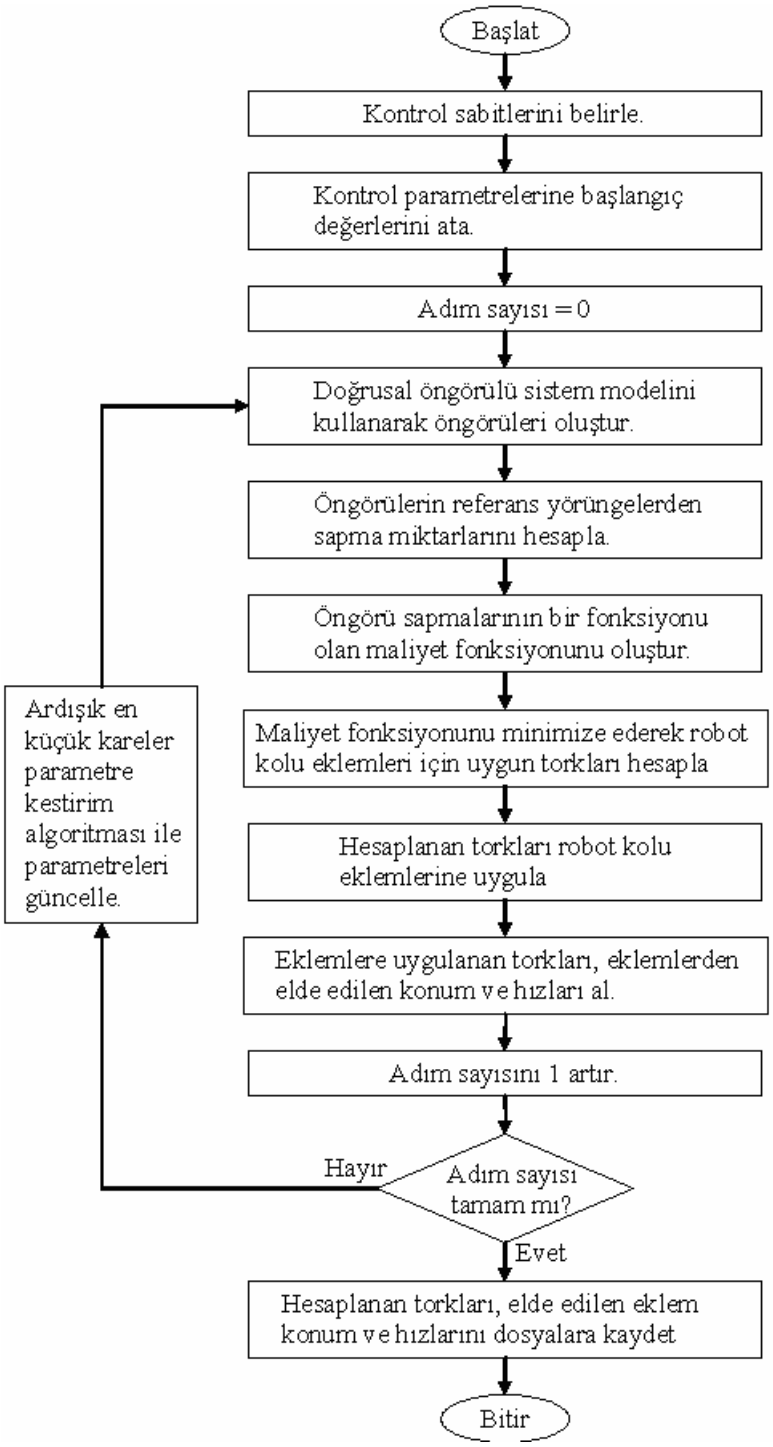

Şekil 4.15. Paket programın grafik ara yüzünden bir kesit (6)



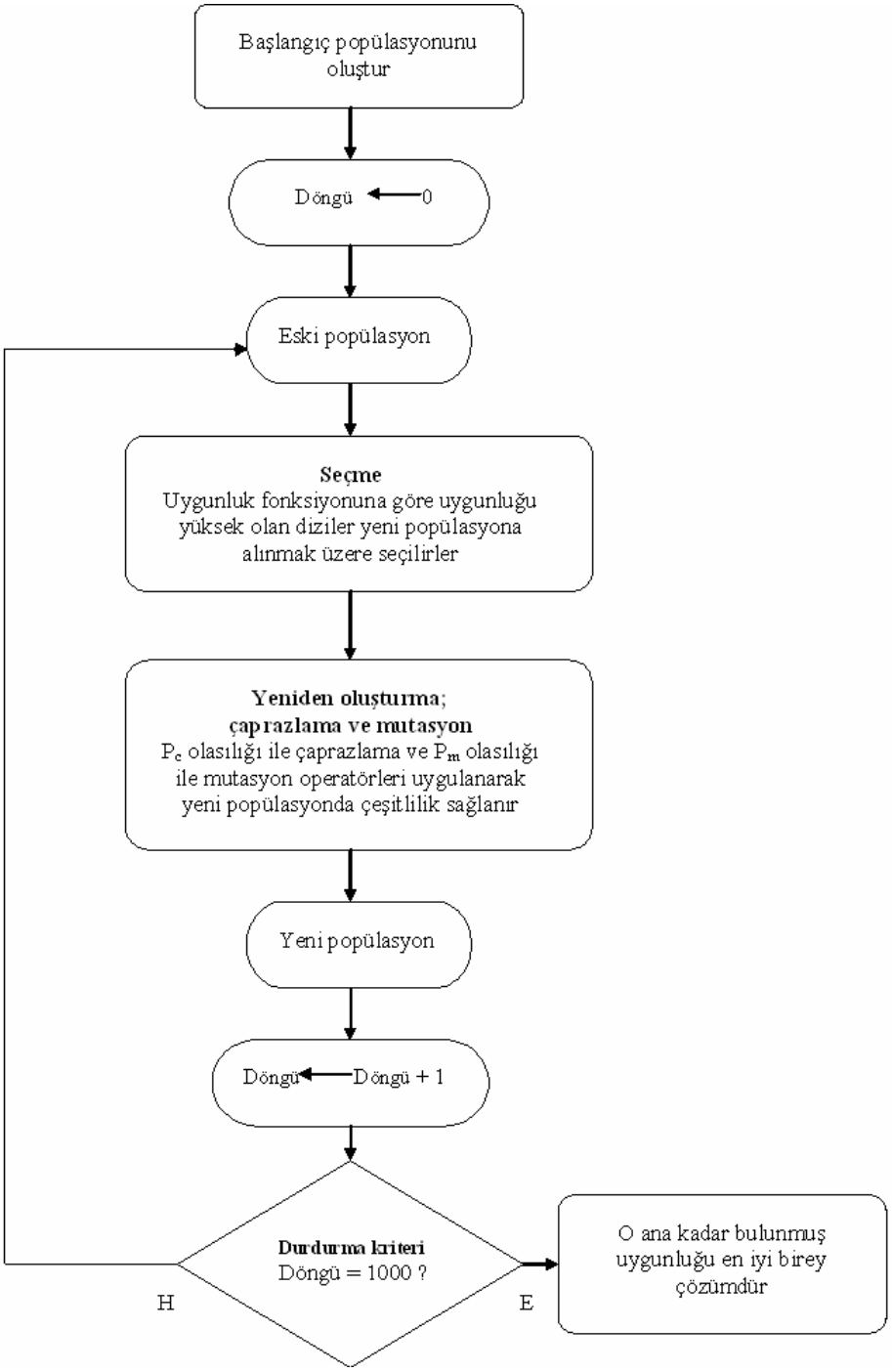

Şekil 4.16. Ana programın akış diyagramı

Ana program son ayarların dosyadan okutulmasıyla başlar ve değişiklikler kullanıcıdan istenir. Kullanıcı ilgili değişiklikleri ve kontrol parametrelerini

belirledikten sonra seçilen kontrol altprogramına dallanılır. Altprogramın ürettiği kontrol verileri her kontrol adımında dinamik modele kontrol süresi boyunca uygulanır. Elde dilen tüm veriler tablosal ve grafiksel olarak gösterilir ve program sonlandırılır.

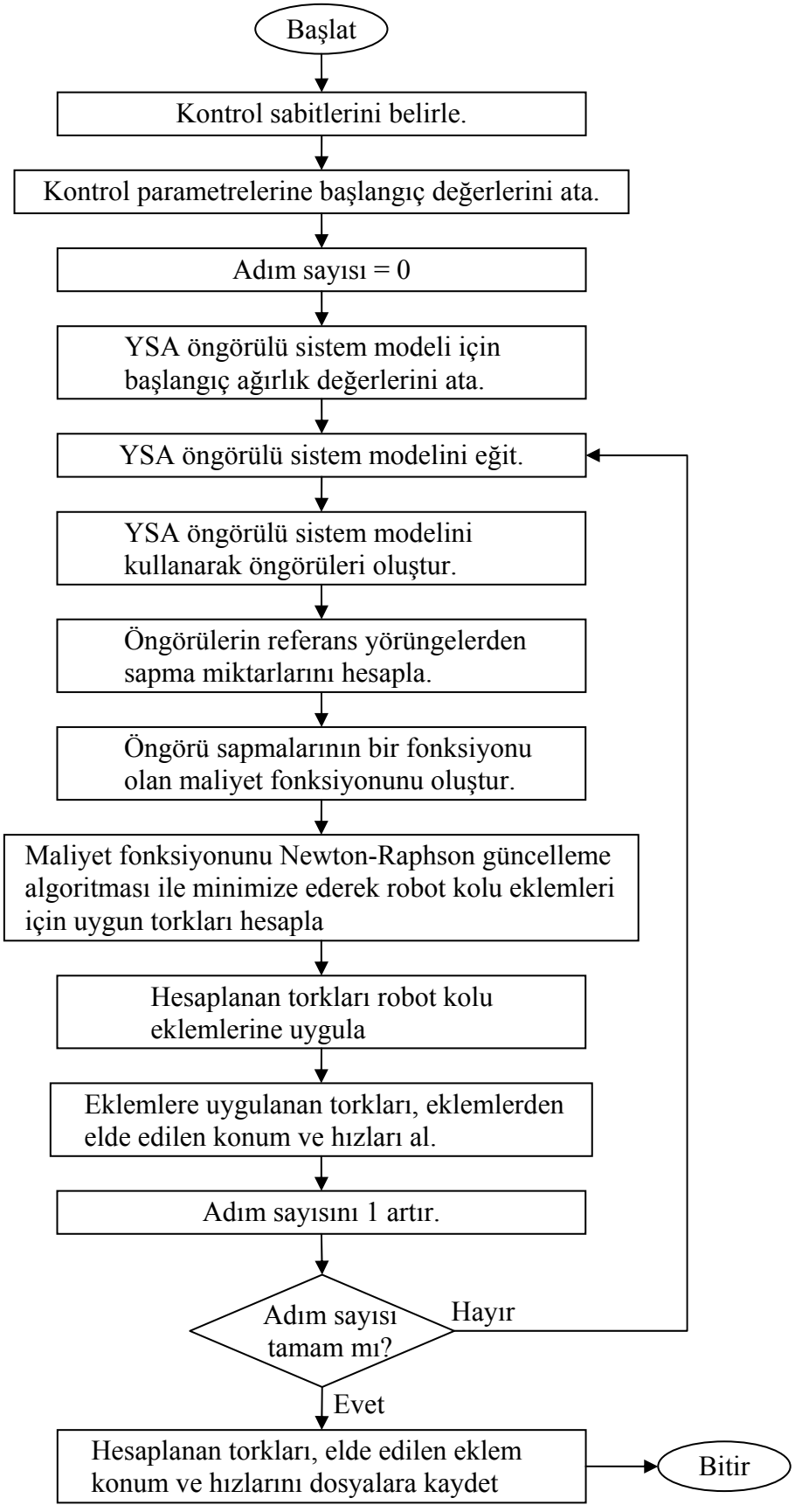



Şekil 4.17. GPC algoritmasının akış diyagramı

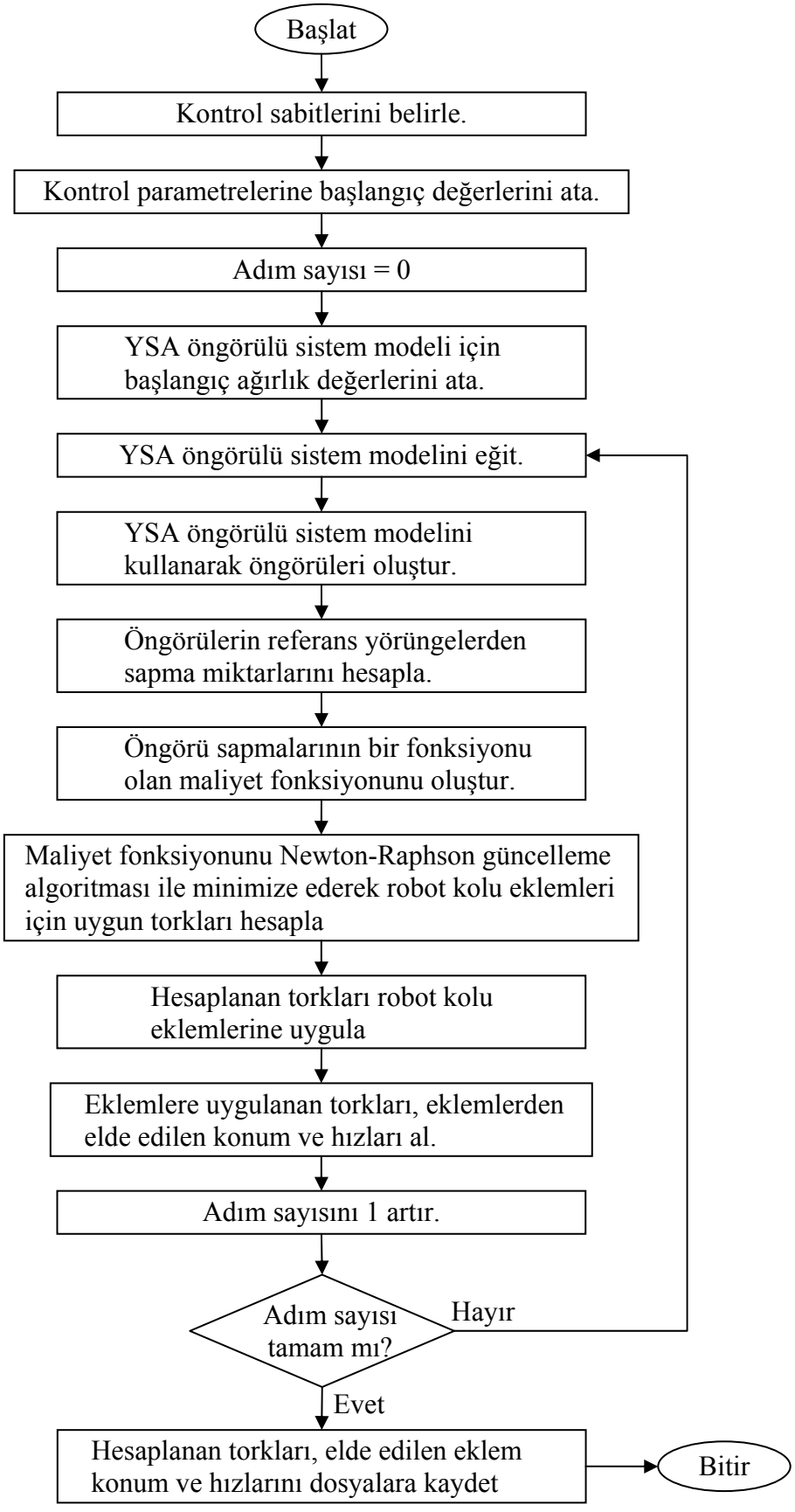



Şekil 4.18. SGA-GPC algoritmasının akış diyagramı

GPC ve SGA-GPC alt programları benzer yapıya sahiptirler. Aralarındaki tek fark kontrol giriş sinyalinin hesaplanma aşamasıdır. GPC kontrol sinyalini üretmek için maliyet fonksiyonunun girişe göre türevini alarak hesaplama yaparken, SGA-GPC bu işlem için genetik işlemleri kullanır. Her iki altprogram parametre kestirimi için en küçük kareler yöntemini kullanmaktadır.




Şekil 4.19. NGPC algoritmasının akış diyagramı




Şekil 4.20. ENGPC algoritmasının akış diyagramı

NGPC ve ENGPC algoritmaları yapısal olarak aynı olup kullanılan YSA modeli yönünden farklıdırlar. Kontrol parametreleri atandıktan sonra YSA eğitilmeye başlar. Bu eğitim kontrol boyunca devam eder ve her kontrol adımında ağırlıklar geri yayılım ile güncellenir. Eğitilen ağ üzerinden öngörüler yapılarak maliyet fonksiyonuna yerleştirilir. Maliyet fonksiyonu minimizasyonu Newton-Raphson metodu ile gerçeklenir. Bu işlem sonrasında elde edilen kontrol sinyali robot kolu dinamik modeline uygulanır. Dinamik modelden elde alınan açı ve hız bilgileri hem YSA eğitimi hem de sonraki öngörüler için tekrardan kontrolöre iletilir. Bu işlemler kontrol bitene kadar tekrarlanır.

### 4.2. Simülasyon Sonuçları

Bu bölümde, öngörülü kontrol sınıfına ait olan GPC, basit genetik algoritma uyarlamalı GPC (SGA-GPC), NGPC ve yinelenen Elman ağ uyarlamalı ENGPC algoritmalarının altı eklemli endüstriyel bir robot kolu manipülatörüne farklı örnek ve durumlar için uygulanmalarıyla ilgili elde edilen sonuçlar yorumlanacaktır. Algoritmalar tek giriş-tek çıkış (SISO-Single Input Single Output) ve çok giriş-çok çıkış (MIMO-Multiple Input Multiple Output) olmak üzere iki şekilde tasarlanmıştır. NGPC-MIMO ve ENGPC-MIMO algoritmaları hariç diğer algoritmalar için robot kolu kontrolü simülasyon sonuçları tablosal ve grafiksel olarak verilmiştir. NGPC-MIMO ve ENGPC-MIMO algoritmalarında kontrol esnasında kilitlendiğinden başarısız sonuçlar alınmıştır ve bu algoritmalar ile yapılan çalışmalar yansıtılmamıştır.

Robot kolunun hareketlerinin dinamik olarak modellenmesinde Lagrange-Euler yöntemi kullanılmıştır. Dinamik modellemeye sürtünme, yük taşıma ve taşınan yükün taşıma esnasında düşmesi durumları da ilave edilmiştir. Ayrıca, eklem torklarına rasgele bozucular eklenerek bozuculu durumlar karşısında kontrol algoritmalarının performansları test edilmiştir.

GPC algoritmasının SISO ve MIMO uygulamaları için $\lambda = 5*10^{-6}$, $\mu = 1$, $\delta = 10^{-5}$, $n_a = 2$ ($A = \mathrm{I}_n + A_1 q^{-1} + A_2 q^{-2}$), $n_b = 1$ ($B = B_0 + B_1 q^{-1}$) olarak alınmıştır.

Basitleştirilmiş Genetik Algoritma uyarlamalı GPC (SGA-GPC) algoritmasında arama uzayının boyutu işlem süresini uzatacağından tork değişimi -186 ile +186 *Nm* arasında sınırlandırılmıştır. Popülasyon kümesindeki her bir birey 21 bit 'lik kromozomlar tarafından temsil edilmiştir. Genetik algoritma işlem sürecinde uygunluk değeri sıralaması, tek noktalı çaprazlama ve tek nokta mutasyon operatörleri tercih edilmiştir. Ayrıca genetik algoritma ile işlem süreci uzun olduğundan, gerçek-zaman (real-time) uygulamalar göz önünde bulundurularak GA işlem süresi 100 döngü ile sınırlandırılmıştır.

NGPC algoritmasının SISO ve MIMO uygulamaları için ise $n_d = 2$ ($u(n)$, $u(n-1)$, $u(n-2)$), $d_d = 3$ ($y(n-1)$, $y(n-2)$, $y(n-3)$), $\lambda = 5*10^{-6}$, $\varepsilon = 10^{-7}$, $s = 10^{-21}$ ($u \cong u_{\max}$ veya $u \cong u_{\min}$ ise $\varepsilon$ ve $s$ parametreleri etkili), Gizli katman eleman sayısı = 5 (Gizli katman eleman sayısını büyük seçmenin fazla bir faydası yoktur, aksine kontrol algoritmasının hızını yavaşlatır) olarak alınmıştır. Ayrıca, her bir kontrol algoritması için öngörü ufuğu $N_1 = 1$'den $N_2 = 3$'e kadar, kontrol ufuğu ise $N_u = 1$ olarak alınmıştır. Öngörü ve kontrol ufuklarının büyük seçilmelerinin kontrol hassasiyetine fazla bir etkisi yoktur. Aksine, hesaplama yükünü artıracaklarından gerçek zamanlı kontrolü güçleştirirler.

NGPC 'yi geliştirmek için önerilen dinamik ağ uyarlamalı ENGPC algoritmasının SISO ve MIMO tasarımlarındaki parametre seçiminde kıyaslama yapmak için NGPC ile aynı parametre değerleri seçilmiştir. Aynı şekilde gizli katman eleman sayısı 5 olarak seçilmiştir. İlave olarak, gizli katmanı girişe bağlayan bağlam gizli katmanı (context hidden layer) eklenmiştir.

Eklemlere uygulanacak her bir tork için $u_{\min} = -186.4\ Nm$ ve $u_{\max} = +186.4\ Nm$ sınırlaması getirilmiştir [49]. Toplam simülasyon süresi 10 *sn*, örnekleme periyodu ise 1 *ms* olarak alınmıştır. Bu durumda toplam kontrol adım sayısı 10000 olur. Her bir kontrol adımı başına düşen Runge-Kutta adım sayısı 4 olarak alınmıştır. Kontrol adım zaman süresi çok kısa olduğundan bu sayı yeterli görülmüştür.

Robot kolunun her bir eklemine ait takip etmesi istenen konum referans ve hız referans yörüngeleri sinüzoidal yörünge esaslarına göre belirlenmiştir. Her bir eklem yörüngesi diğer eklemlerden bağımsız olacak şekilde çıkartılmıştır. Bu sayede uç eleman pozisyonu rahatlıkla hesaplanabilmektedir.

Yük taşıma durumları için taşınan yük $5\ Kg$ , yük düşmesi durumları için yük düşme zamanı $3\ sn$ (yani 3000 adım) olarak alınmıştır. Yük düşme durumu ile kontrol algoritmalarının değişken yük karşısında adaptif özellikleri sınanması hedeflenmiştir. Bozuculu durumlar için adım başına her bir torka $-0.5\ Nm$ ile $+0.5\ Nm$ arasında rasgele bozucu ilave edilmiştir. Bu sayede güçleşen kontrol durumlarında tasarlanan algoritmalarının başarımları test edilmiştir.

Bu bölümde, kontrol algoritmalarının altı eklemli robot koluna uygulanmaları ile elde edilen sonuçlardan birbirinden tamamen farklı aşağıdaki 3 örnek ve 4 durum için olanları verilecek ve yorumlanacaktır. 8 algoritma, 3 örnek ve 4 durum için toplam 96 farklı simülasyon çalışması sunulmuştur.

Örnekler ( 3 farklı örnek ),

Örnek 1 ) Başlangıç eklem açıları;

$\theta_1 = 0,\ \theta_2 = -5,\ \theta_3 = 20,\ \theta_4 = -20,\ \theta_5 = -5,\ \theta_6 = 0$ Derece

Bitiş eklem açıları;

$\theta_1 = 10,\ \theta_2 = 0,\ \theta_3 = 30,\ \theta_4 = -10,\ \theta_5 = 20,\ \theta_6 = 40$ Derece

Örnek 2 ) Başlangıç eklem açıları;

$\theta_1 = -10,\ \theta_2 = -45,\ \theta_3 = 30,\ \theta_4 = 20,\ \theta_5 = 50,\ \theta_6 = -60$ Derece

Bitiş eklem açıları;

$\theta_1 = 0,\ \theta_2 = -30,\ \theta_3 = 70,\ \theta_4 = -30,\ \theta_5 = 20,\ \theta_6 = 80$ Derece

Örnek 3 ) Başlangıç eklem açıları;

$\theta_1 = 30,\ \theta_2 = 10,\ \theta_3 = 40,\ \theta_4 = -40,\ \theta_5 = -50,\ \theta_6 = 150$ Derece

Bitiş eklem açıları;

$\theta_1 = 70,\ \theta_2 = -5,\ \theta_3 = 60,\ \theta_4 = 20,\ \theta_5 = 70,\ \theta_6 = 260$ Derece

Durumlar ( 4 farklı durum ),

D1: Yük ve sürtünme yok

D2: Yük ve sürtünme var

D3: Yük ve sürtünme var, yük düşmesi var

D4: Yük ve sürtünme var, yük düşmesi var, bozucu var

Bu örnek ve durumlar için simülasyon sonunda elde edilen açısal hatalar, hız hataları ve uç nokta koordinat hataları tablolar halinde aşağıda verilmiştir. Tablolardan sonra, yer darlığı sebebiyle bu sonuçlardan sadece Örnek 1'e ait grafikler verilmiştir. Ayrıca, bu tablolar ve grafikler kısaca yorumlanmıştır.

Tablo 4.1 Örnek 1 için eklemlere ait açısal bitiş konum hataları

| Kontrol | Eklem No | Durumlar | | | | Birim |
|---|---|---|---|---|---|---|
| | | D1 | D2 | D3 | D4 | |
| GPC SISO | 1 | 0.000089602626 | 0.000068673471 | 0.000089559884 | 0.000170535328 | *rd* |
| | 2 | 0.001109374788 | 0.001110285521 | 0.000884313461 | 0.001600483619 | |
| | 3 | -0.000266681963 | -0.00020077362 | -0.000284931336 | -0.000290633430 | |
| | 4 | 0.000011657874 | 0.000000932011 | 0.000012213894 | -0.000006273836 | |
| | 5 | 0.000037717935 | -0.00000442332 | 0.000052444163 | -0.000196691549 | |
| | 6 | 0.000007147880 | 0.000002478735 | 0.000007147880 | -0.000343092890 | |
| GPC MIMO | 1 | 0.013234116180 | -0.034869829598 | -0.027540381879 | 0.006198692862 | |
| | 2 | -0.012392547054 | 0.018818789094 | 0.004856863816 | -0.021377931266 | |
| | 3 | -0.012280195849 | -0.022633348846 | -0.028038863541 | -0.052245132819 | |
| | 4 | 0.015102167127 | -0.031467188881 | -0.037458706364 | -0.030376763321 | |
| | 5 | 0.036629995376 | -0.049438216881 | -0.068537067501 | -0.066244342683 | |
| | 6 | 0.061830034187 | 0.031830695568 | 0.017497512267 | -0.040100478376 | |
| SGAGPC SISO | 1 | 0.000085804643 | 0.000062689911 | -0.000065350966 | -0.000408937164 | |
| | 2 | 0.000916693456 | 0.000840560421 | 0.000634479275 | 0.001207880804 | |
| | 3 | -0.000156480997 | 0.000079017629 | 0.000200960158 | -0.000171777512 | |
| | 4 | 0.000007027452 | -0.000017120536 | 0.000016222040 | -0.000046661943 | |
| | 5 | -0.000004917792 | -0.000073048216 | -0.000026379939 | -0.000103528233 | |
| | 6 | -0.000012924682 | -0.000000382932 | 0.000004793529 | 0.000324904345 | |
| SGAGPC MIMO | 1 | 0.013999841539 | -0.013291115400 | -0.010484066551 | 0.471373361342 | |
| | 2 | -0.008216157148 | -0.000498455414 | 0.002399387612 | 1.385677142477 | |
| | 3 | -0.007771156467 | -0.018375935764 | -0.008144037706 | 0.241154680462 | |
| | 4 | 0.014820156778 | -0.009298511542 | -0.021756004762 | 0.653750275025 | |
| | 5 | 0.038152607054 | -0.055411902764 | -0.055422031563 | 1.167871491519 | |
| | 6 | 0.062147993874 | -0.030571497444 | 0.005282926513 | 0.628867672588 | |
| NGPC SISO | 1 | 0.000000579601 | -0.000059556313 | -0.000060132625 | 0.000106186925 | |
| | 2 | 0.000280764676 | 0.000479261062 | 0.000250001028 | 0.000229595245 | |
| | 3 | 0.000059610728 | 0.000187631402 | 0.000025410624 | 0.000136324882 | |
| | 4 | 0.000000102744 | -0.000009484042 | -0.000015468056 | -0.000196841972 | |
| | 5 | 0.000001435818 | 0.000125161254 | -0.000010540757 | -0.001377181510 | |
| | 6 | 0.000000079300 | -0.000012333374 | -0.000011607182 | -0.000461015193 | |
| NGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |
| ENGPC SISO | 1 | 0.000001520553 | -0.000032443683 | -0.000032180638 | -0.000031680198 | |
| | 2 | 0.000167356162 | 0.000286018024 | 0.000149058418 | 0.000193717199 | |
| | 3 | 0.000034987114 | 0.000108135473 | 0.000025011214 | 0.000018760836 | |
| | 4 | 0.000000063539 | -0.000004563744 | -0.000008186837 | -0.000103860330 | |
| | 5 | 0.000034308588 | 0.000067019849 | 0.000024229469 | 0.000104392929 | |
| | 6 | 0.000037975596 | 0.000039490868 | 0.000039490868 | -0.000212651010 | |
| ENGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |

1. Durum için 2. eklem üzerinden karşılaştırma yapılırsa 0.000167356162 *rd* ile en düşük hata ENGPC SISO algoritmasına aittir. Bunu, NGPC SISO 0.000280764676 *rd* , SGA-GPC SISO 0.000916693456 *rd* , GPC SISO 0.001109374788 *rd* , SGA-GPC MIMO -0.008216157148 *rd* ve GPC MIMO -0.012392547054 *rd* 'lık hata ile takip etmektedir.

3. Durum için 3. eklem üzerinden karşılaştırma yapılırsa 0.000025011214 *rd* ile en düşük hata ENGPC SISO algoritmasına aittir. Bunu, NGPC SISO 0.000025410624 *rd* , SGA-GPC SISO 0.000200960158 *rd* , GPC SISO -0.000284931336 *rd* , SGA-GPC MIMO -0.008144037706 *rd* ve GPC MIMO -0.028038863541 *rd* 'lık hata ile takip etmektedir.

Diğer durum ve eklemler için de sıra aşağı yukarı bu şekildedir. Açısal hatalar oldukça düşüktür. Yüklü, yük düşmeli ve bozuculu durumlarda kontrol zorlaşmasına rağmen özellikle SISO yapıdaki algoritmalar daha başarılı olmuşlardır.

Tablo 4.2. Örnek 1 için eklemlere ait açısal hız hatalarının kareleri toplamı

| Kontrol | Eklem No | Durumlar | | | | Birim |
|---|---|---|---|---|---|---|
| | | D1 | D2 | D3 | D4 | |
| GPC SISO | 1 | 0.001688112152 | 0.000435825521 | 0.001689331374 | 0.009183039552 | $(rd/s)^2$ |
| | 2 | 0.194058171184 | 0.229788528784 | 0.197652449009 | 0.380999967396 | |
| | 3 | 0.010826067371 | 0.004758453934 | 0.010852898391 | 0.028926168979 | |
| | 4 | 0.000453965210 | 0.000002294924 | 0.000453965310 | 0.053647668581 | |
| | 5 | 0.007136095041 | 0.000019328339 | 0.007202387767 | 0.077114339128 | |
| | 6 | 0.000192973559 | 0.000003034256 | 0.000192973559 | 0.060383424562 | |
| GPC MIMO | 1 | 0.047215570320 | 0.254581181289 | 0.190456678902 | 0.045885271437 | |
| | 2 | 0.177284516250 | 0.456961617689 | 0.414685716711 | 1.107217456044 | |
| | 3 | 0.061692840623 | 0.316016794115 | 0.346847658909 | 0.543411247278 | |
| | 4 | 0.029976920923 | 0.801463614033 | 0.842986267496 | 0.185603227706 | |
| | 5 | 0.170715290033 | 0.406377493681 | 0.737394150097 | 0.701465315627 | |
| | 6 | 0.484058012097 | 0.237395194918 | 0.162638389305 | 0.240207576523 | |
| SGAGPC SISO | 1 | 0.000616176879 | 0.003371749726 | 0.000665648892 | 0.011194501666 | |
| | 2 | 0.232488991682 | 0.114057913795 | 0.110211501310 | 0.312624842726 | |
| | 3 | 0.003996953228 | 0.007886915964 | 0.025180501804 | 0.005175431955 | |
| | 4 | 0.000048236936 | 0.000112310459 | 0.000182194553 | 0.052854364843 | |
| | 5 | 0.000053072744 | 0.000744360760 | 0.001121582753 | 0.077445418439 | |
| | 6 | 0.000026408177 | 0.000114962158 | 0.000355713134 | 0.061257354217 | |
| SGAGPC MIMO | 1 | 0.043274840245 | 0.098942214371 | 0.299708941733 | 0.471373361342 | |
| | 2 | 0.489084107264 | 0.296861496434 | 1.067973959866 | 1.385677142477 | |
| | 3 | 0.056742622138 | 0.057707983388 | 0.067645465056 | 0.241154680462 | |
| | 4 | 0.075439257676 | 0.038190751211 | 0.097534748435 | 0.653750275025 | |
| | 5 | 0.256255068804 | 0.533669768587 | 0.449676598264 | 1.167871491519 | |
| | 6 | 0.478199772048 | 0.161189348663 | 0.112522932211 | 0.628867672588 | |
| NGPC SISO | 1 | 0.000058404882 | 0.000046859910 | 0.000061824613 | 0.004141260289 | |
| | 2 | 0.010532247905 | 0.019382170945 | 0.022385416869 | 0.023145113090 | |
| | 3 | 0.000253281935 | 0.002603983693 | 0.003644961082 | 0.009179293384 | |
| | 4 | 0.000000027283 | 0.000005109848 | 0.000005110883 | 0.068870051750 | |
| | 5 | 0.000000199197 | 0.000369725841 | 0.000776665039 | 0.104702698998 | |
| | 6 | 0.000000552937 | 0.000007573439 | 0.000007573439 | 0.101267077380 | |
| NGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |
| ENGPC SISO | 1 | 0.000038701822 | 0.000019696375 | 0.000031085952 | 0.002096285881 | |
| | 2 | 0.005695893990 | 0.009890234916 | 0.011870792905 | 0.012284853600 | |
| | 3 | 0.000128479514 | 0.001302775449 | 0.001770859034 | 0.004366153475 | |
| | 4 | 0.000000108638 | 0.000002113777 | 0.000002126275 | 0.009745953098 | |
| | 5 | 0.000001787827 | 0.000187382109 | 0.000374673328 | 0.014931463418 | |
| | 6 | 0.000006072001 | 0.000008618357 | 0.000008618357 | 0.012870224291 | |
| ENGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |

2. Durum için 2. eklem üzerinden karşılaştırma yapılırsa 0.009890234916 $(rd/s)^2$ ile en düşük açısal hız farkları kare toplam hatası ENGPC SISO algoritmasına aittir. Bunu sırasıyla, NGPC SISO 0.019382170945 $(rd/s)^2$, SGA-GPC SISO 0.114057913795 $(rd/s)^2$, GPC SISO 0.229788528784 $(rd/s)^2$, SGA-GPC MIMO 0.296861496434 $(rd/s)^2$ ve GPC MIMO 0.456961617689 $(rd/s)^2$ olarak takip etmektedir.

4. Durum için 3. eklem üzerinden karşılaştırma yapılırsa 0.004366153475 $(rd/s)^2$ ile en düşük açısal hız farkları kare toplam hatası NGPC SISO algoritmasına aittir. Bunu sırasıyla, NGPC SISO 0.009179293384 $(rd/s)^2$, SGA-GPC SISO 0.002715729272 $(rd/s)^2$, GPC SISO 0.028926168979 $(rd/s)^2$, SGA-GPC MIMO 0.241154680462 $(rd/s)^2$ ve GPC MIMO 0.543411247278 $(rd/s)^2$ olarak takip etmektedir.

Diğer durum ve eklemler için de sıra aşağı yukarı bu şekildedir. Açısal hız farklarının kare toplam hataları oldukça düşüktür. Hız hatalarının kareleri eklemlerdeki sarsıntı ile doğru orantılıdır [40]. Dolayısıyla düşük hız hataları eklemlerdeki sarsıntının çok az olduğunu göstermektedir. Kontrol algoritmaları çok az sarsıntı ile eklemleri kontrol etmişlerdir.

Tablo 4.3. Örnek 1 için uç nokta koordinat hataları

| Kontrol | Uç Nokta Hatası | Durumlar | | | | Birim |
|---|---|---|---|---|---|---|
| | | D1 | D2 | D3 | D4 | |
| **GPC SISO** | $\Delta x$ | 0.227171224522 | 0.359128515457 | 0.327484215931 | 0.491361519461 | *mm* |
| | $\Delta y$ | 0.101081021832 | 0.110379817039 | 0.118933966143 | 0.205085282941 | |
| | $\Delta z$ | -0.529097721161 | -0.699304641528 | -0.684549773110 | -0.999722685716 | |
| | Toplam $\Delta$ | 0.584609730301 | 0.793841279026 | 0.768114309104 | 1.132670103967 | |
| **GPC MIMO** | $\Delta x$ | -12.48913777706 | 5.152011030554 | -5.475977051650 | -35.56818141033 | |
| | $\Delta y$ | 6.694270011115 | -22.955699421708 | -19.440600248253 | -2.102902693665 | |
| | $\Delta z$ | 9.606018887252 | -5.098877936477 | 6.226457188081 | 28.47078549966 | |
| | Toplam $\Delta$ | 17.119211788473 | 24.072928982777 | 21.135090058894 | 45.60815010001 | |
| **SGAGPC SISO** | $\Delta x$ | 0.362686582328 | 0.362480662205 | 0.292781642439 | 0.532407420640 | |
| | $\Delta y$ | 0.019787964899 | 0.107168564941 | 0.110521064885 | -0.185945481745 | |
| | $\Delta z$ | -0.475003190070 | -0.582313183099 | -0.579551442376 | -0.767790956304 | |
| | Toplam $\Delta$ | 0.597964172114 | 0.694237693441 | 0.658646999759 | 0.952647015592 | |
| **SGAGPC MIMO** | $\Delta x$ | -1.796992992597 | -6.616683188736 | -8.997410102260 | -28.91894956820 | |
| | $\Delta y$ | -7.279430556058 | -9.763832105623 | 7.859632832897 | 2.352577239485 | |
| | $\Delta z$ | 2.675446126819 | 6.986075084992 | 5.700168619435 | 25.87510958969 | |
| | Toplam $\Delta$ | 7.960986434696 | 13.708324438095 | 13.237036643683 | 38.87623387205 | |
| **NGPC SISO** | $\Delta x$ | 0.129397029617 | 0.164068437469 | 0.144219958957 | 0.296818104526 | |
| | $\Delta y$ | -0.018519085757 | 0.012725260572 | 0.025820117129 | 0.010236042507 | |
| | $\Delta z$ | -0.173965425458 | -0.202452731078 | -0.208190767961 | -0.373606382135 | |
| | Toplam $\Delta$ | 0.217601739574 | 0.260897475556 | 0.254577043101 | 0.477270879599 | |
| **NGPC MIMO** | $\Delta x$ | Kontrol Başarısız | | | | |
| | $\Delta y$ | | | | | |
| | $\Delta z$ | | | | | |
| | Toplam $\Delta$ | | | | | |
| **ENGPC SISO** | $\Delta x$ | 0.081520833042 | 0.100297580419 | 0.084840009740 | 0.174481012888 | |
| | $\Delta y$ | -0.008021953074 | -0.006957682687 | 0.015681181819 | 0.007857281727 | |
| | $\Delta z$ | -0.107476162968 | -0.139312026465 | -0.122659367244 | -0.221685848941 | |
| | Toplam $\Delta$ | 0.135133724722 | 0.171801789001 | 0.149963485851 | 0.282223273943 | |
| **ENGPC MIMO** | $\Delta x$ | Kontrol Başarısız | | | | |
| | $\Delta y$ | | | | | |
| | $\Delta z$ | | | | | |
| | Toplam $\Delta$ | | | | | |

3. Durum için x-ekseni üzerinden karşılaştırma yapılırsa 0.084840009740 *mm* ile en düşük eksensel hata ENGPC SISO algoritmasına aittir. Bunu sırası ile NGPC SISO 0.144219958957 *mm*, SGA-GPC SISO 0.292781642439 *mm*, GPC SISO

0.327484215931 *mm*, SGA-GPC MIMO -8.997410102260 *mm* ve GPC MIMO -12.48913777706 *mm* olarak takip etmektedir.

1. Durum için z-ekseni üzerinden karşılaştırma yapılırsa -0.107476162968 *mm* ile en düşük eksensel hata ENGPC SISO algoritmasına aittir. Bunu sırası ile NGPC SISO -0.173965425458 *mm*, SGA-GPC SISO -0.475003190070 *mm*, GPC SISO 0.584609730301 *mm*, SGA-GPC MIMO 7.960986434696 *mm* ve GPC MIMO 21.135090058894 *mm* olarak takip etmektedir.

2. Durum için uç nokta toplam koordinat hatası üzerinden karşılaştırma yapılırsa 0.171801789001 *mm* ile en düşük eksensel hata ENGPC SISO algoritmasına aittir. Bunu sırası ile NGPC SISO 0.260897475556 *mm*, SGA-GPC SISO 0.694237693441 *mm*, GPC SISO 0.793841279026 *mm*, SGA-GPC MIMO 13.708324438095 *mm* ve GPC MIMO 24.072928982777 *mm* olarak takip etmektedir.

Diğer durum ve eksenler için de sıralama bu şekildedir. Açısal hatalar az olduğundan doğal olarak eksensel hatalar da azdır.

Örnek 2 ‘ye ait sonuçlar Tablo 4.4, 4.5 ve 4.6 ‘da verilmiştir.

Tablo 4.4. Örnek 2 için eklemlere ait açısal bitiş konum hataları

| Kontrol | Eklem No | Durumlar | | | | Birim |
|---|---|---|---|---|---|---|
| | | D1 | D2 | D3 | D4 | |
| GPC SISO | 1 | 0.000183690962 | 0.000298352862 | 0.000298387844 | 0.000189158041 | *rd* |
| | 2 | 0.001089906779 | 0.001455416284 | 0.001248735079 | 0.002057784329 | |
| | 3 | -0.000372551612 | -0.000685050762 | -0.000689948476 | -0.000242008313 | |
| | 4 | -0.000001706802 | 0.000115831420 | 0.000118207440 | 0.000004563460 | |
| | 5 | 0.000004546844 | 0.000947101006 | 0.000969579561 | 0.000279011880 | |
| | 6 | 0.000006018116 | 0.000008612976 | 0.000008612976 | -0.000077077384 | |
| GPC MIMO | 1 | 0.006745402139 | -0.040374227249 | -0.072875997725 | 0.000500613043 | |
| | 2 | -0.009451996119 | -0.035854096377 | 0.014137302791 | -0.040565849134 | |
| | 3 | -0.064591405648 | -0.102364882784 | -0.052287083598 | -0.107059156441 | |
| | 4 | -0.078923168734 | 0.041841464705 | 0.092124231538 | 0.160851338893 | |
| | 5 | -0.047845039998 | 0.081075093180 | 0.078919883613 | 0.052974904980 | |
| | 6 | 0.218552377425 | -0.059788182291 | -0.048289066488 | -0.170900967432 | |
| SGAGPC SISO | 1 | 0.000042285695 | -0.000016416666 | -0.000652896422 | 0.000338764407 | |
| | 2 | 0.000763803639 | 0.000945048136 | 0.000683746656 | 0.001907165781 | |
| | 3 | -0.000429156845 | -0.000058586040 | -0.001001803285 | -0.000758062539 | |
| | 4 | -0.000013399447 | 0.000007210349 | 0.000179004222 | 0.000027109796 | |
| | 5 | 0.000002915307 | 0.000008800574 | 0.000555891078 | 0.000352504363 | |
| | 6 | -0.000011338220 | 0.000008013303 | 0.000076203244 | 0.000070781803 | |
| SGAGPC MIMO | 1 | -0.015646950876 | 0.011179302857 | 0.007717266040 | -0.031755065787 | |
| | 2 | 0.048166173657 | 0.018568637990 | 0.007475397130 | -0.015109745265 | |
| | 3 | -0.077931561156 | -0.008027548538 | -0.043265284190 | -0.081861406840 | |
| | 4 | 0.136864325157 | 0.047167142100 | -0.076937996725 | 0.060925191134 | |
| | 5 | 0.033578566118 | 0.180366327134 | -0.046820658093 | 0.054453461387 | |
| | 6 | -0.023744222255 | -0.028829939080 | 0.222578853076 | -0.052799880177 | |
| NGPC SISO | 1 | -0.000062911172 | -0.000066618897 | 0.000000364201 | -0.000044689779 | |
| | 2 | 0.000235349801 | 0.000333023120 | 0.000271621585 | 0.000277580440 | |
| | 3 | 0.000047511986 | 0.000140620309 | 0.000085222268 | 0.000189741537 | |
| | 4 | 0.000017977272 | 0.000025905792 | -0.000000001866 | 0.000168600105 | |
| | 5 | 0.000020311790 | 0.000074696546 | 0.000001377438 | 0.001013281516 | |
| | 6 | -0.000014160598 | -0.000014423312 | 0.000000723020 | 0.000005692609 | |
| NGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |
| ENGPC SISO | 1 | -0.000032511340 | -0.000036185319 | 0.000001289222 | -0.000130102532 | |
| | 2 | 0.000142674241 | 0.000195954365 | 0.000164484690 | 0.000171052723 | |
| | 3 | 0.000020183279 | 0.000073406721 | 0.000044633514 | 0.000224140180 | |
| | 4 | 0.000007119651 | -0.000001411705 | -0.000000853519 | -0.000153060265 | |
| | 5 | -0.000036489065 | 0.000027450521 | -0.000043311816 | -0.000624723361 | |
| | 6 | -0.000007672312 | -0.000007703134 | 0.000001002896 | -0.000485247584 | |
| ENGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |

Örnek 2 için durum 3 için 3. eklem üzerinden karşılaştırma yapılırsa 3. Durum için 3. eklem üzerinden karşılaştırma yapılırsa 0.000044633514 *rd* ile en düşük hata ENGPC SISO algoritmasına aittir. Bunu, NGPC SISO 0.000085222268 *rd* , SGA-GPC SISO -0.001001803285 *rd* , GPC SISO -0.000689948476 *rd* , SGA-GPC MIMO -0.043265284190 *rd* ve GPC MIMO -0.052287083598 *rd* 'lık hata ile takip etmektedir.

Durum 4 için 2. eklem üzerinden karşılaştırma yapılırsa 0.000171052723 *rd* ile en düşük hata ENGPC SISO algoritmasına aittir. Bunu, NGPC SISO 0.000277580440 *rd* , SGA-GPC SISO 0.001907165781 *rd* , GPC SISO 0.002057784329 *rd* , SGA-GPC MIMO -0.015109745265 *rd* ve GPC MIMO -0.040565849134 *rd* 'lık hata ile takip etmektedir.

Diğer durum ve eklemler için de sıra aşağı yukarı bu şekildedir. Açısal hatalar oldukça düşüktür. Yüklü, yük düşmeli ve bozuculu durumlarda bile hassas kontrol sağlanmıştır.

Tablo 4.5. Örnek 2 için eklemlere ait açısal hız hatalarının kareleri toplamı

| Kontrol | Eklem No | Durumlar | | | | Birim |
|---|---|---|---|---|---|---|
| | | D1 | D2 | D3 | D4 | |
| GPC SISO | 1 | 0.007328296118 | 0.007193317957 | 0.007197737623 | 0.005584590836 | $(rd/s)^2$ |
| | 2 | 0.137343101045 | 0.144863591965 | 0.146816788613 | 0.553244981341 | |
| | 3 | 0.015471812785 | 0.063053495000 | 0.063061974025 | 0.010716038227 | |
| | 4 | 0.000002885460 | 0.003763106888 | 0.003763135782 | 0.015912044888 | |
| | 5 | 0.000016185537 | 0.128570216275 | 0.128885400717 | 0.024125675501 | |
| | 6 | 0.000018246039 | 0.000117116393 | 0.000117116393 | 0.015167502160 | |
| GPC MIMO | 1 | 0.026728376564 | 0.329985303270 | 0.610083126203 | 0.923235431034 | |
| | 2 | 0.270303621116 | 0.344303558163 | 0.515795628312 | 1.045878791226 | |
| | 3 | 1.009371588808 | 1.355950172261 | 0.636532127538 | 3.383105619021 | |
| | 4 | 1.126112976278 | 0.297954744514 | 1.320727172267 | 3.416805083744 | |
| | 5 | 0.413610168796 | 0.811787819018 | 0.731686618815 | 1.344738252911 | |
| | 6 | 8.594003520863 | 0.605992768038 | 0.694771212600 | 3.828802325409 | |
| SGAGPC SISO | 1 | 0.003576230920 | 0.000377810183 | 0.007182089746 | 0.014341505841 | |
| | 2 | 0.081210879414 | 0.094394542442 | 0.034583390496 | 0.152534751040 | |
| | 3 | 0.022023500301 | 0.000591316029 | 0.131386434581 | 0.073982672302 | |
| | 4 | 0.000074067741 | 0.000099549231 | 0.015958885158 | 0.015410795791 | |
| | 5 | 0.000030887075 | 0.000114980740 | 0.036917288650 | 0.029766302418 | |
| | 6 | 0.000036670770 | 0.000087707563 | 0.014804881021 | 0.015006349400 | |
| SGAGPC MIMO | 1 | 0.556244355672 | 0.113456099345 | 0.235366409075 | 0.352280023655 | |
| | 2 | 1.338625919968 | 0.322466272728 | 1.373761666836 | 0.652968688461 | |
| | 3 | 1.797949367948 | 0.291317730912 | 1.746890596054 | 1.248016349863 | |
| | 4 | 3.714739166580 | 0.604631764112 | 4.405440304406 | 3.511237118293 | |
| | 5 | 0.699162762904 | 4.385575698019 | 1.876569162093 | 0.765635879702 | |
| | 6 | 1.097102830543 | 0.865131642574 | 37.008786268834 | 3.268241449064 | |
| NGPC SISO | 1 | 0.000148832503 | 0.000045291534 | 0.000250537453 | 0.003581928180 | |
| | 2 | 0.005929116294 | 0.003057034879 | 0.004219436584 | 0.004658019692 | |
| | 3 | 0.002464082912 | 0.000511824163 | 0.000589367323 | 0.006464117708 | |
| | 4 | 0.000025733650 | 0.000017240270 | 0.000000055111 | 0.022120479053 | |
| | 5 | 0.003003010490 | 0.000096730809 | 0.000000278250 | 0.030322959153 | |
| | 6 | 0.000029586198 | 0.000030113521 | 0.000007547425 | 0.036600528123 | |
| NGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |
| ENGPC SISO | 1 | 0.000081762211 | 0.000018601828 | 0.000152587837 | 0.001939094793 | |
| | 2 | 0.002886195323 | 0.001464780877 | 0.002181944744 | 0.002603140732 | |
| | 3 | 0.001232549437 | 0.000246062551 | 0.000313745266 | 0.003438624382 | |
| | 4 | 0.000020350099 | 0.000017592808 | 0.000011678847 | 0.047424692619 | |
| | 5 | 0.001337491302 | 0.000039362591 | 0.000002520674 | 0.059418285536 | |
| | 6 | 0.000115312939 | 0.000114934943 | 0.000103469974 | 0.072501128648 | |
| ENGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |

Durum 2 için 1. eklem üzerinden karşılaştırma yapılırsa 0.000018601828 $(rd/s)^2$ ile en düşük açısal hız farkları kare toplam hatası ENGPC SISO algoritmasına aittir. Bunu sırasıyla, NGPC SISO 0.000045291534 $(rd/s)^2$, SGA-GPC SISO 0.000377810183 $(rd/s)^2$, GPC SISO 0.007193317957 $(rd/s)^2$, SGA-GPC MIMO 0.113456099345 $(rd/s)^2$ ve GPC MIMO 0.329985303270 $(rd/s)^2$ olarak takip etmektedir.

Tablo 4.6. Örnek 2 için uç nokta koordinat hataları

| Kontrol | Uç Nokta Hatası | Durumlar | | | | Birim |
|---|---|---|---|---|---|---|
| | | D1 | D2 | D3 | D4 | |
| **GPC SISO** | $\Delta x$ | 0.478185539726 | 0.582976951063 | 0.459776944441 | 1.103102027803 | *mm* |
| | $\Delta y$ | 0.125493484558 | 0.181011397439 | 0.180444507298 | 0.122202277500 | |
| | $\Delta z$ | -0.630039253482 | -0.825629727242 | -0.683738809180 | -1.345009642740 | |
| | Toplam $\Delta$ | 0.800849228006 | 1.026787075288 | 0.843477218443 | 1.743794259491 | |
| **GPC MIMO** | $\Delta x$ | -33.43320380721 | -54.896422312566 | -0.583668329680 | -64.83327227976 | |
| | $\Delta y$ | 4.549689153504 | -26.529065523984 | -49.629565205785 | 2.118757579223 | |
| | $\Delta z$ | 27.796168227735 | 48.684171292924 | 2.110786416968 | 53.46384165358 | |
| | Toplam $\Delta$ | 43.716195584159 | 78.022798173189 | 49.677860567182 | 84.06083923291 | |
| **SGAGPC SISO** | $\Delta x$ | 0.284204693927 | 0.538352737653 | 0.133222897563 | 0.802216491802 | |
| | $\Delta y$ | 0.028613914181 | -0.011341985319 | -0.458011866787 | 0.222960063956 | |
| | $\Delta z$ | -0.389342397468 | -0.628477818230 | -0.182961059595 | -1.085269681260 | |
| | Toplam $\Delta$ | 0.482885666179 | 0.827609013229 | 0.510879594308 | 1.367871620767 | |
| **SGAGPC MIMO** | $\Delta x$ | 2.572033193737 | 10.024056135535 | -14.82304652137 | -35.00824788083 | |
| | $\Delta y$ | -9.020507772662 | 4.352664323983 | 5.322322167092 | -20.76847707073 | |
| | $\Delta z$ | -11.46803223352 | -19.122124280313 | 10.450158373159 | 31.15197620051 | |
| | Toplam $\Delta$ | 14.815555289476 | 22.024600453070 | 18.901207142645 | 51.25770850048 | |
| **NGPC SISO** | $\Delta x$ | 0.166113793172 | 0.260442804271 | 0.192261628553 | 0.265370544826 | |
| | $\Delta y$ | -0.043246935818 | -0.047093145076 | 0.000212564315 | -0.054514871522 | |
| | $\Delta z$ | -0.176596894354 | -0.274662008988 | -0.212124866211 | -0.294328093405 | |
| | Toplam $\Delta$ | 0.246273329525 | 0.381428155478 | 0.286289255583 | 0.400028028825 | |
| **NGPC MIMO** | $\Delta x$ | Kontrol Başarısız | | | | |
| | $\Delta y$ | | | | | |
| | $\Delta z$ | | | | | |
| | Toplam $\Delta$ | | | | | |
| **ENGPC SISO** | $\Delta x$ | 0.095559347463 | 0.148847910661 | 0.112665931401 | 0.187773813017 | |
| | $\Delta y$ | -0.021141398659 | -0.025486792764 | 0.002011794565 | -0.074980780749 | |
| | $\Delta z$ | -0.102204452605 | -0.157878041187 | -0.124333523220 | -0.157793553068 | |
| | Toplam $\Delta$ | 0.141507235705 | 0.218473689497 | 0.167798940438 | 0.256475978849 | |
| **ENGPC MIMO** | $\Delta x$ | Kontrol Başarısız | | | | |
| | $\Delta y$ | | | | | |
| | $\Delta z$ | | | | | |
| | Toplam $\Delta$ | | | | | |

1. Durum için x-ekseni üzerinden karşılaştırma yapılırsa 0.095559347463 *mm* ile en düşük eksensel hata ENGPC SISO algoritmasına aittir. Bunu sırası ile NGPC SISO 0.166113793172 *mm*, SGA-GPC SISO 0.284204693927 *mm*, GPC SISO

0.478185539726 *mm*, SGA-GPC MIMO 2.572033193737 *mm* ve GPC MIMO -33.43320380721 *mm* olarak takip etmektedir.

3. Durum için toplam uç nokta koordinat hatası üzerinden karşılaştırma yapılırsa 0.167798940438 *mm* ile en düşük eksensel hata ENGPC SISO algoritmasına aittir. Bunu sırası ile NGPC SISO 0.286289255583 *mm*, SGA-GPC SISO 0.510879594308 *mm*, GPC SISO 0.843477218443 *mm*, SGA-GPC MIMO 18.901207142645 *mm* ve GPC MIMO 49.677860567182 *mm* olarak takip etmektedir.

Diğer durum ve eksenler için de sıra aşağı yukarı bu şekildedir. Açısal hatalar az olduğundan doğal olarak eksensel hatalar da azdır.

Örnek 3 ‘e ait sonuçlar Tablo 4.7, 4.8 ve 4.9‘da verilmiştir.

Tablo 4.7. Örnek 3 için eklemlere ait açısal bitiş konum hataları

| Kontrol | Eklem No | Durumlar | | | | Birim |
|---|---|---|---|---|---|---|
| | | D1 | D2 | D3 | D4 | |
| GPC SISO | 1 | -0.000012614111 | 0.004334412105 | 0.004333618050 | -0.001635667014 | *rd* |
| | 2 | 0.001210856300 | 0.000579226261 | 0.000293530856 | 0.003620323227 | |
| | 3 | 0.000087828627 | 0.000335055582 | 0.000284744723 | 0.001156146029 | |
| | 4 | 0.000000812521 | -0.000266010467 | -0.000271622330 | 0.000025148474 | |
| | 5 | 0.000006459565 | -0.000150757820 | -0.000135590740 | -0.011410268914 | |
| | 6 | 0.000001696267 | -0.000006174741 | -0.000006174741 | -0.000156154685 | |
| GPC MIMO | 1 | 0.051619749059 | -0.108316844264 | -0.090007402215 | 0.192012653694 | |
| | 2 | 0.012260276406 | 0.043247548308 | 0.025174786792 | 0.018272144966 | |
| | 3 | 0.013522067677 | 0.019603346721 | -0.050418573484 | -0.113774219780 | |
| | 4 | 0.088078551017 | -0.218074215137 | -0.182807146693 | -0.151250637899 | |
| | 5 | 0.163126775196 | -0.230274797676 | -0.256290605606 | -0.338944439176 | |
| | 6 | 0.169740264964 | -0.036458386234 | -0.054755909898 | -0.137826888898 | |
| SGAGPC SISO | 1 | 0.000010645732 | 0.004033861057 | 0.004033533321 | -0.001255026122 | |
| | 2 | 0.001029370190 | 0.000781614146 | 0.000501071352 | 0.002285623753 | |
| | 3 | 0.000291332680 | 0.000469797180 | 0.000441907922 | 0.003063535371 | |
| | 4 | 0.000002093790 | -0.000135401852 | -0.000140275384 | 0.000292256938 | |
| | 5 | 0.000003911187 | -0.000160019367 | -0.000144768055 | 0.000352421894 | |
| | 6 | 0.000001399856 | 0.000016533878 | 0.000016533878 | -0.000595372607 | |
| SGAGPC MIMO | 1 | 0.004947072479 | -0.026709948897 | -0.030970398428 | 0.009679386577 | |
| | 2 | -0.006723505649 | 0.023430065154 | 0.035365582746 | -0.006527585143 | |
| | 3 | -0.025447039978 | -0.070213441373 | -0.062164104650 | -0.077872113373 | |
| | 4 | 0.090469257354 | -0.151684875199 | -0.152119746368 | -0.144402169158 | |
| | 5 | -0.047787864972 | -0.258049278377 | -0.228790819162 | -0.207881027103 | |
| | 6 | 0.124765475161 | -0.133159500815 | -0.168579811723 | -0.129545421604 | |
| NGPC SISO | 1 | 0.000002050232 | -0.000062320570 | -0.000064451814 | -0.000385719566 | |
| | 2 | 0.000310417265 | 0.000652795019 | 0.000357779993 | 0.000720190679 | |
| | 3 | 0.000096346701 | 0.000338364236 | 0.000059487508 | 0.000165462851 | |
| | 4 | -0.000000271188 | -0.000055221055 | -0.000015761555 | -0.001210699671 | |
| | 5 | 0.000002411837 | 0.000169976273 | -0.000014009086 | -0.001820158657 | |
| | 6 | 0.000000427201 | -0.000013765295 | -0.000013765295 | -0.000451913945 | |
| NGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |
| ENGPC SISO | 1 | 0.000037797490 | -0.000031870520 | -0.000033685380 | -0.000261977889 | |
| | 2 | 0.000158788273 | 0.000367196686 | 0.000181695903 | 0.000268626788 | |
| | 3 | 0.000056661339 | 0.000197365651 | 0.000045242887 | 0.000145646961 | |
| | 4 | 0.000049495941 | -0.000031140790 | 0.000057782169 | -0.002093243646 | |
| | 5 | 0.000002738606 | 0.000090037060 | -0.000009603087 | -0.001511656513 | |
| | 6 | 0.000002120706 | -0.000006259087 | -0.000006259087 | -0.000512574238 | |
| ENGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |

Durum 2 için 1. eklem üzerinden karşılaştırma yapılırsa -0.000031870520 *rd* ile en düşük hata ENGPC SISO algoritmasına aittir. Bunu, NGPC SISO -0.000062320570 *rd* , SGA-GPC SISO 0.004033861057 *rd* , GPC SISO 0.004334412105 *rd* , SGA-GPC MIMO -0.026709948897 *rd* ve GPC MIMO -0.108316844264 *rd* 'lık hata ile takip etmektedir.

Durum 4 için 5. eklem üzerinden karşılaştırma yapılırsa -0.001511656513 *rd* ile en düşük hata ENGPC SISO algoritmasına aittir. Bunu, NGPC SISO -0.001820158657 *rd* , SGA-GPC SISO 0.000352421894 *rd* , GPC SISO -0.011410268914 *rd* , SGA-GPC MIMO -0.207881027103 *rd* ve GPC MIMO -0.338944439176 *rd* 'lık hata ile takip etmektedir.

Diğer durum ve eklemler için de sıra aşağı yukarı bu şekildedir. Açısal hatalar oldukça düşüktür. Yüklü, yük düşmeli ve bozuculu durumlarda bile hassas kontrol sağlanmıştır.

Tablo 4.8. Örnek 3 için açısal hız farklarının kare toplam hataları

| Kontrol | Eklem No | Durumlar | | | | Birim |
|---|---|---|---|---|---|---|
| | | D1 | D2 | D3 | D4 | |
| GPC SISO | 1 | 0.000080523310 | 0.741800876257 | 0.741801942161 | 0.036324084999 | $(rd/s)^2$ |
| | 2 | 0.238610233874 | 0.052382051409 | 0.069760411022 | 0.201836183856 | |
| | 3 | 0.001022026321 | 0.008405553689 | 0.008836585021 | 0.051108481648 | |
| | 4 | 0.000009237573 | 0.006216183807 | 0.006231059663 | 0.340475395492 | |
| | 5 | 0.000020324685 | 0.002340938402 | 0.002512945299 | 2.630564398749 | |
| | 6 | 0.000006172727 | 0.000099679682 | 0.000099679682 | 0.160614250682 | |
| GPC MIMO | 1 | 0.396640292713 | 1.581227588755 | 1.487572251550 | 7.201984640814 | |
| | 2 | 0.416205996293 | 1.275156727465 | 1.038713589255 | 8.229471030814 | |
| | 3 | 0.176803566019 | 0.127002759998 | 0.590185837852 | 2.178132277342 | |
| | 4 | 0.961130089665 | 6.514539922768 | 4.112144737212 | 2.992954490912 | |
| | 5 | 3.339558279704 | 6.893781975168 | 8.458089320531 | 14.98621459809 | |
| | 6 | 3.571201325152 | 0.342035732874 | 0.820127383692 | 3.211405435930 | |
| SGAGPC SISO | 1 | 0.000187623186 | 0.657496496823 | 0.657497459106 | 0.037963282401 | |
| | 2 | 0.185923140305 | 0.097255896797 | 0.115574128580 | 0.205569610019 | |
| | 3 | 0.012233354163 | 0.016940184866 | 0.017167136364 | 0.245461591583 | |
| | 4 | 0.000008126319 | 0.002330280128 | 0.002346963870 | 0.130368449024 | |
| | 5 | 0.000015771779 | 0.002293528549 | 0.002459841562 | 0.158345454438 | |
| | 6 | 0.000005776619 | 0.000134025306 | 0.000134025306 | 0.129768256534 | |
| SGAGPC MIMO | 1 | 0.179266662306 | 0.343810698694 | 0.350640766676 | 0.159284861355 | |
| | 2 | 1.474263001933 | 1.752483872049 | 1.958454716822 | 1.001825282074 | |
| | 3 | 1.106703210409 | 0.672991379693 | 1.425881116943 | 1.202148901734 | |
| | 4 | 1.464060065355 | 3.349141705419 | 2.642495775923 | 2.631980470922 | |
| | 5 | 0.510208460486 | 8.384591947732 | 6.803885850837 | 6.450237717038 | |
| | 6 | 2.169076856444 | 2.585057675399 | 3.973871092135 | 3.762738660540 | |
| NGPC SISO | 1 | 0.000003663940 | 0.000138028801 | 0.000143861742 | 0.002762690659 | |
| | 2 | 0.013974327403 | 0.034911802678 | 0.049014383311 | 0.054722512077 | |
| | 3 | 0.000978503939 | 0.011007338074 | 0.018807502478 | 0.024942823275 | |
| | 4 | 0.000000852976 | 0.000347894997 | 0.000434245009 | 0.060623029710 | |
| | 5 | 0.000004956530 | 0.000479073120 | 0.001330881290 | 0.106808595563 | |
| | 6 | 0.000003727259 | 0.000021428762 | 0.000021428762 | 0.084829030477 | |
| NGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |
| ENGPC SISO | 1 | 0.000008471651 | 0.000058159717 | 0.000061309023 | 0.001242126634 | |
| | 2 | 0.007147724289 | 0.016905516966 | 0.024579847212 | 0.028089624758 | |
| | 3 | 0.000445903489 | 0.005084469587 | 0.008680543587 | 0.012352852894 | |
| | 4 | 0.000014750007 | 0.000175467419 | 0.000206310272 | 0.047683168547 | |
| | 5 | 0.000075366488 | 0.000325199426 | 0.000691562237 | 0.065195298267 | |
| | 6 | 0.000062816416 | 0.000071307616 | 0.000071307616 | 0.057819123805 | |
| ENGPC MIMO | 1 | Kontrol Başarısız | | | | |
| | 2 | | | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 | | | | | |
| | 6 | | | | | |

Durum 3 için 5. eklem üzerinden karşılaştırma yapılırsa 0.000691562237 $(rd/s)^2$ ile en düşük açısal hız farkları kare toplam hatası ENGPC SISO algoritmasına aittir. Bunu sırasıyla, NGPC SISO 0.001330881290 $(rd/s)^2$, SGA-GPC SISO 0.002459841562 $(rd/s)^2$, GPC SISO 0.002512945299 $(rd/s)^2$, SGA-GPC MIMO 6.803885850837 $(rd/s)^2$ ve GPC MIMO 8.458089320531 $(rd/s)^2$ olarak takip etmektedir.

Durum 4 için 4. eklem üzerinden karşılaştırma yapılırsa 0.047683168547 $(rd/s)^2$ ile en düşük açısal hız farkları kare toplam hatası ENGPC SISO algoritmasına aittir. Bunu sırasıyla, NGPC SISO 0.060623029710 $(rd/s)^2$, SGA-GPC SISO 0.130368449024 $(rd/s)^2$, GPC SISO 0.340475395492 $(rd/s)^2$, SGA-GPC MIMO 2.631980470922 $(rd/s)^2$ ve GPC MIMO 2.992954490912 $(rd/s)^2$ olarak takip etmektedir.

Diğer durum ve eklemler için de sıra aşağı yukarı bu şekildedir. Açısal hız farklarının kare toplam hataları oldukça düşüktür.

Tablo 4.9. Örnek 3 için uç nokta koordinat hataları

| Kontrol | Uç Nokta Hatası | Durumlar | | | | Birim |
|---|---|---|---|---|---|---|
| | | D1 | D2 | D3 | D4 | |
| **GPC SISO** | $\Delta x$ | 0.130183195851 | -3.482655737708 | -3.512292486310 | 1.979763881123 | *mm* |
| | $\Delta y$ | 0.327754290839 | 0.748591794020 | 0.663603256927 | 1.302007885416 | |
| | $\Delta z$ | -1.024400987371 | -0.600545557633 | -0.348202753372 | -2.894950575157 | |
| | Toplam $\Delta$ | 1.083405705437 | 3.612469463989 | 3.591352523695 | 3.741046429983 | |
| **GPC MIMO** | $\Delta x$ | -48.06747607400 | 109.61064061711 | 84.238414472217 | -152.7874803761 | |
| | $\Delta y$ | 6.043320063192 | -2.224912128951 | -16.421196416673 | -8.731321042570 | |
| | $\Delta z$ | -20.56574555969 | -34.843002763129 | 9.709476263131 | 44.06796084834 | |
| | Toplam $\Delta$ | 52.630351166766 | 115.03685327766 | 86.371523628174 | 159.2552520332 | |
| **SGAGPC SISO** | $\Delta x$ | 0.110420359061 | -3.211420982221 | -3.239226880356 | 1.474280859803 | |
| | $\Delta y$ | 0.329213155323 | 0.795629052255 | 0.717052002560 | 1.094755409374 | |
| | $\Delta z$ | -0.954727301694 | -0.815930494121 | -0.576455422153 | -3.070089823062 | |
| | Toplam $\Delta$ | 1.015912485371 | 3.407637434510 | 3.367351364281 | 3.577351671503 | |
| **SGAGPC MIMO** | $\Delta x$ | -19.14968005159 | 29.226450812141 | 34.387321300240 | -5.187578724300 | |
| | $\Delta y$ | 3.185537571263 | -9.822920410994 | -5.820277757822 | -16.05556399473 | |
| | $\Delta z$ | 10.643043172466 | 19.082772549696 | 4.682854499980 | 43.58530654454 | |
| | Toplam $\Delta$ | 22.138930951319 | 36.260548819050 | 35.189382285742 | 46.73725553336 | |
| **NGPC SISO** | $\Delta x$ | 0.035004614973 | 0.139360484187 | 0.092417280198 | 0.491482410209 | |
| | $\Delta y$ | 0.101082539424 | 0.229132671150 | 0.097280185839 | 0.217150523912 | |
| | $\Delta z$ | -0.291219808996 | -0.673330265995 | -0.315151959076 | -0.588714006531 | |
| | Toplam $\Delta$ | 0.310245032189 | 0.724773600958 | 0.342527583628 | 0.797053003925 | |
| **NGPC MIMO** | $\Delta x$ | Kontrol Başarısız | | | | |
| | $\Delta y$ | | | | | |
| | $\Delta z$ | | | | | |
| | Toplam $\Delta$ | | | | | |
| **ENGPC SISO** | $\Delta x$ | -0.014348189546 | 0.076449201334 | 0.045674080983 | 0.384514981960 | |
| | $\Delta y$ | 0.058169256592 | 0.131060838260 | 0.053173668431 | 0.097026963494 | |
| | $\Delta z$ | -0.151153340515 | -0.381194690937 | -0.164778621753 | -0.238711566354 | |
| | Toplam $\Delta$ | 0.162594173649 | 0.410281386501 | 0.179068575900 | 0.462870624373 | |
| **ENGPC MIMO** | $\Delta x$ | Kontrol Başarısız | | | | |
| | $\Delta y$ | | | | | |
| | $\Delta z$ | | | | | |
| | Toplam $\Delta$ | | | | | |

Durum 3 için X ekseni üzerinden karşılaştırma yapılırsa ile en düşük eksensel hata 0.045674080983 *mm* ile ENGPC SISO algoritmasına aittir. Bunu sırası ile NGPC SISO 0.092417280198 *mm*, SGA-GPC SISO -3.239226880356 *mm*, GPC SISO -

3.512292486310 $mm$, SGA-GPC MIMO 34.387321300240 $mm$ ve GPC MIMO 84.238414472217 $mm$ olarak takip etmektedir.

Durum 1 için uç nokta toplam koordinat hataları üzerinden karşılaştırma yapılırsa 0.162594173649 $mm$ ile en düşük eksensel hata ENGPC SISO algoritmasına aittir. Bunu sırası ile NGPC SISO 0.310245032189 $mm$, SGA-GPC SISO 1.015912485371 $mm$, GPC SISO 1.083405705437 $mm$, SGA-GPC MIMO 22.138930951319 $mm$ ve GPC MIMO 52.630351166766 $mm$ olarak takip etmektedir. Diğer durum ve eksenler için de sıra aşağı yukarı bu şekildedir. Açısal hatalar az olduğundan doğal olarak eksensel hatalar da azdır.

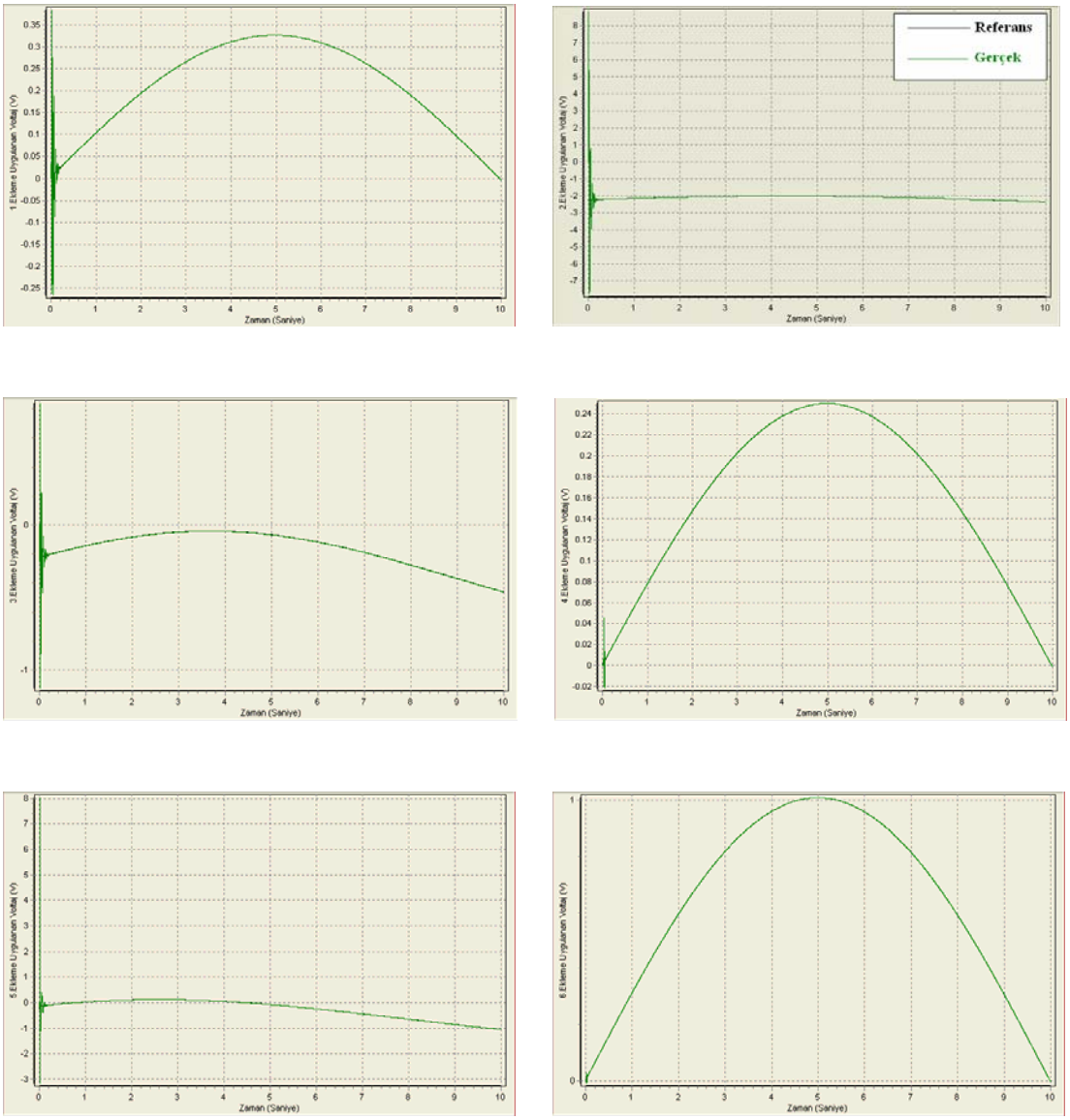
Kullanılan öngörülü kontrol algoritmaları altı eklemli robot kolunu en rahat yüksüz ve sürtünmesiz durumlarda kontrol etmektedir. Robot kolunun 5 $Kg$'lık bir yük taşıdığı ve taşınan yükün taşıma esnasında düştüğü durumlarda da algoritmalar oldukça başarılıdır. Her kontrol adımında, eklemlere uygulanacak her bir torka $-0.5\ Nm$ ile $+0.5\ Nm$ arasında rasgele bozucu ilave edilmesi ile oluşan bozuculu durumlarda algoritmaların genel olarak başarılı olduğu gözükmektedir. Özellikle SISO yapıya sahip algoritmalar yük düşmesi ve bozuculu durumlarda yeni duruma kendilerini çok çabuk adapte ederek küçük konum ve uç nokta hataları ile hareketlerini tamamlamaktadırlar. MIMO yapılara sahip algoritmalar ise bu durumlarda nispeten daha büyük konum ve uç nokta koordinat hataları ile hareketlerini tamamlamışlardır.

1. eklem diğer eklemleri taşısa da hareket yönü yere paralel ve ikinci eklemin hareketine dik olduğundan, diğer eklemlerden ve hareketlerinden en az düzeyde etkilenmektedir. Ayrıca yük taşıma, taşınan yükün düşmesi ve bozucu ilavesi durumlarından da diğer eklemlere kıyasla en az etkilenmektedir. Bu sebeplerden dolayı, tablodaki sonuçlardan da anlaşıldığı üzere, 1. eklemin kontrolü diğer iki eklemin kontrolünden daha kolay olmaktadır.

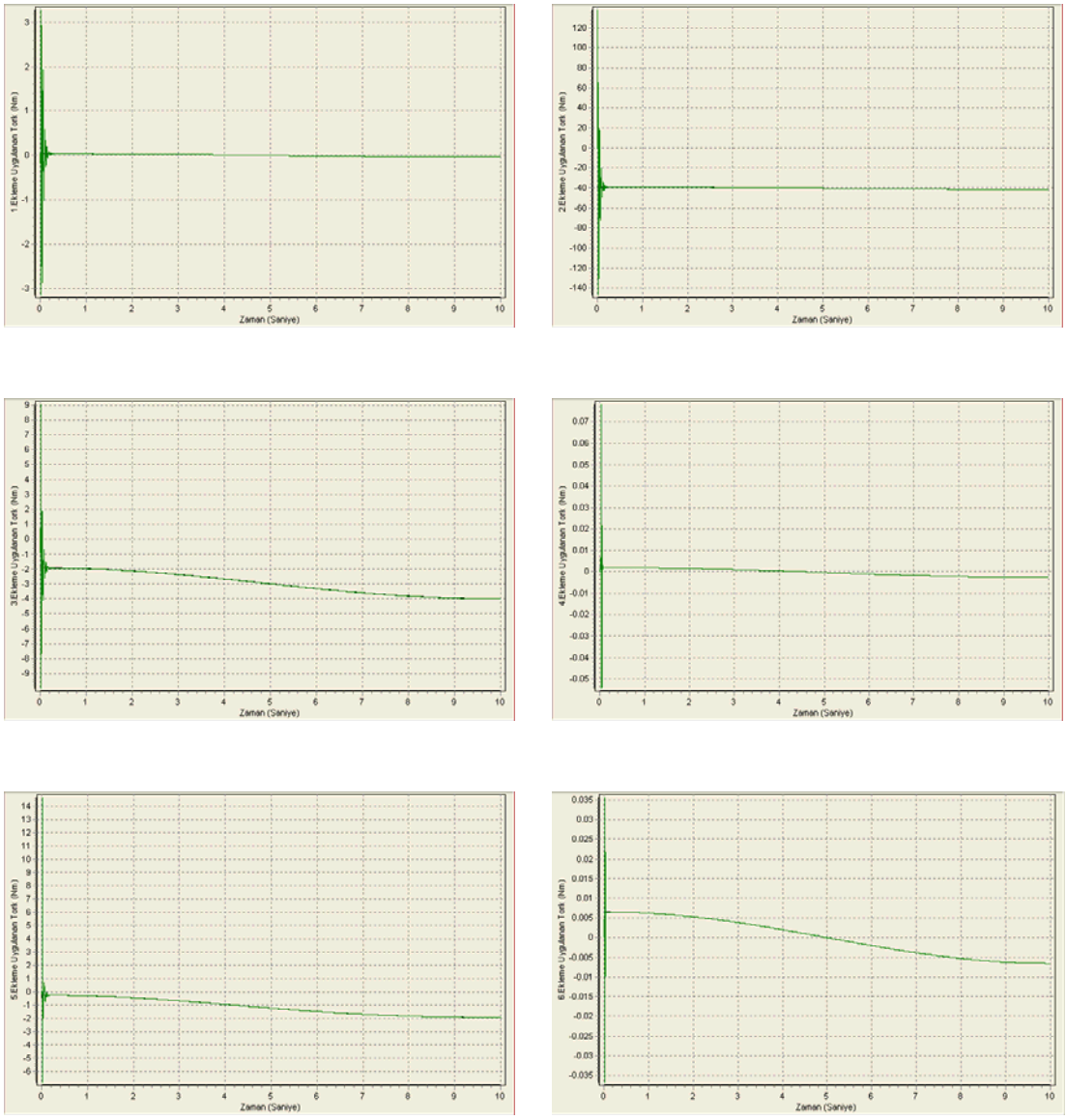
2 ve 3. eklemlerin kontrolü diğer eklemlere göre çok daha zordur. Özellikle 2. eklem kendinden sonraki tüm eklemlerden ve hareketlerinden etkilenmektedir. Yük taşıma, yükün düşmesi ve bozuculu durumlarda kontrolü oldukça zordur.

Uç eklemler olan 4, 5 ve 6. eklemler bilek vazifesi görmektedirler ve diğer eklemlere göre daha kolay kontrol edilmektedirler.

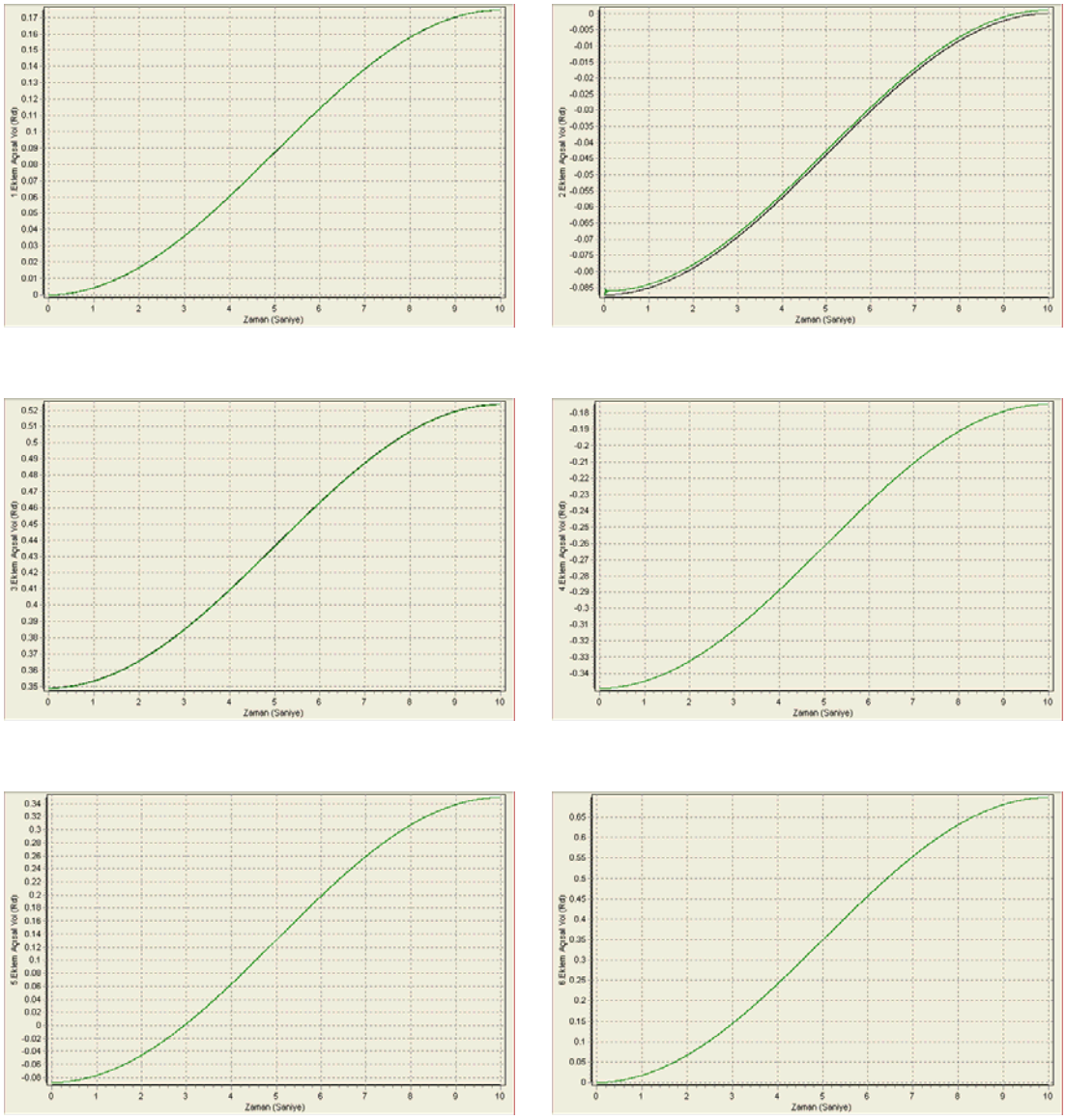
Yukarıda tablosal olarak verilen simülasyon çalışması sonuçları, yer darlığı sebebiyle sadece bir kısmı grafiksel olarak aşağıda tekrardan Şekil 4.21'den Şekil 4.134'e kadar verilmiştir. Bunlar; Örnek 1 'e ait tüm durumlar için algoritmalar ile yapılan altı eklemin simülasyon grafiksel sonuçlarıdır.



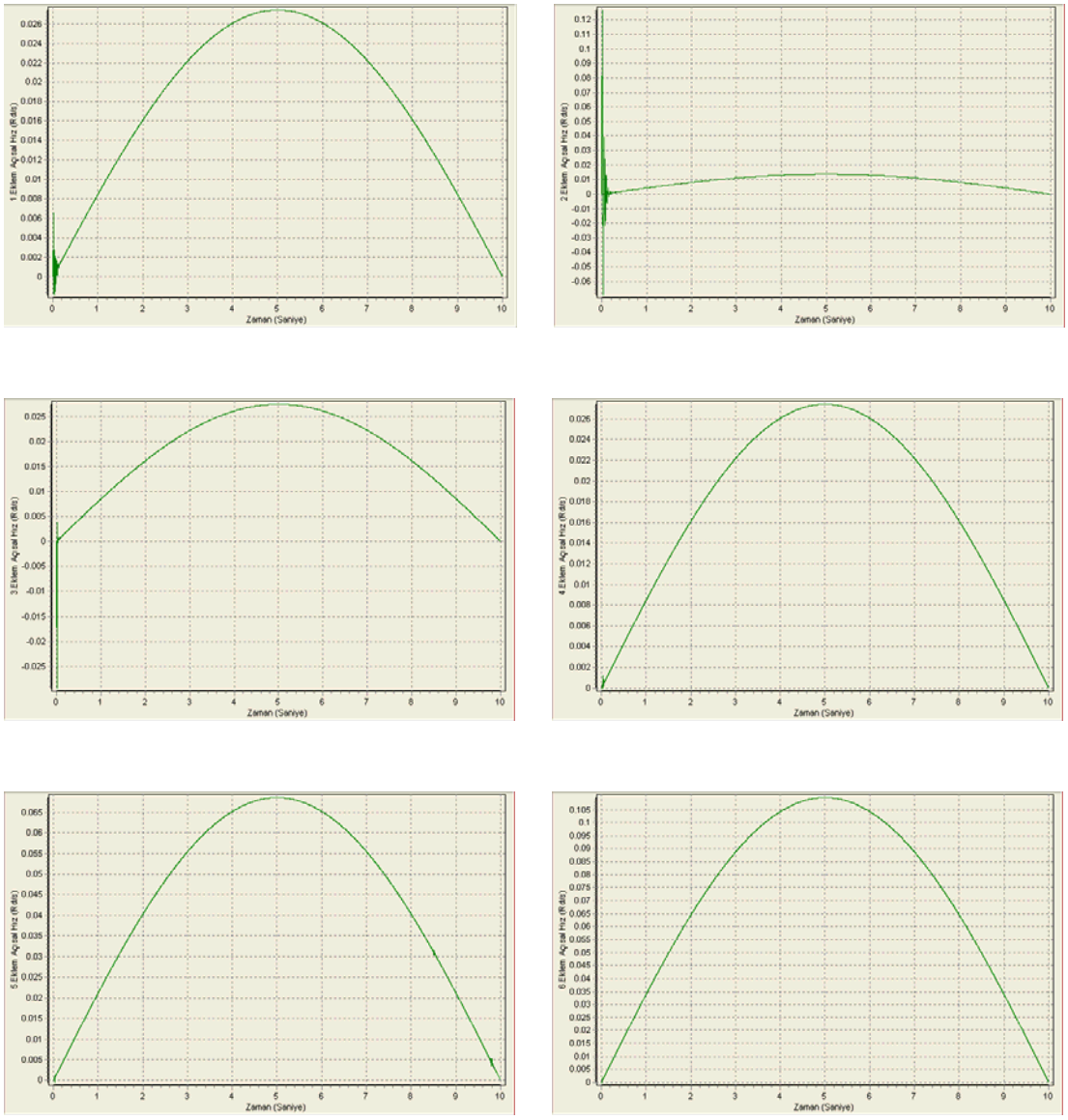

Şekil 4.21. Eklemlere uygulanan gerilim grafikleri (GPC SISO, Örnek 1, Durum 1)




Şekil 4.22. Eklemlere uygulanan tork grafikleri (GPC SISO, Örnek 1, Durum 1)




Şekil 4.23. Eklemlerin takip ettiği açısal yol grafikleri (GPC SISO, Örnek 1, Durum 1)



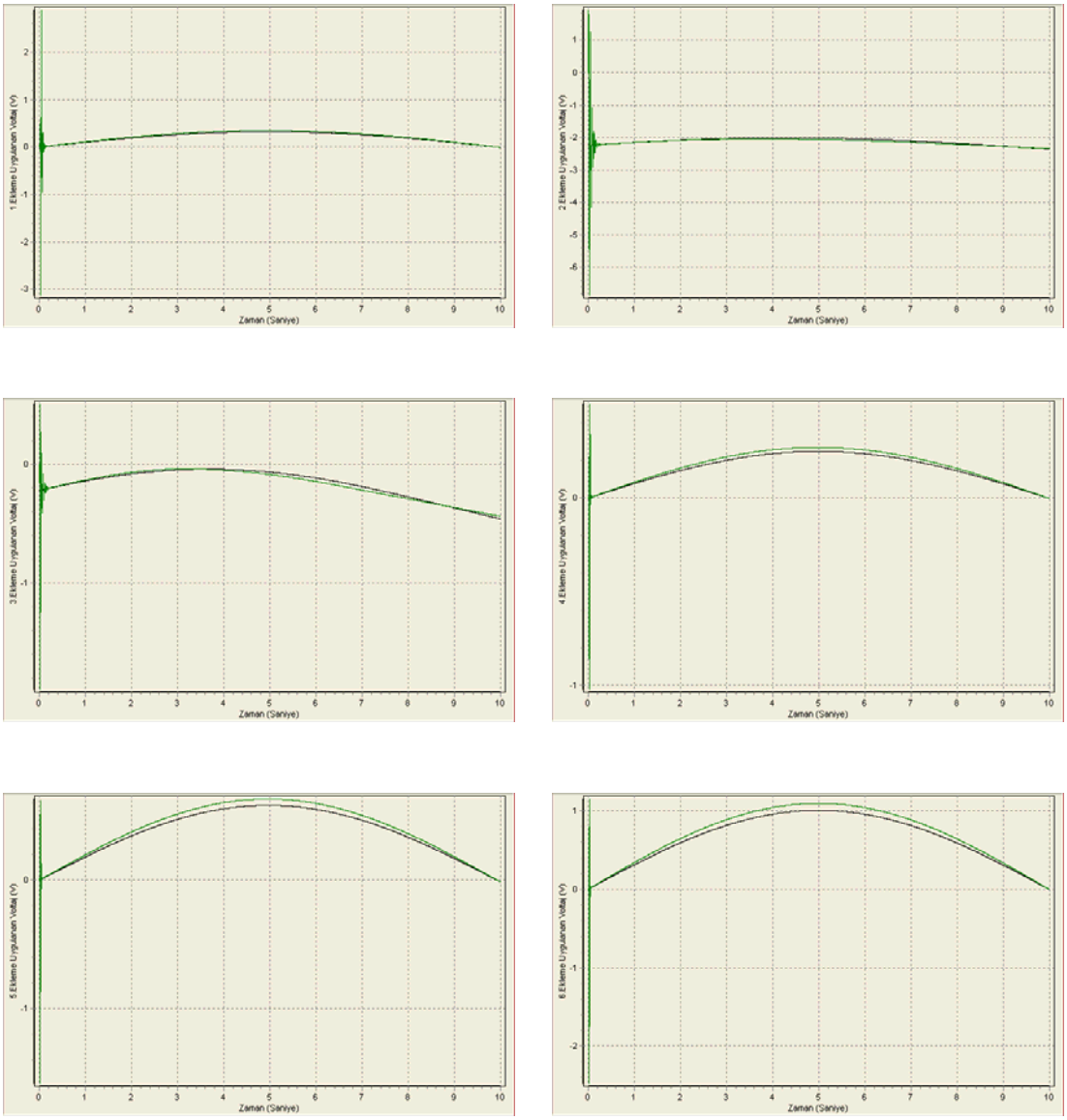

Şekil 4.24. Eklemlerin açısal hız grafikleri (GPC SISO, Örnek 1, Durum 1)


Şekil 4.25. Eklemlerin açısal hız hata grafikleri (GPC SISO, Örnek 1, Durum 1)

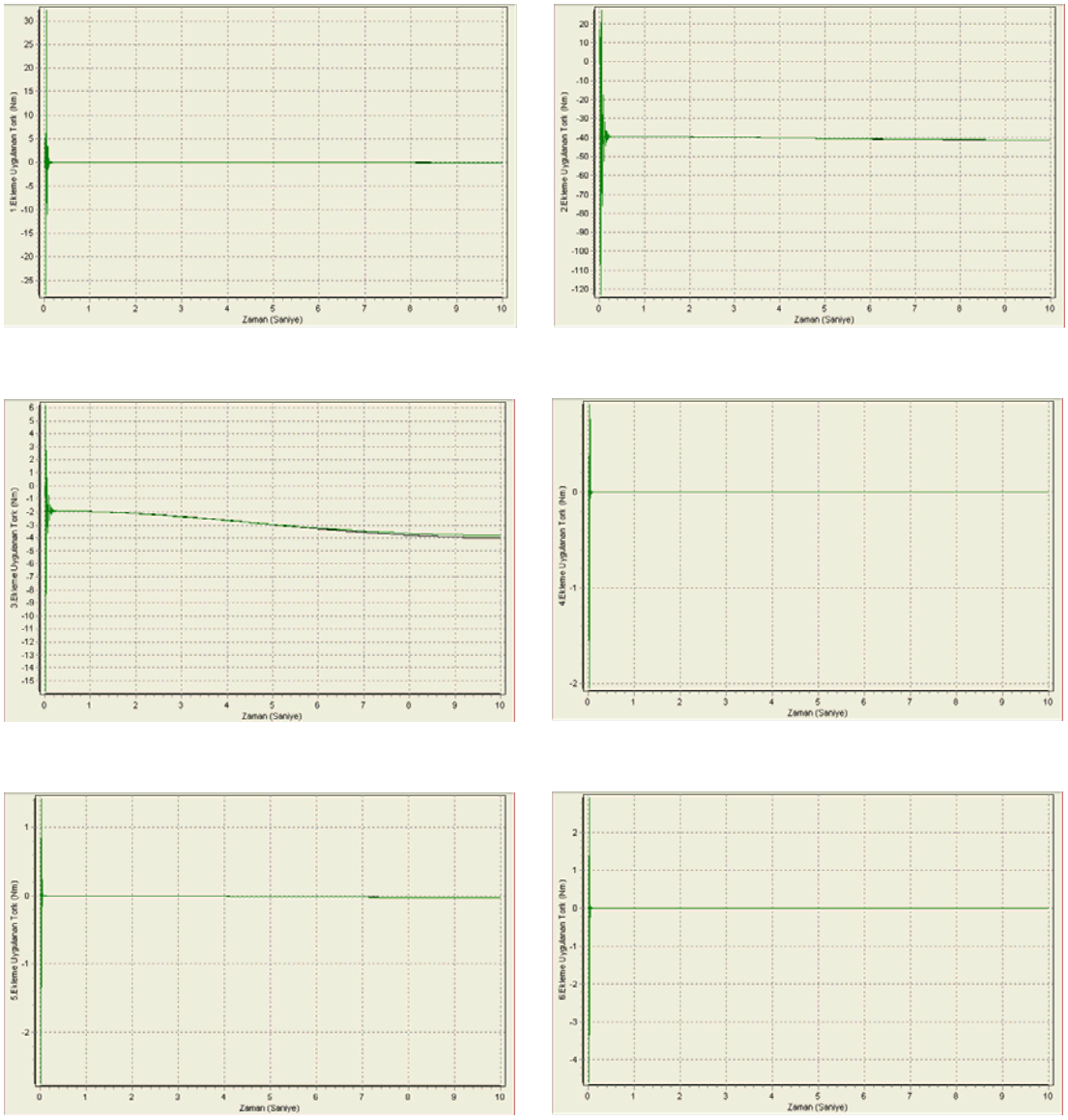
1. Örnek uygulama Durum 1 ( Yük, sürtünme yok) için GPC SISO algoritması ile yapılan simülasyon sonuçlarının grafikleri kısaca şu şekilde yorumlanabilir.

Tork, açısal yol ve açısal hız grafikleri gayet düzgündür. Eklemlere uygulanan gerilim ile tork eğrileri paralellik göstermektedirler. Eklemler referans yörüngelerini çok yakın takip ettikleri için açısal yol grafiklerinde referans yörünge ile gerçeklenen yörünge eğrileri üst üste çakışmaktadırlar. Açısal hız hataları oldukça düşüktür ve grafiklerine bakıldığında 0 ekseni üzerinde bir takip sağlanmıştır. Sadece hareket başlangıcında hareketsiz durumdaki robot kolunun eylemsizliğinden dolayı açısal hız

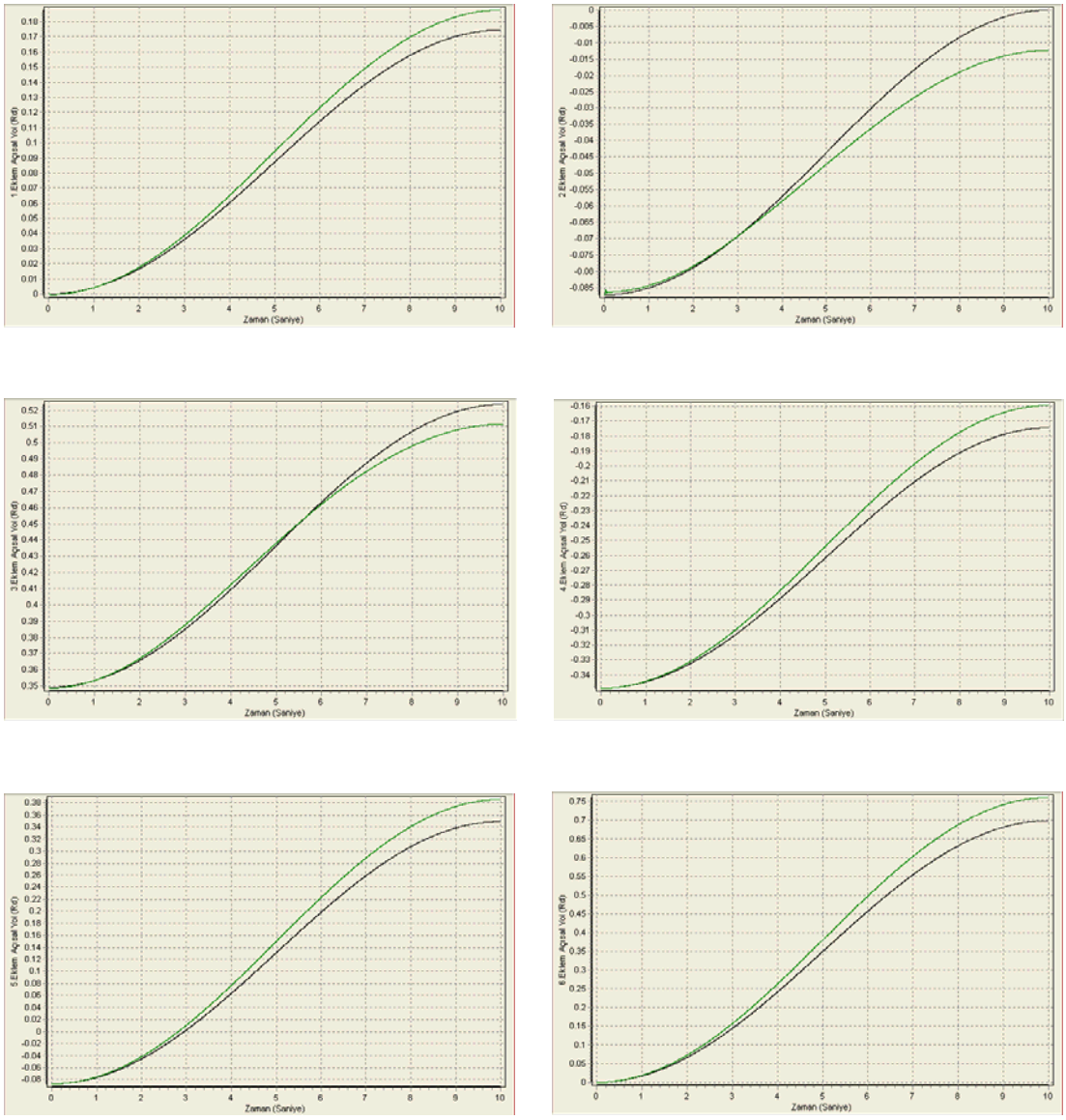
hataları oluşmuştur ve farklı genlikteki salınımlardan sonra referans yörüngeler yakalanmış ve açısal hızlar sıfıra yakınsamıştır.



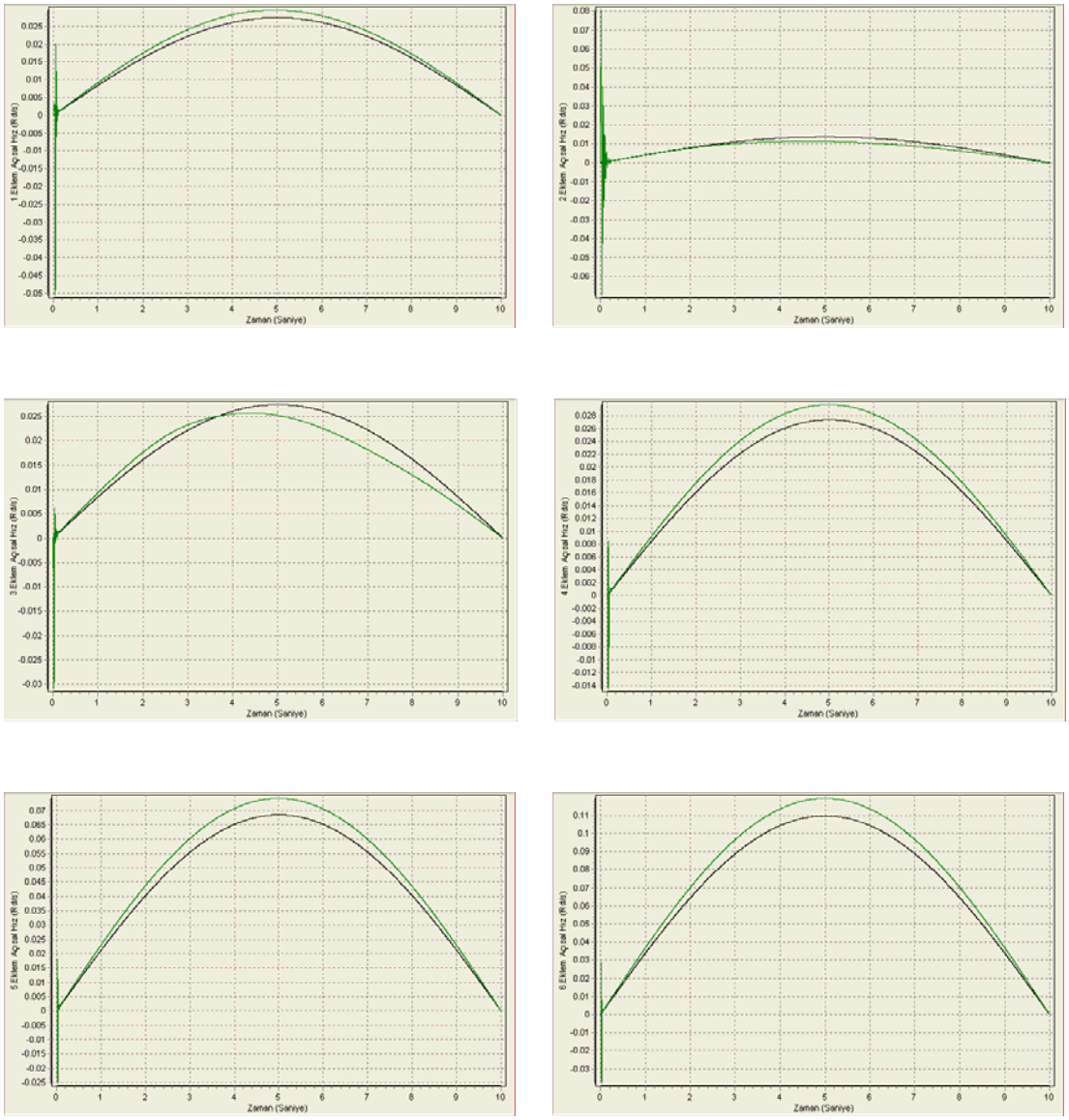

Şekil 4.26. Eklemlere uygulanan gerilim grafikleri (GPC MIMO, Örnek 1, Durum 1)



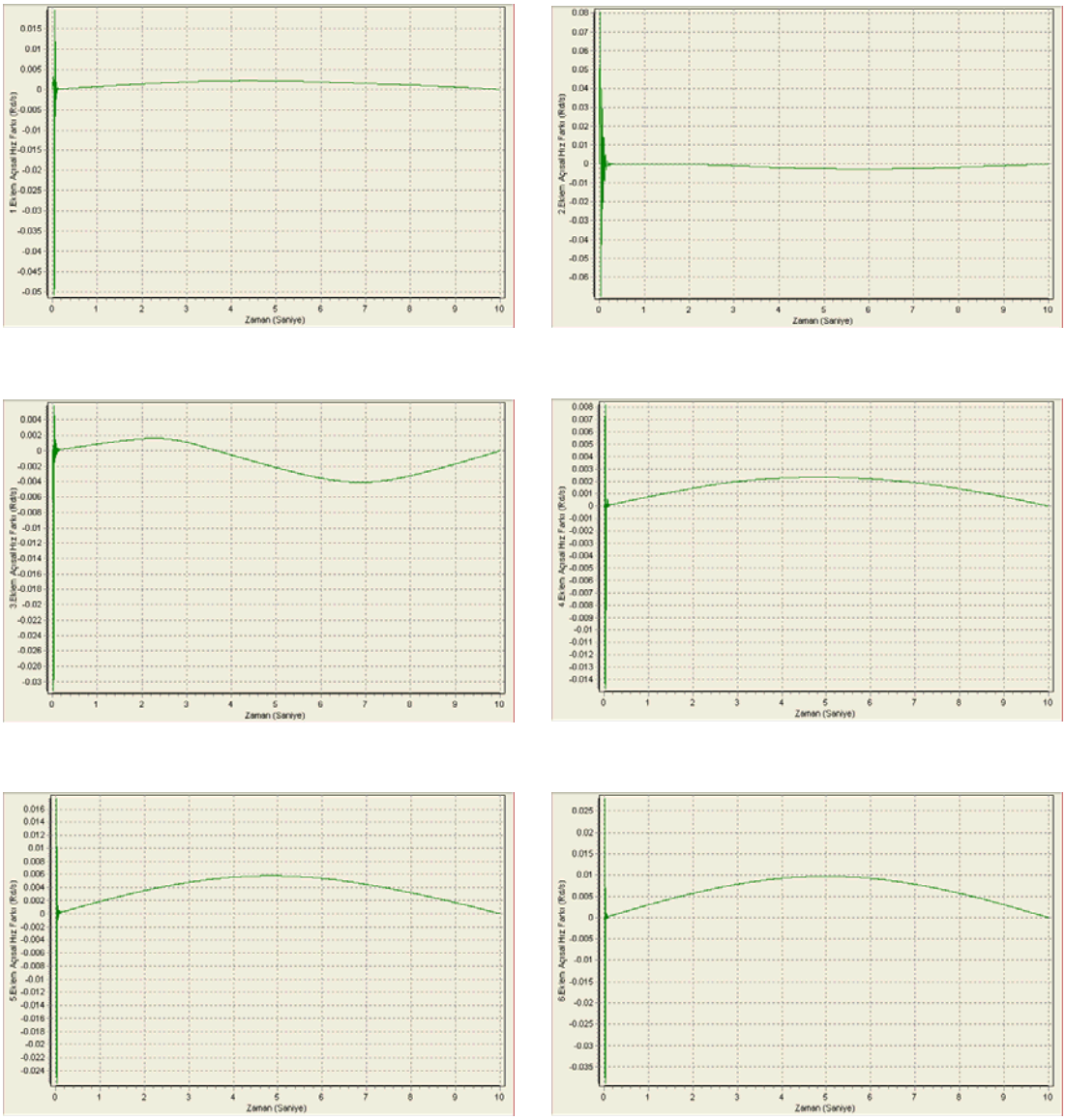

Şekil 4.27. Eklemlere uygulanan tork grafikleri (GPC MIMO, Örnek 1, Durum 1)




Şekil 4.28. Eklemlerin takip ettiği açısal yol grafikleri (GPC MIMO, Örnek 1, Durum 1)

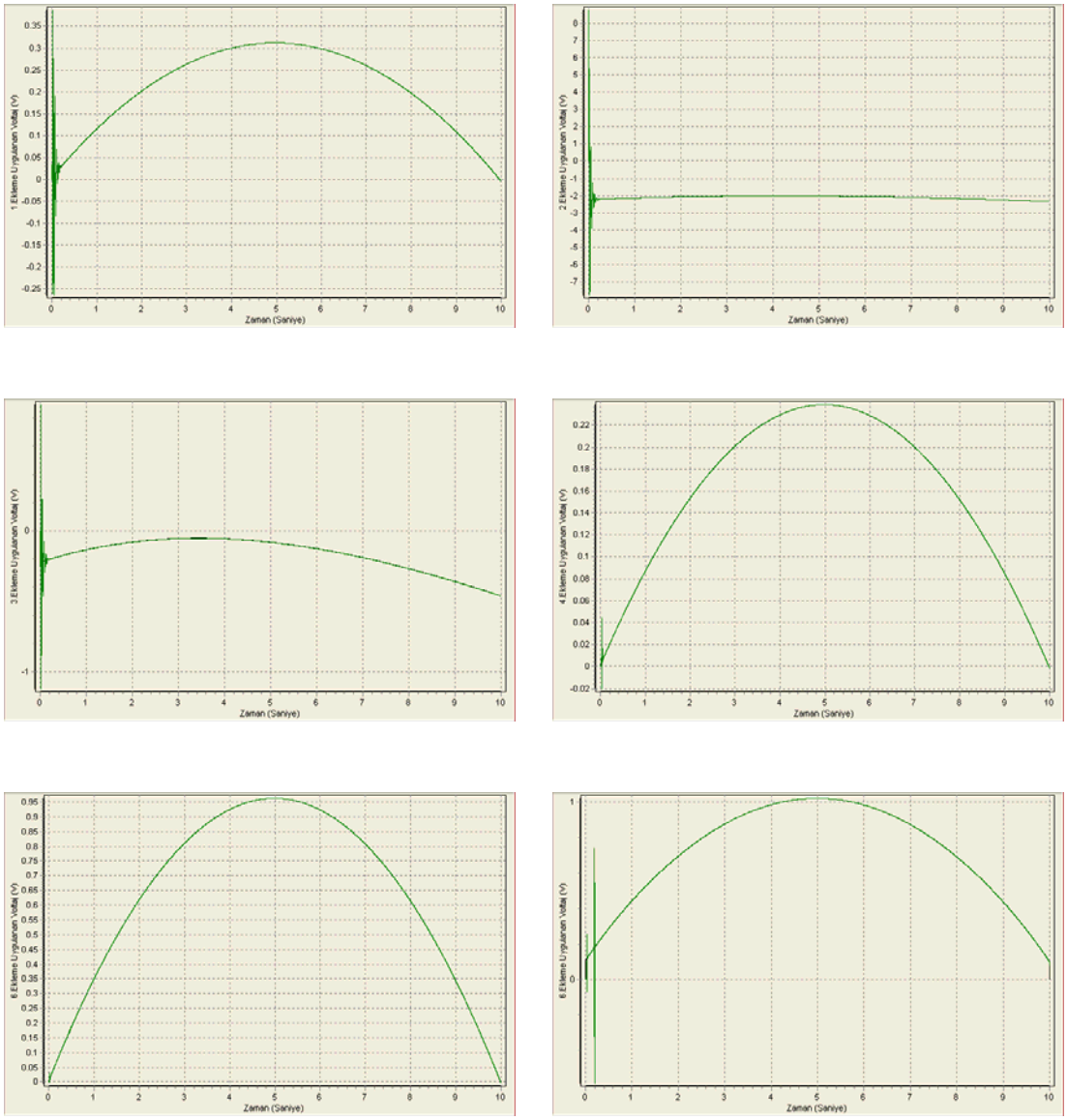



Şekil 4.29. Eklemlerin açısal hız grafikleri (GPC MIMO, Örnek 1, Durum 1)

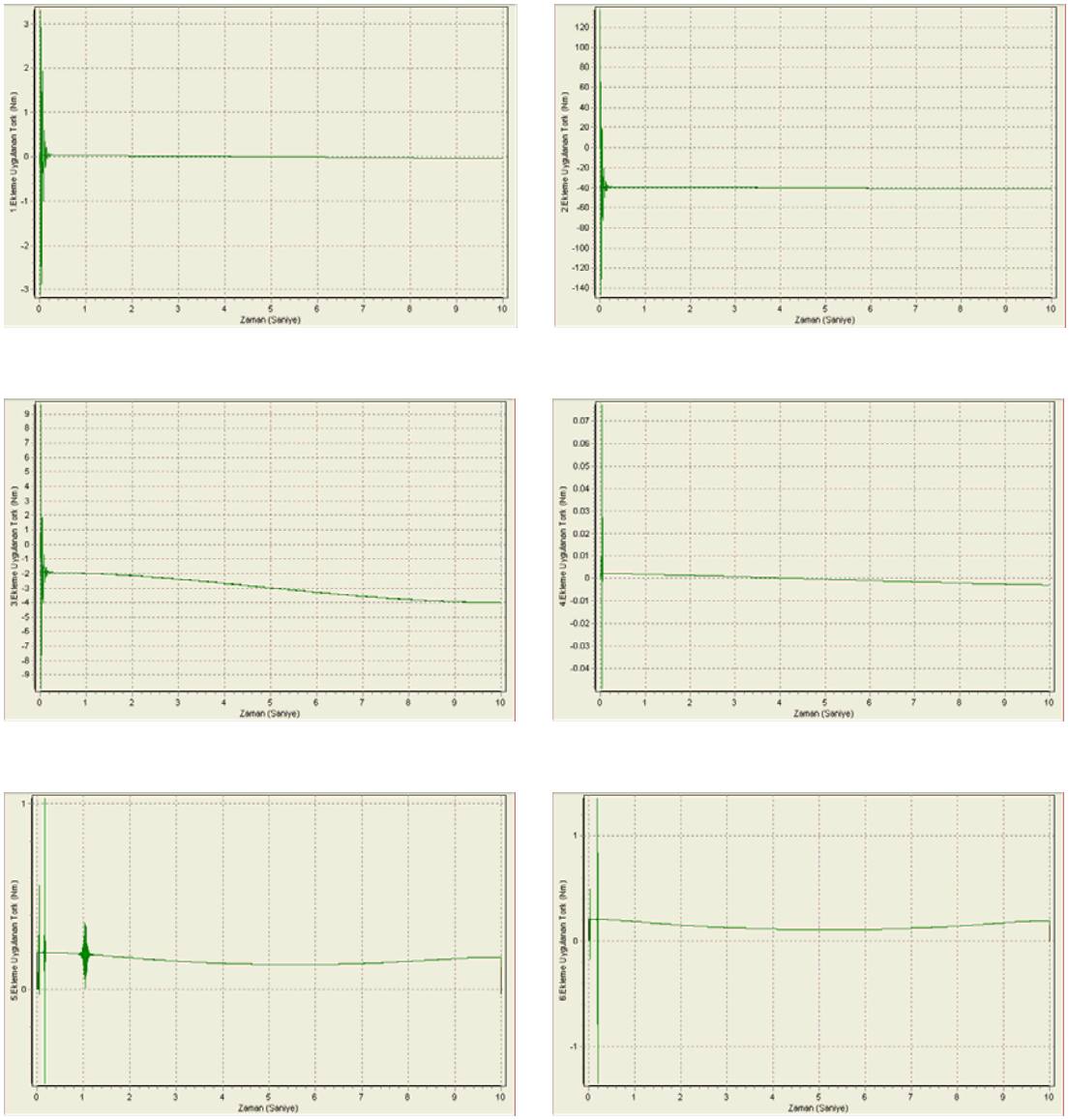



Şekil 4.30. Eklemlerin açısal hız hata grafikleri (GPC MIMO, Örnek 1, Durum 1)

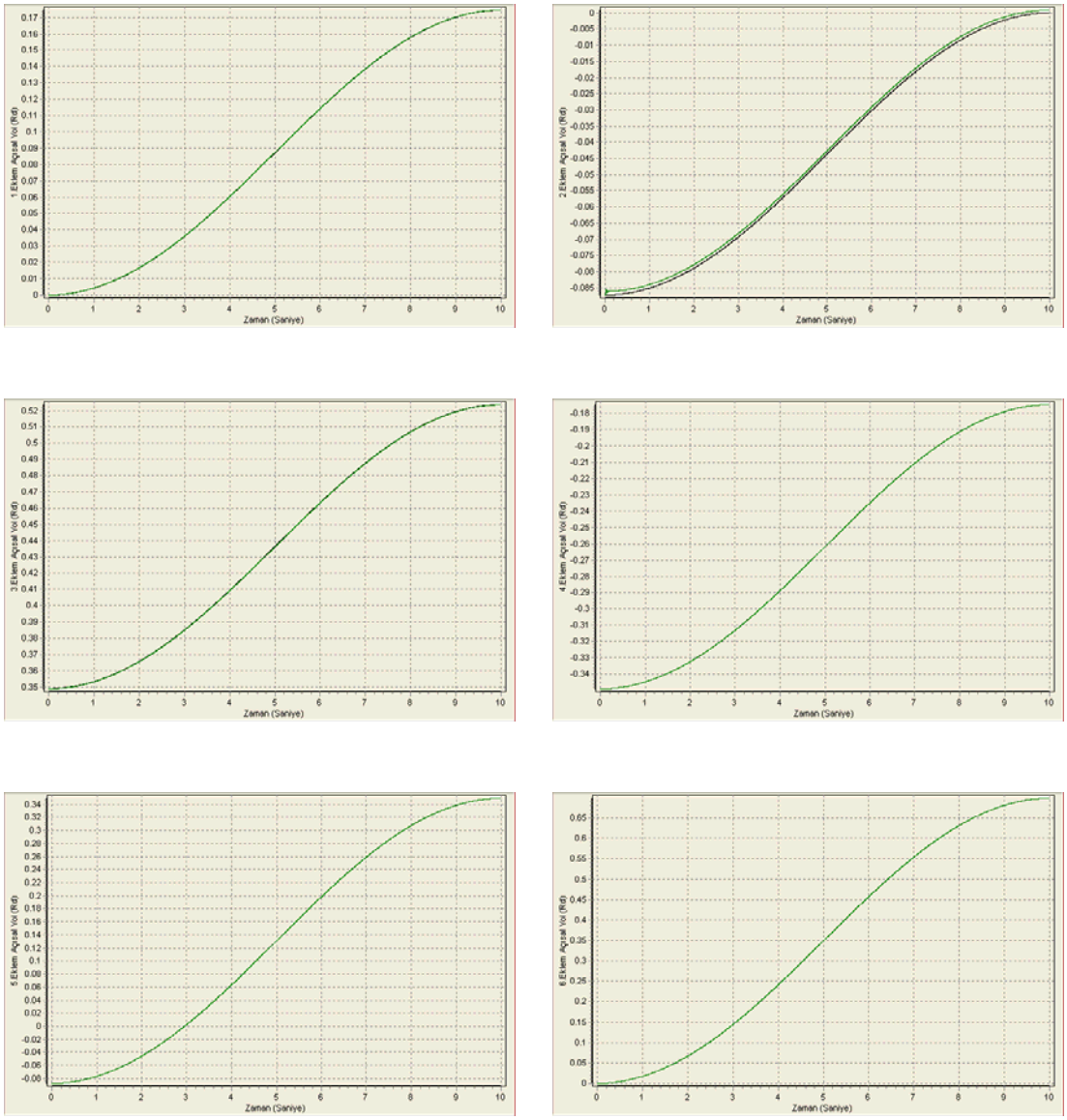
Örnek 1 Durum 1 ( Yük, sürtünme yok) için GPC MIMO algoritması ile yapılan simülasyon sonuçlarına bakıldığında tork ve gerilim eğrileri gayet düzgündür. Açısal yol grafiklerinden görüldüğü gibi referans yörüngeye yakın bir takip sağlanmakla beraber eğriler arasında ki fark açıkça görülebilmektedir. Dolayısıyla bu fark açısal hızlara da yansımıştır ve referans hızlardan bir miktar sapmalar oluşmuştur. Hareket başlangıcındaki salınımlar GPC-MIMO algoritmasına ait grafiklerde de açıkça görülmektedir.



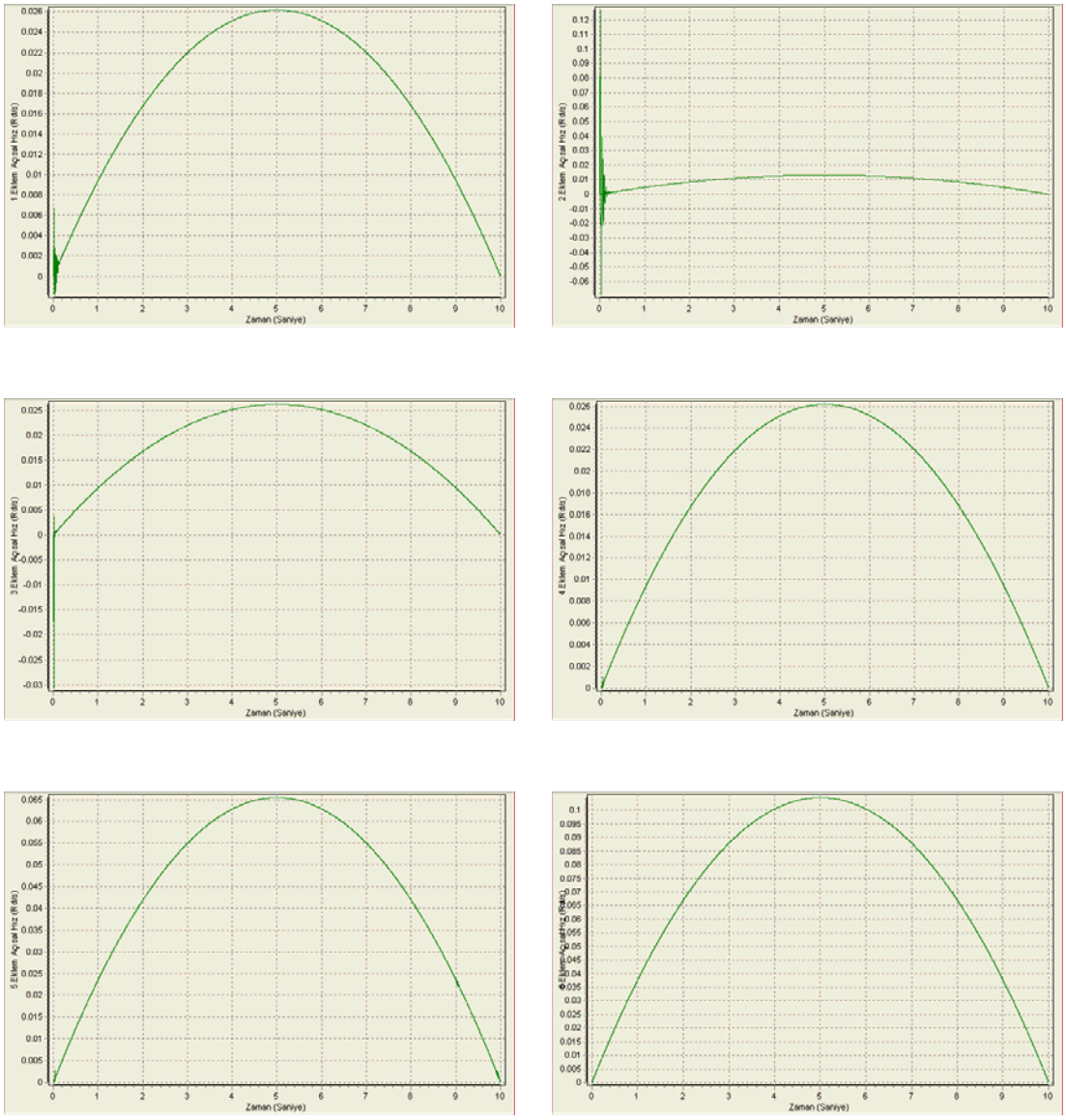

Şekil 4.31. Eklemlere uygulanan gerilim grafikleri (SGA-GPC SISO, Örnek 1, Durum 1)




Şekil 4.32. Eklemlere uygulanan tork grafikleri (SGA-GPC SISO, Örnek 1, Durum 1)



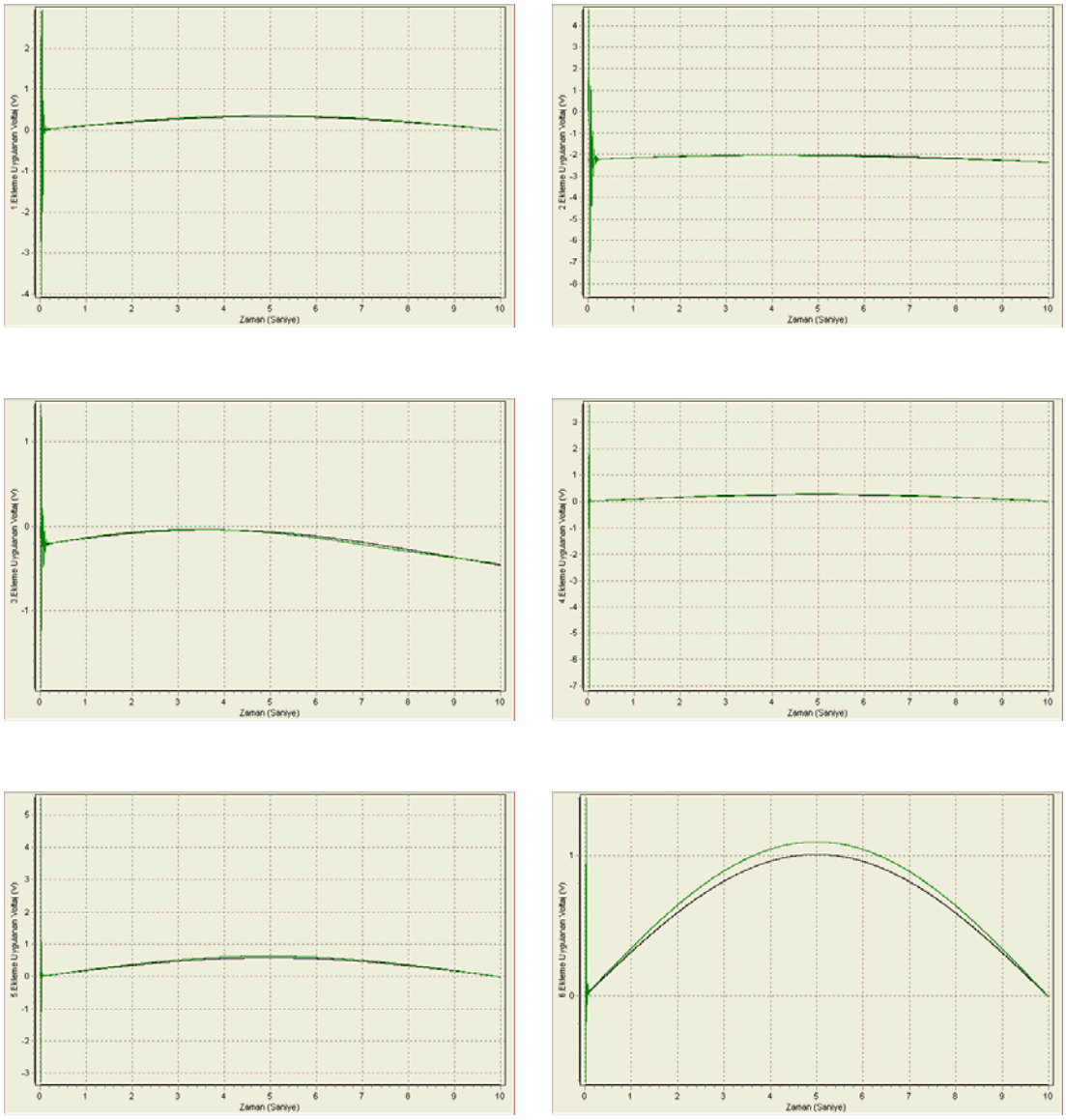

Şekil 4.33. Eklemlerin takip ettiği açısal yol grafikleri (SGA-GPC SISO, Örnek 1, Durum 1)



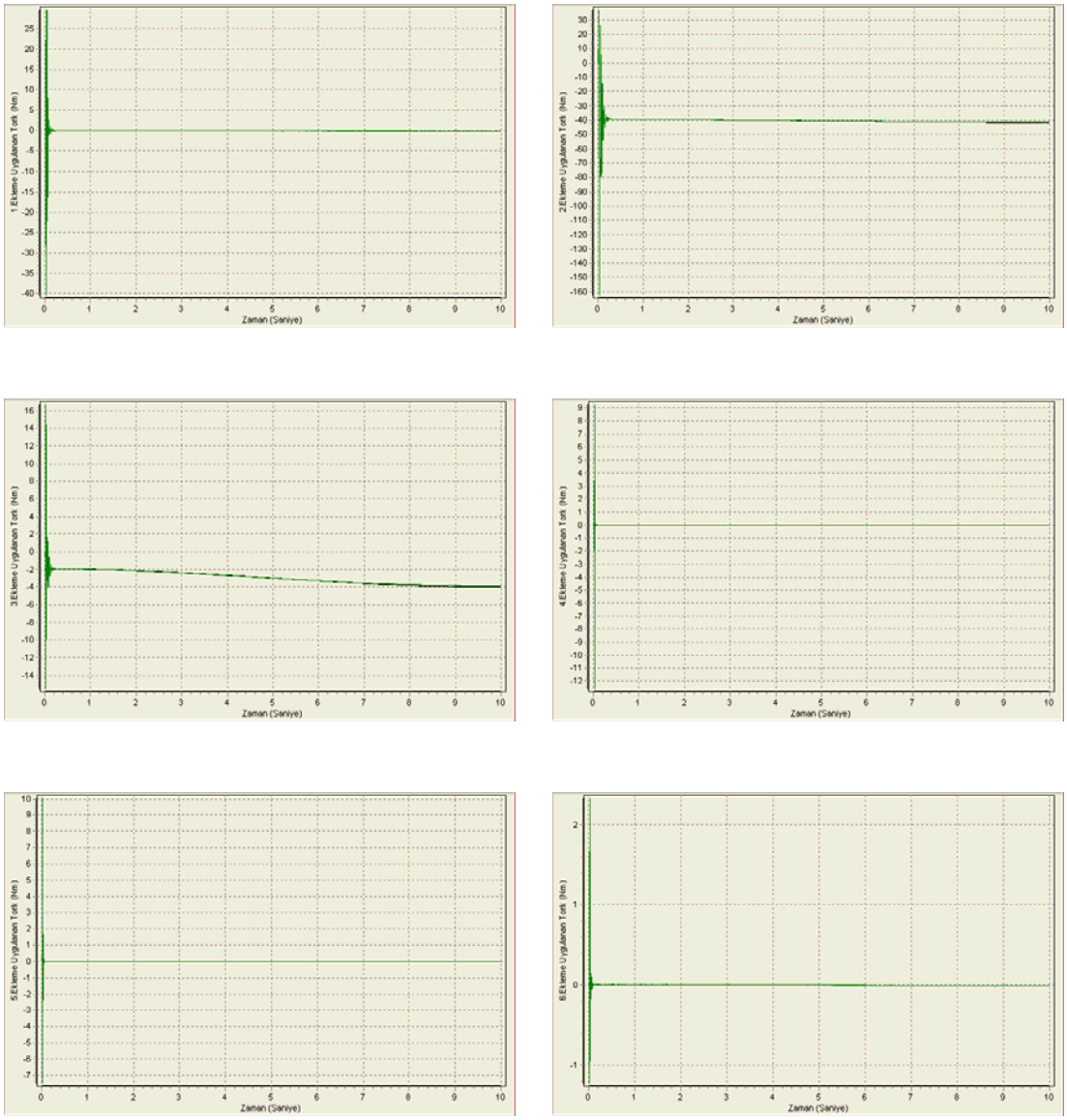

Şekil 4.34. Eklemlerin açısal hız grafikleri (SGA-GPC SISO, Örnek 1, Durum 1)


Şekil 4.35. Eklemlerin açısal hız hata grafikleri (SGA-GPC SISO, Örnek 1, Durum 1)

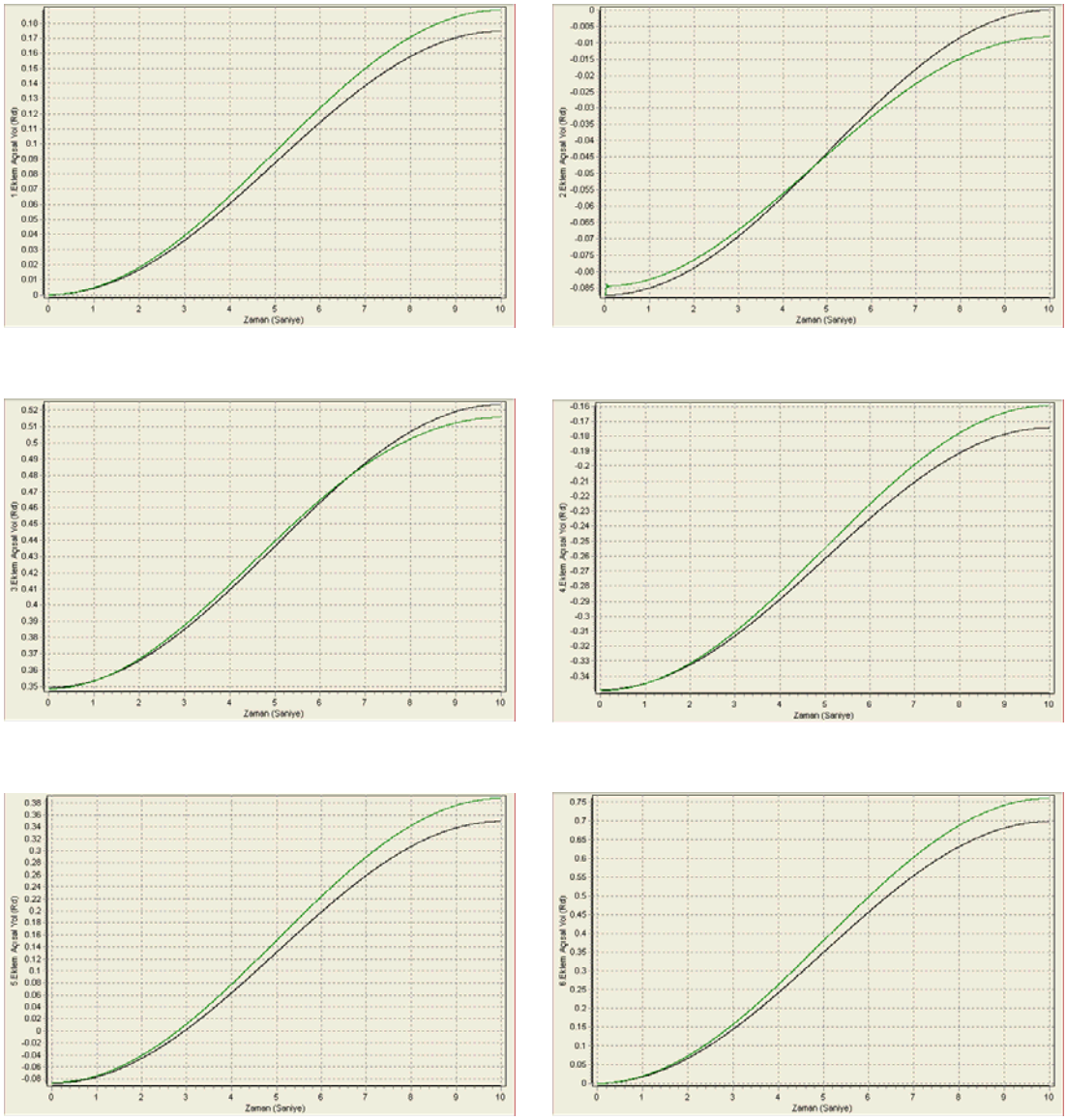
Örnek 1 Durum 1 ( Yük, sürtünme yok) için SGA-GPC SISO algoritması ile yapılan simülasyon sonuçlarına bakıldığında Tork, açısal yol ve açısal hız grafikleri gayet düzgündür. Eklemlere uygulanan gerilim ile tork eğrileri paralellik göstermektedirler. Eklemler referans yörüngelerini çok yakın takip ettikleri için açısal yol grafiklerinde referans yörünge ile gerçeklenen yörünge eğrileri üst üste çakışmaktadırlar. Açısal hız hataları oldukça düşüktür. Tork grafiklerinde görüldüğü gibi, hareket başlangıcındaki robot kolu eylemsizliğinden kaynaklanan salınımlar ile referans yörüngelerden sapmalar oluşmakla birlikte kontrolör hatayı telafi ederek istenilen yörüngeyi kısa sürede yakalayabilmektedir.



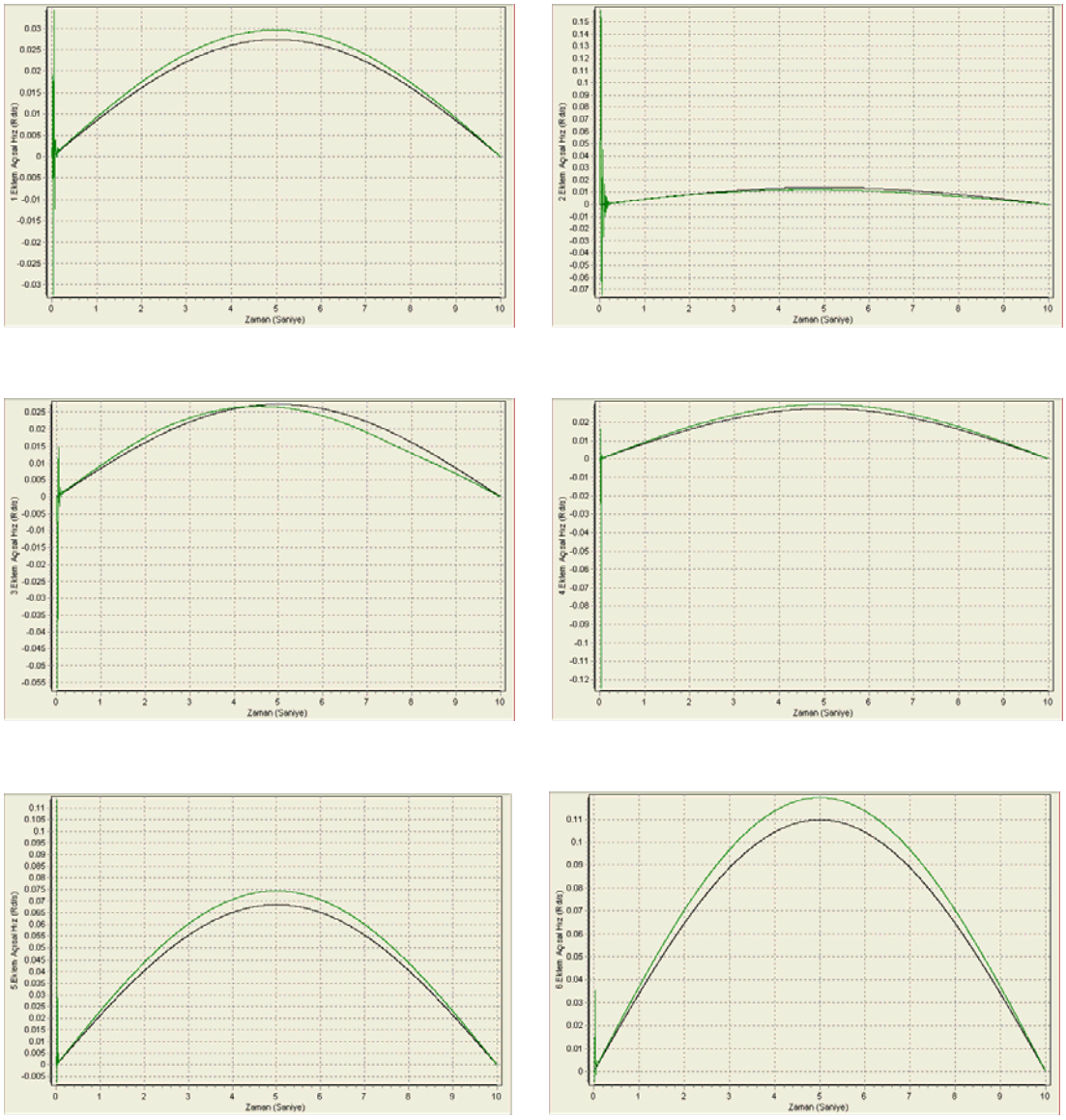

Şekil 4.36. Eklemlere uygulanan gerilim grafikleri (SGA-GPC MIMO, Örnek 1, Durum 1)



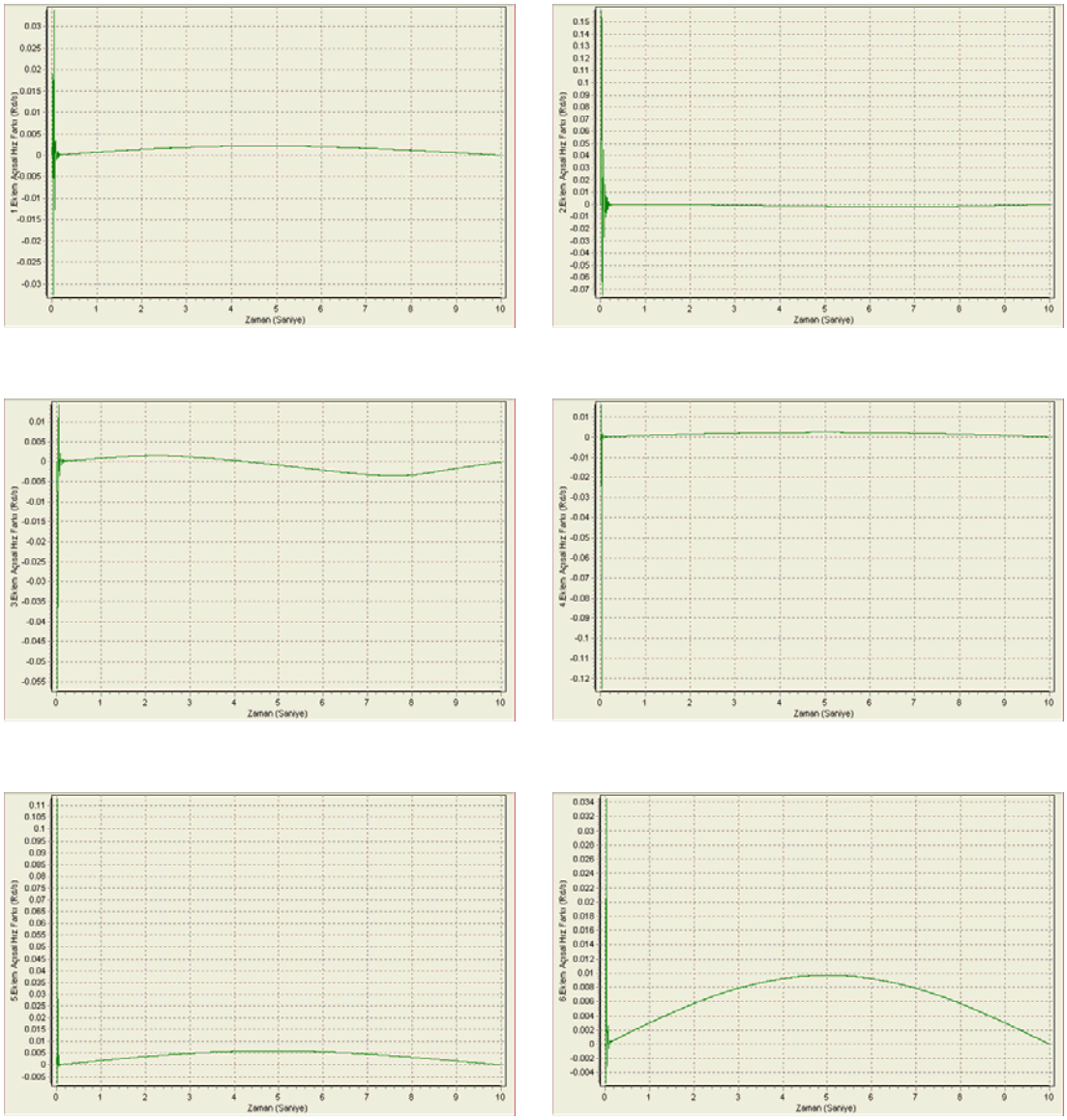

Şekil 4.37. Eklemlere uygulanan tork grafikleri (SGA-GPC MIMO, Örnek 1, Durum 1)




Şekil 4.38. Eklemlerin takip ettiği açısal yol grafikleri (SGA-GPC MIMO, Örnek 1, Durum 1)

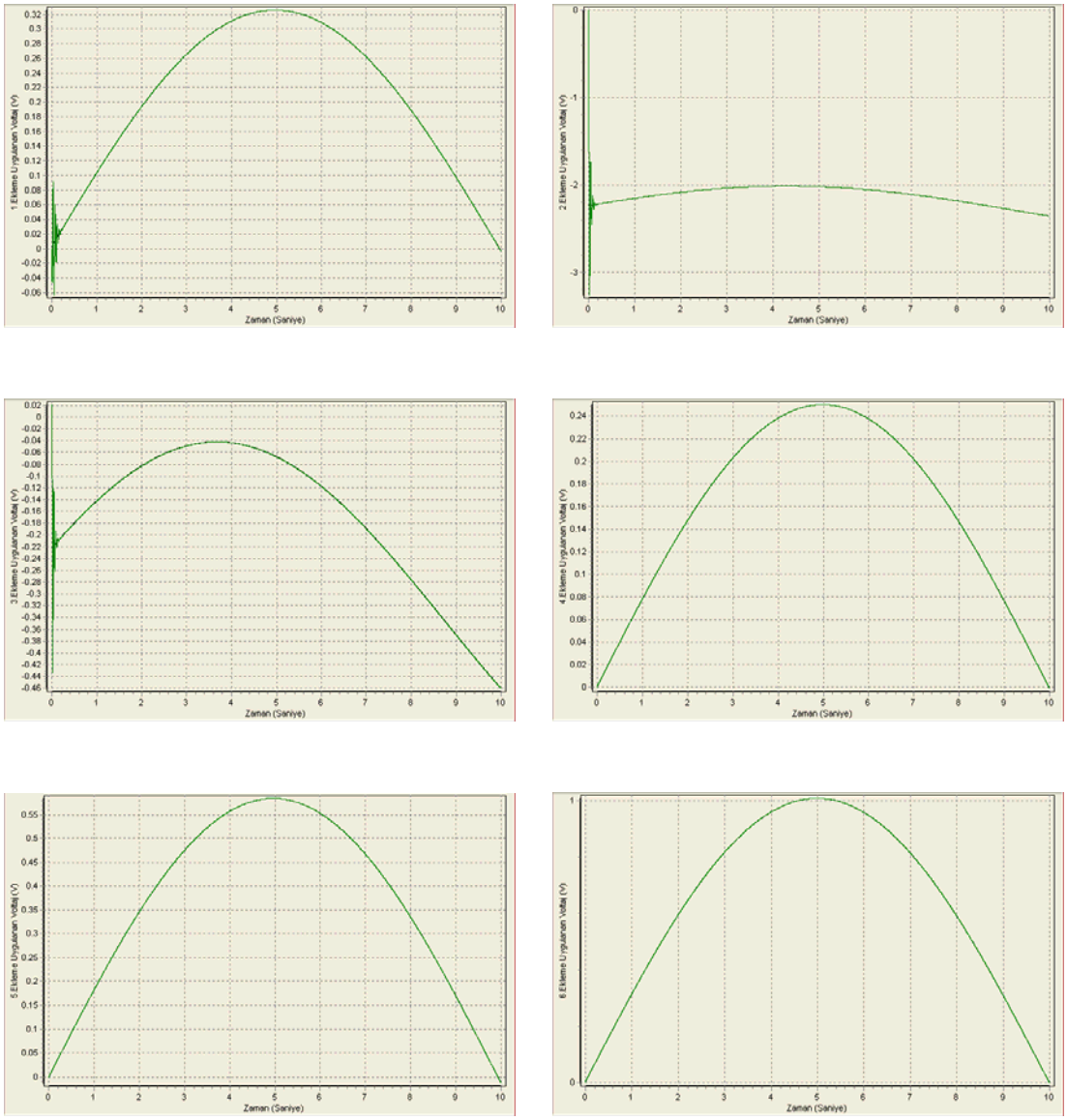



Şekil 4.39. Eklemlerin açısal hız grafikleri (SGA-GPC MIMO, Örnek 1, Durum 1)

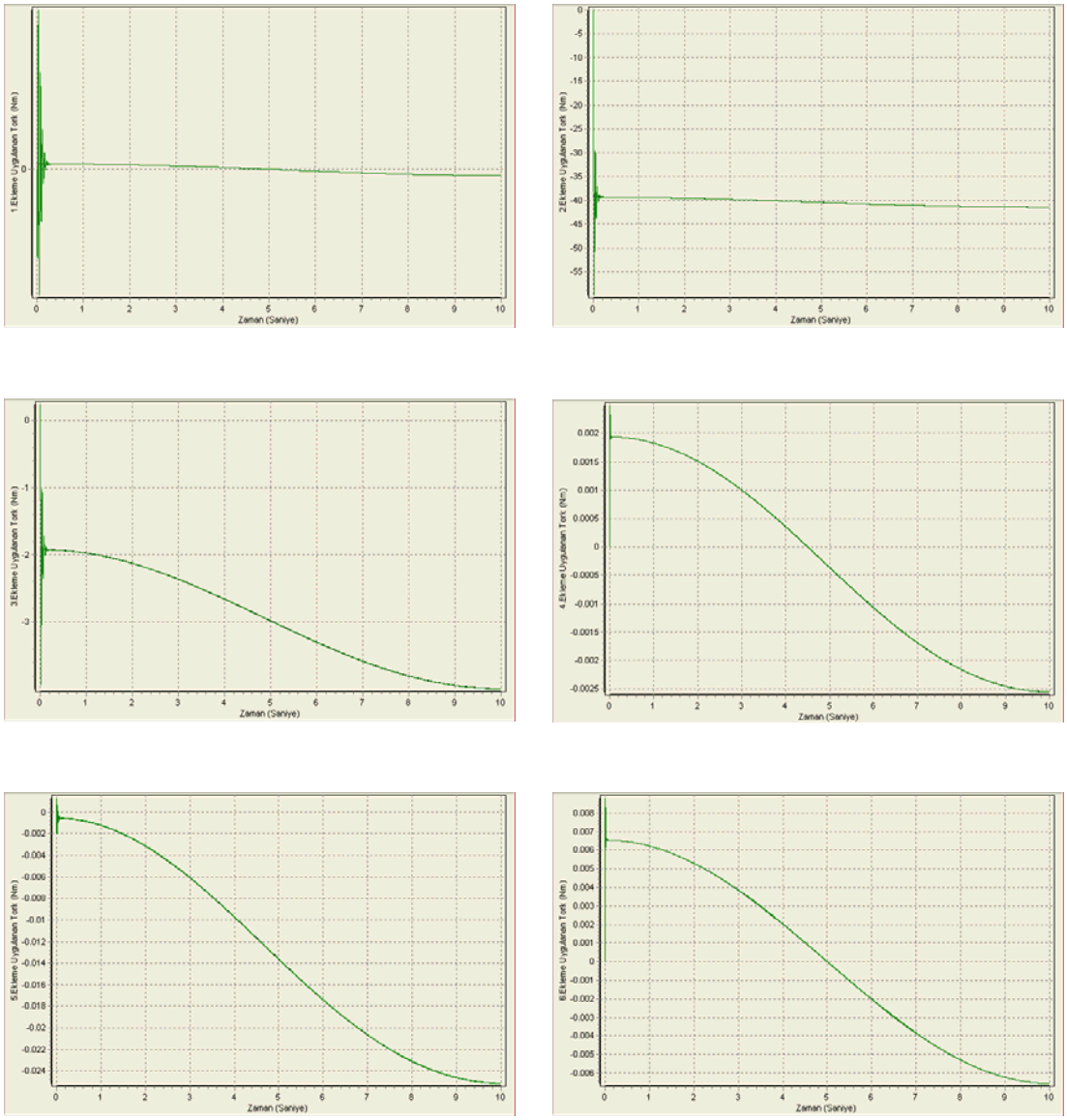



Şekil 4.40. Eklemlerin açısal hız hataları (SGA-GPC MIMO, Örnek 1, Durum 1)

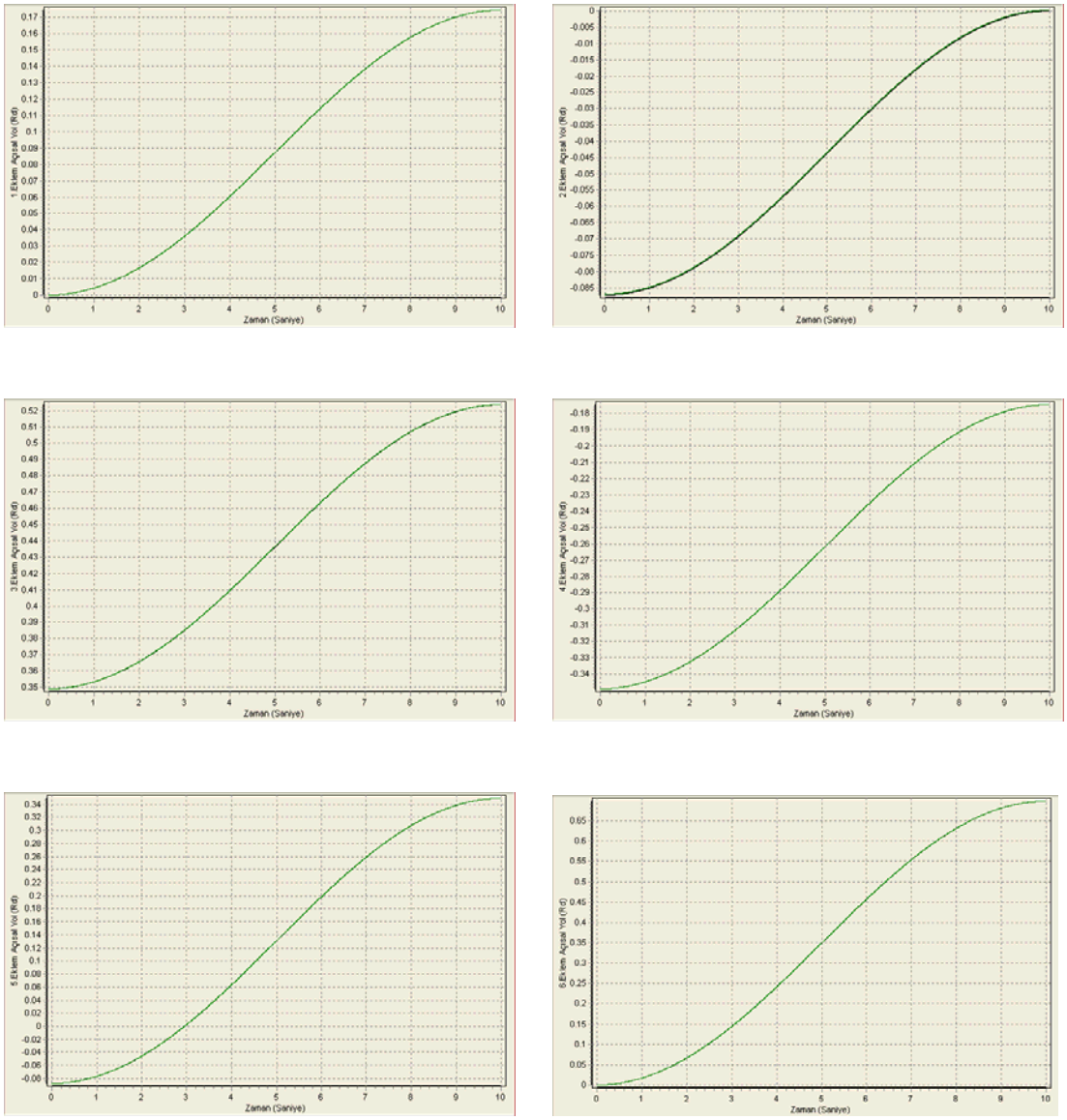
Örnek 1 Durum 1 ( Yük, sürtünme yok) için SGA-GPC MIMO algoritması ile yapılan simülasyon sonuçlarında tork ve gerilim eğrileri gayet düzgündür. Açısal yol grafiklerinden görüldüğü gibi referans yörüngeye yakın bir takip sağlanmakla beraber eğriler arasındaki fark açıkça görülebilmektedir. Dolayısıyla bu fark açısal hızlara da yansımıştır ve referans hızlardan bir miktar sapmalar oluşmuştur.



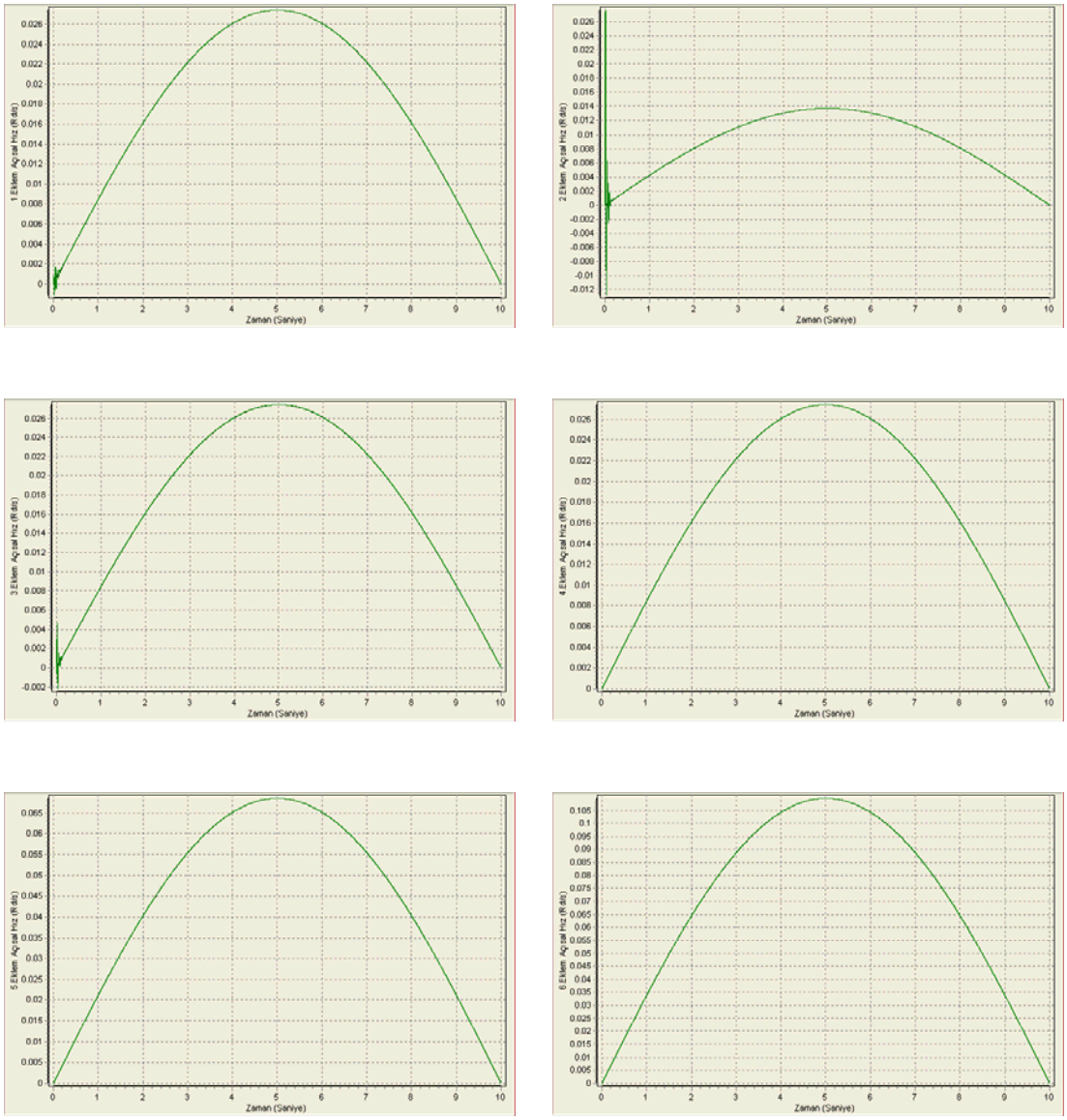

Şekil 4.41. Eklemlere uygulanan gerilim grafikleri (NGPC SISO, Örnek 1, Durum 1)



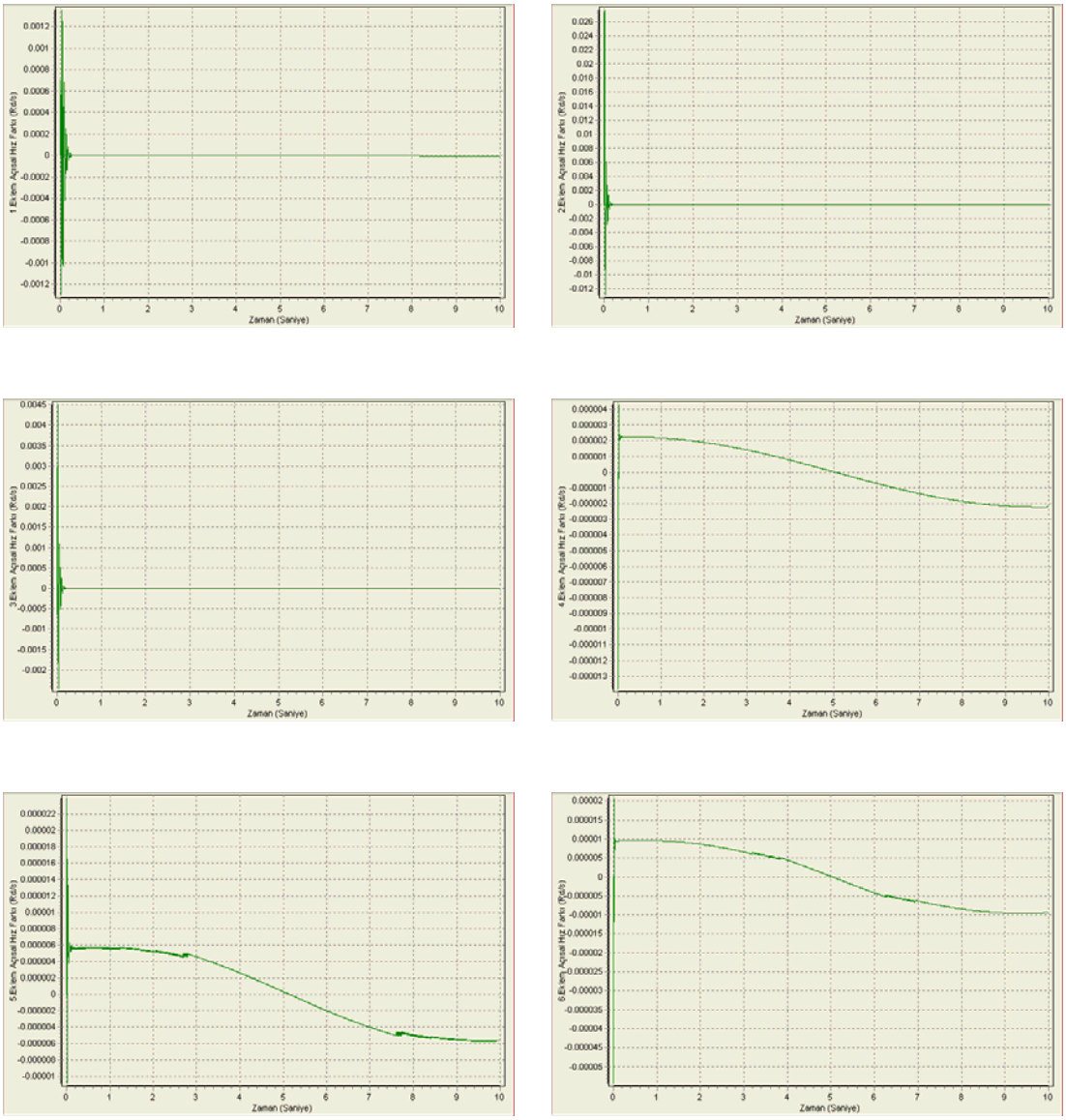

Şekil 4.42. Eklemlere uygulanan tork grafikleri (NGPC SISO, Örnek 1, Durum 1)




Şekil 4.43. Eklemlerin takip ettiği açısal yol grafikleri (NGPC SISO, Örnek 1, Durum 1)




Şekil 4.44. Eklemlerin açısal hız grafikleri (NGPC SISO, Örnek 1, Durum 1)

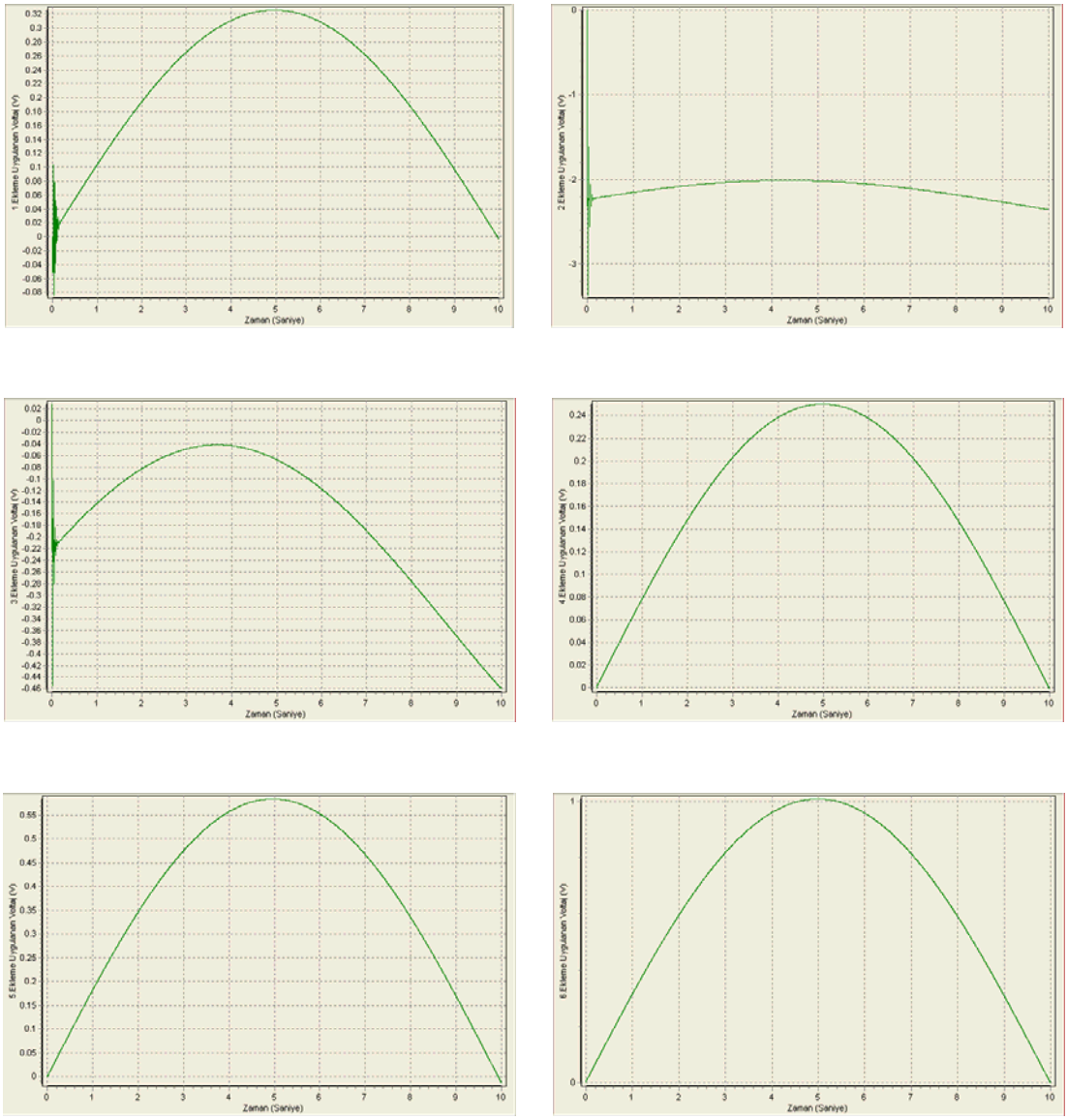



Şekil 4.45. Eklemlerin açısal hız hataları (NGPC SISO, Örnek 1, Durum 1)

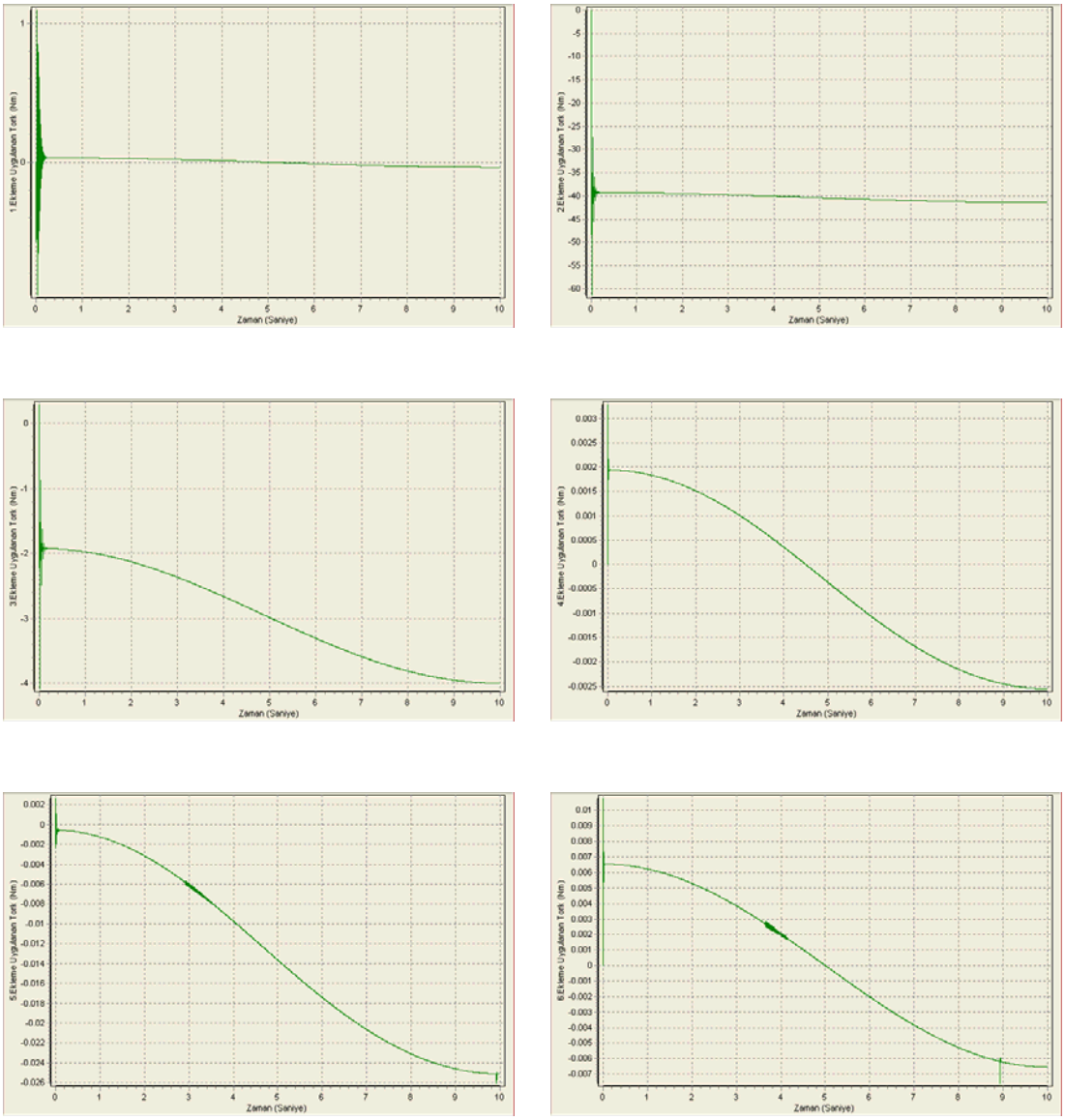
Örnek 1 Durum 1 ( Yük, sürtünme yok) için NGPC SISO algoritması ile yapılan simülasyon sonuçlarının grafikleri kısaca şu şekilde yorumlanabilir.

Tork, açısal yol ve açısal hız grafikleri gayet düzgündür. Eklemlere uygulanan gerilim ile tork eğrileri paralellik göstermektedirler. Eklemler referans yörüngelerini çok yakın takip ettikleri için açısal yol grafiklerinde referans yörünge ile gerçeklenen yörünge eğrileri üst üste çakışmaktadırlar. Açısal hız hataları oldukça düşüktür ve grafiklerine bakıldığında 0 ekseni üzerinde bir takip sağlanmıştır. Sadece hareket başlangıcında hareketsiz durumdaki robot kolunun eylemsizliğinden dolayı açısal hız

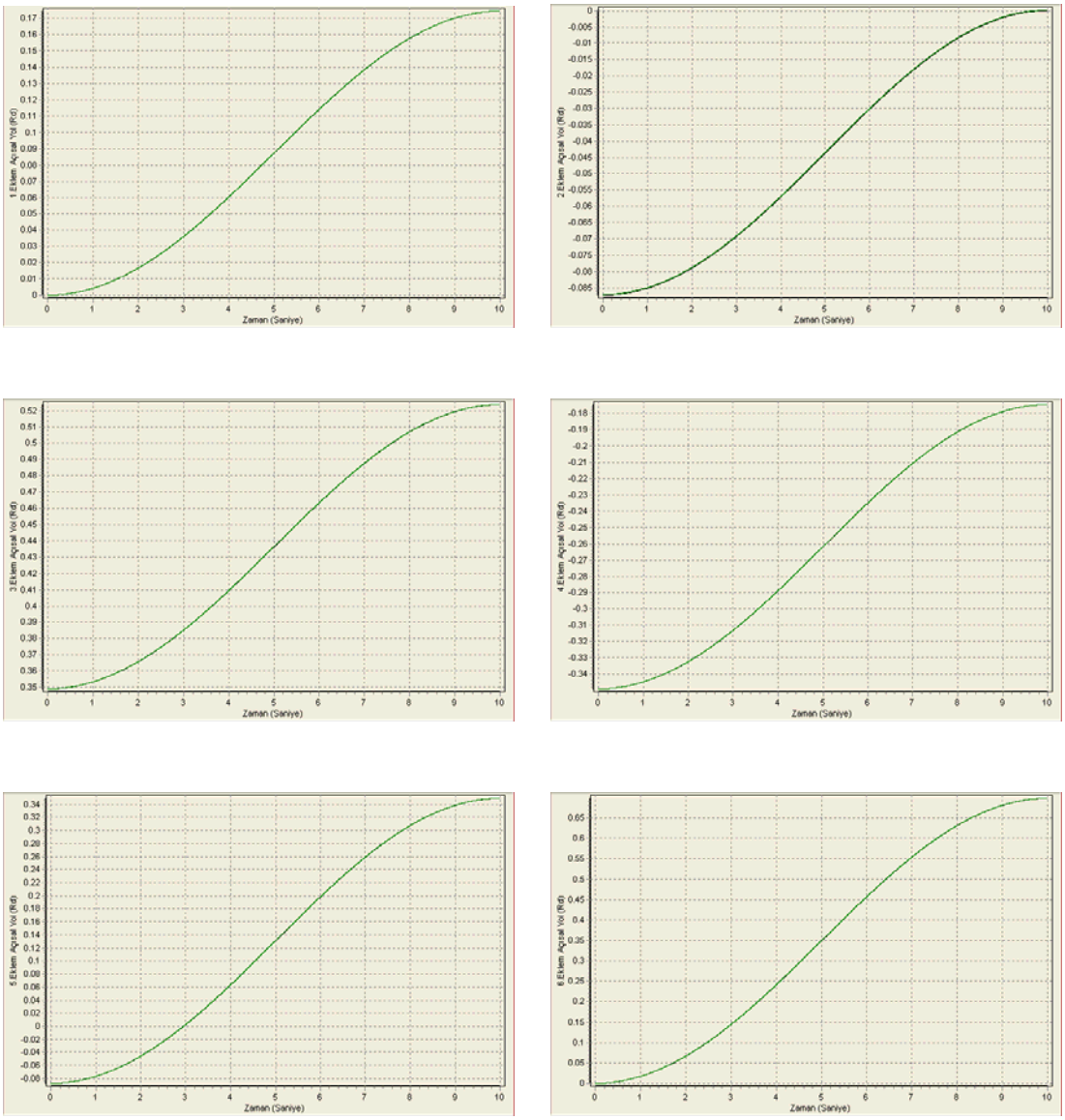
hataları oluşmuştur. Ve farklı genlikteki salınımlardan sonra referans yörüngeler yakalanmış ve açısal hızlar sıfıra yakınsamıştır.



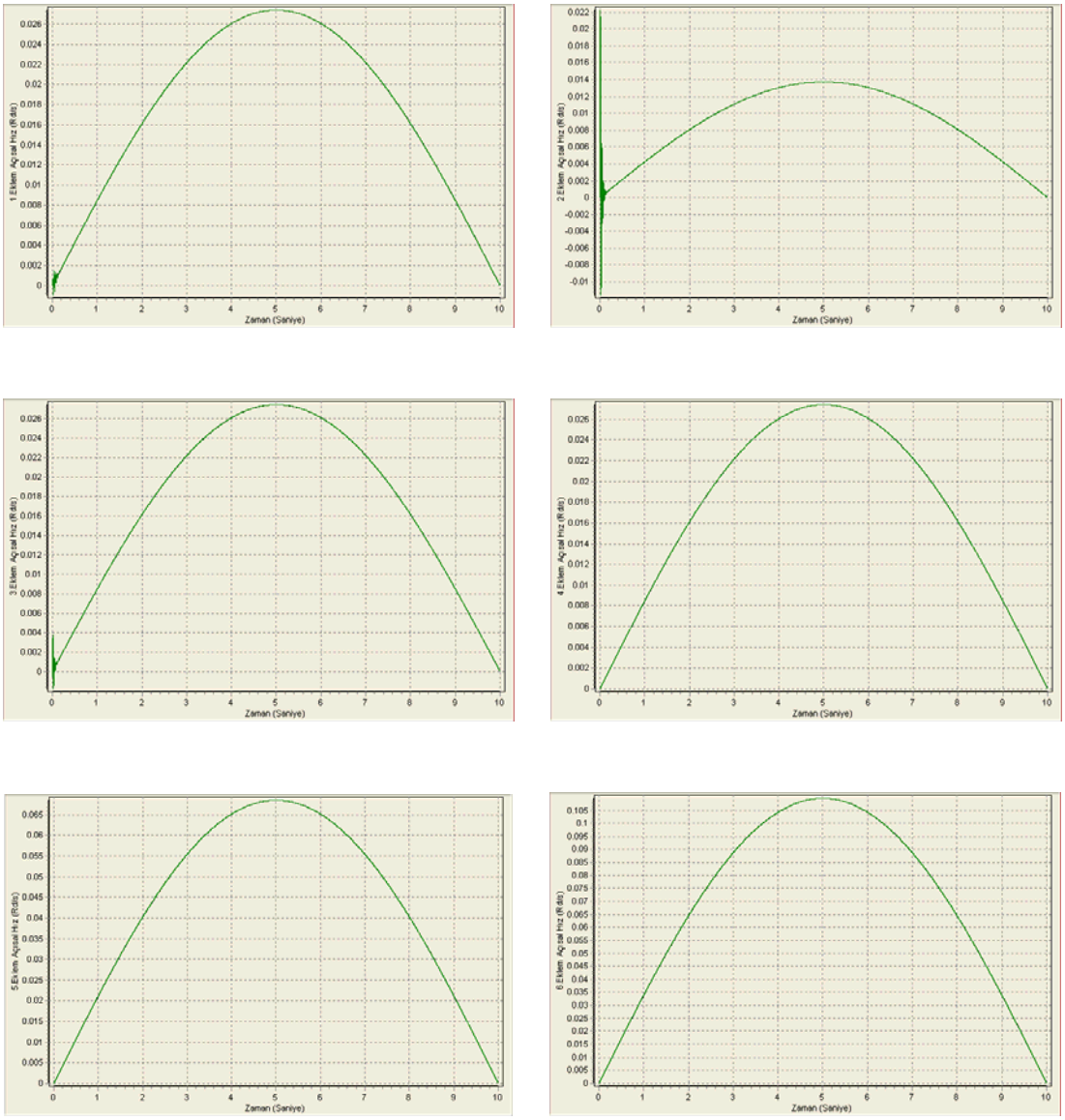

Şekil 4.46. Eklemlere uygulanan gerilim grafikleri (ENGPC SISO, Örnek 1, Durum 1)



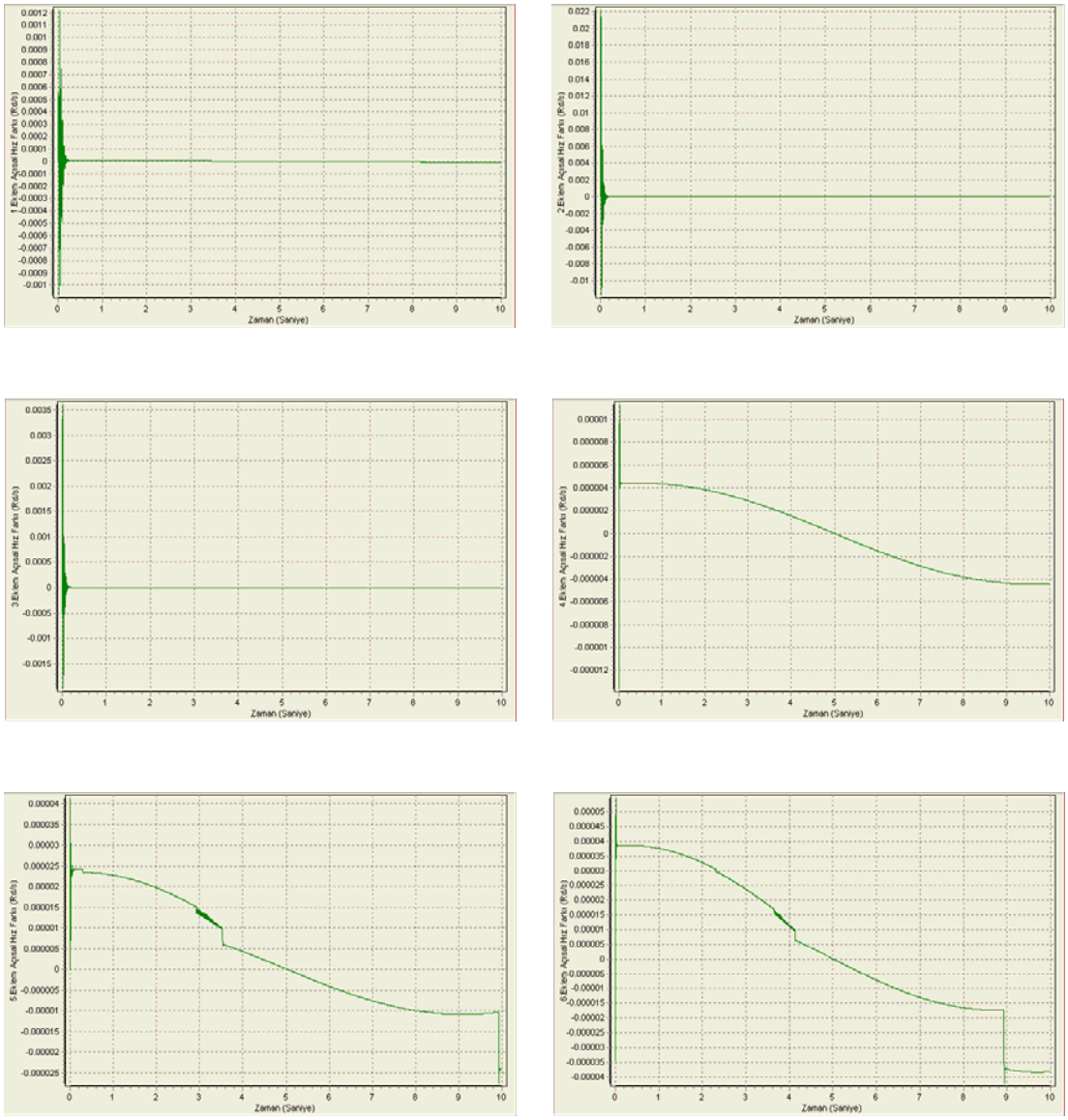

Şekil 4.47. Eklemlere uygulanan tork grafikleri (ENGPC SISO, Örnek 1, Durum 1)




Şekil 4.48. Eklemlerin takip ettiği açısal yol grafikleri (ENGPC SISO, Örnek 1, Durum 1)




Şekil 4.49. Eklemlerin açısal hız grafikleri (ENGPC SISO, Örnek 1, Durum 1)

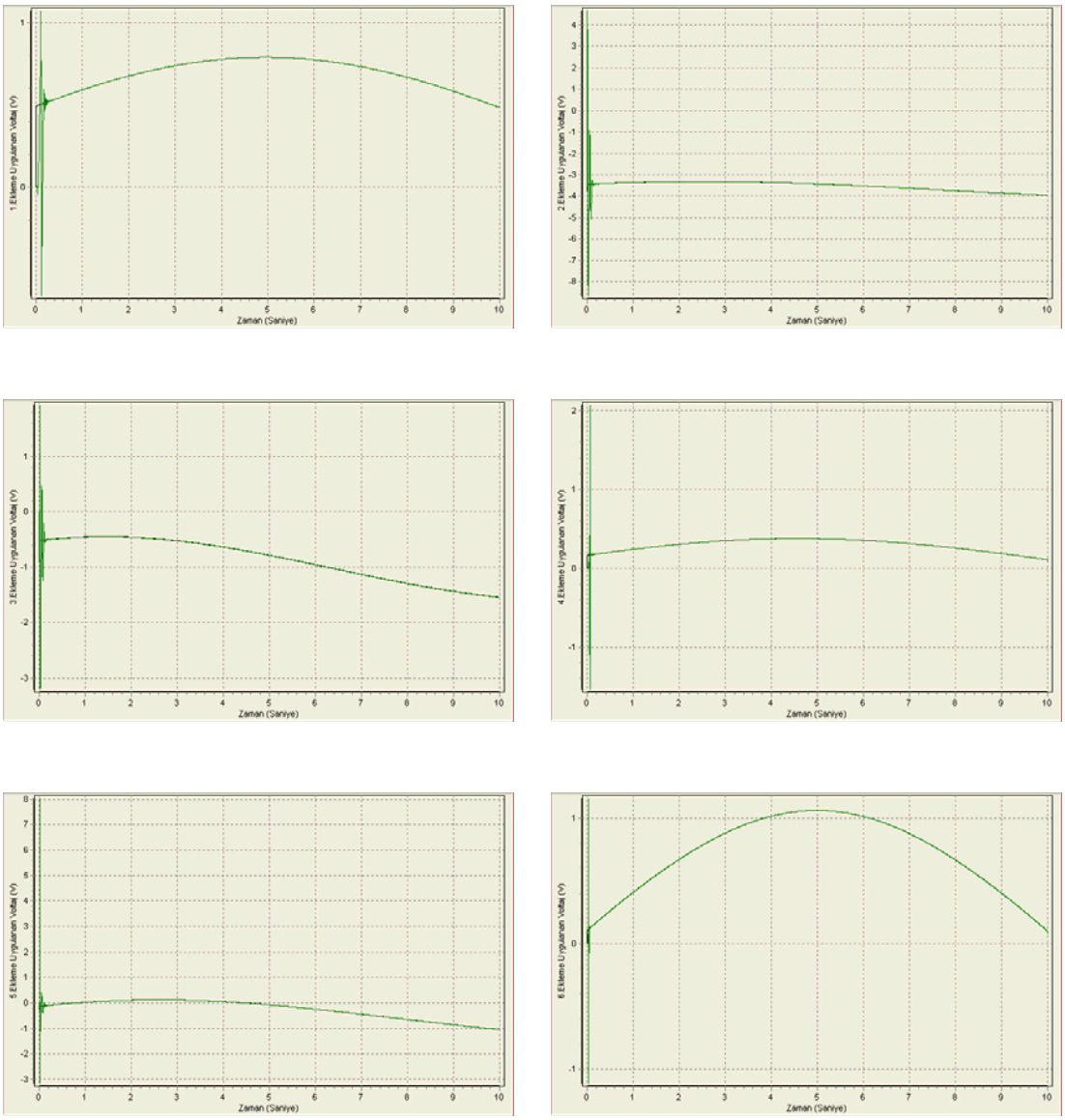



Şekil 4.50. Eklemlerin açısal hız hataları (ENGPC SISO, Örnek 1, Durum 1)

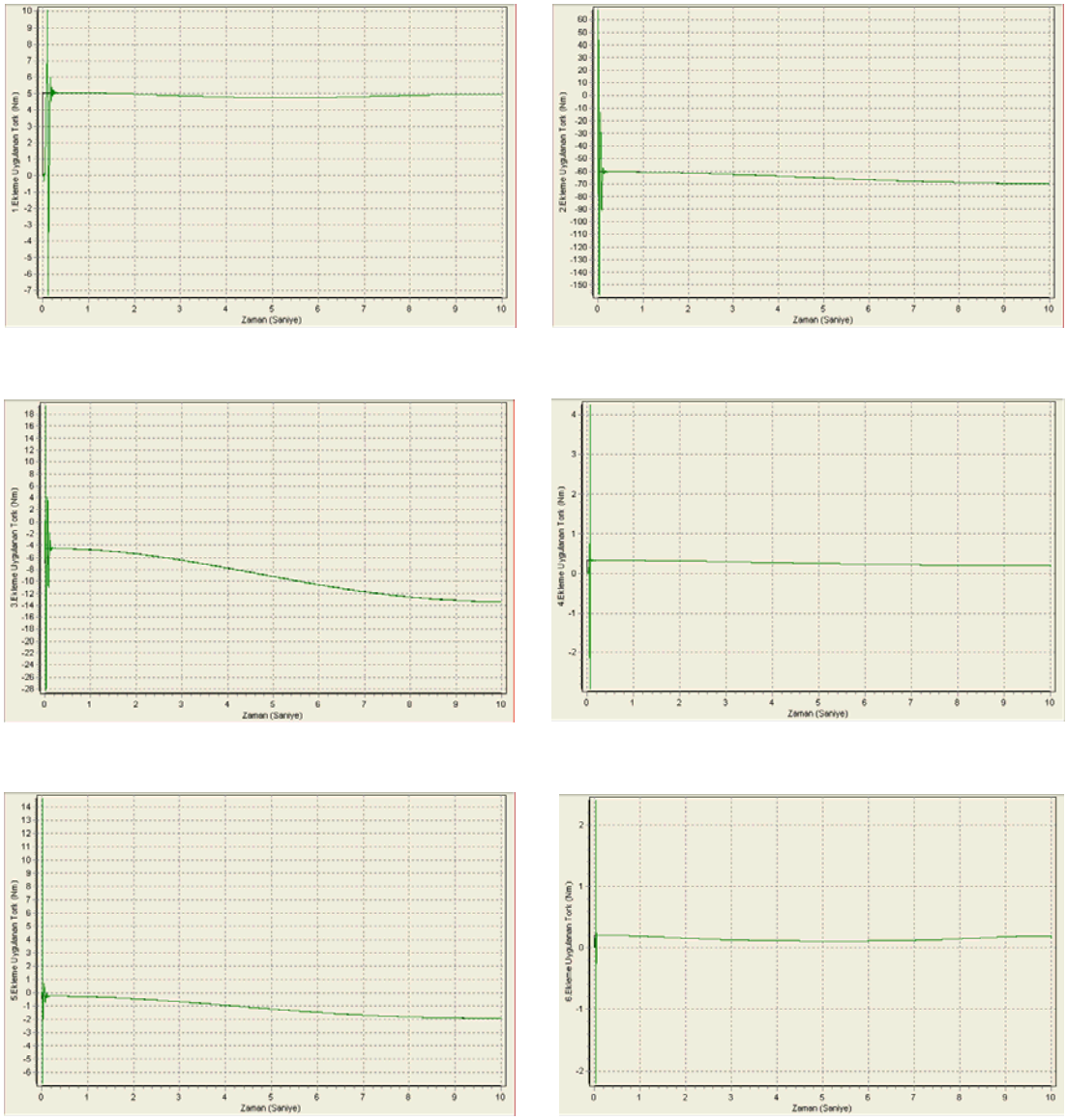
Örnek 1 Durum 1 ( Yük, sürtünme yok) için ENGPC SISO algoritması ile tork, açısal yol ve açısal hız grafikleri gayet düzgündür. Eklemlere uygulanan gerilim ile tork eğrileri paralellik göstermektedirler. Eklemler referans yörüngelerini çok yakın takip ettikleri için açısal yol grafiklerinde referans yörünge ile gerçeklenen yörünge eğrileri üst üste çakışmaktadırlar. Açısal hız hataları oldukça düşüktür ve grafiklerine bakıldığında 0 ekseni üzerinde bir takip sağlanmıştır. Sadece hareket başlangıcında hareketsiz durumdaki robot kolunun eylemsizliğinden dolayı açısal hız hataları oluşmuştur. Ve farklı genlikteki salınımlardan sonra referans yörüngeler yakalanmış ve açısal hızlar sıfıra yakınsamıştır.

1. Durum için kontrol algoritmaları genel olarak başarılıdır. Özellikle SISO yapıdaki algoritmalar referans yörüngeleri çok yakın takip etmişlerdir. Bunun sonucu eklemler çok az açısal hız hatalar ile hareketlerini tamamlamışlardır. MIMO yapılardaki algoritmalar aşağı yukarı referans yörüngeleri takip etmekle birlikte aralarındaki farklar grafiklerde açıkça görülmektedir.

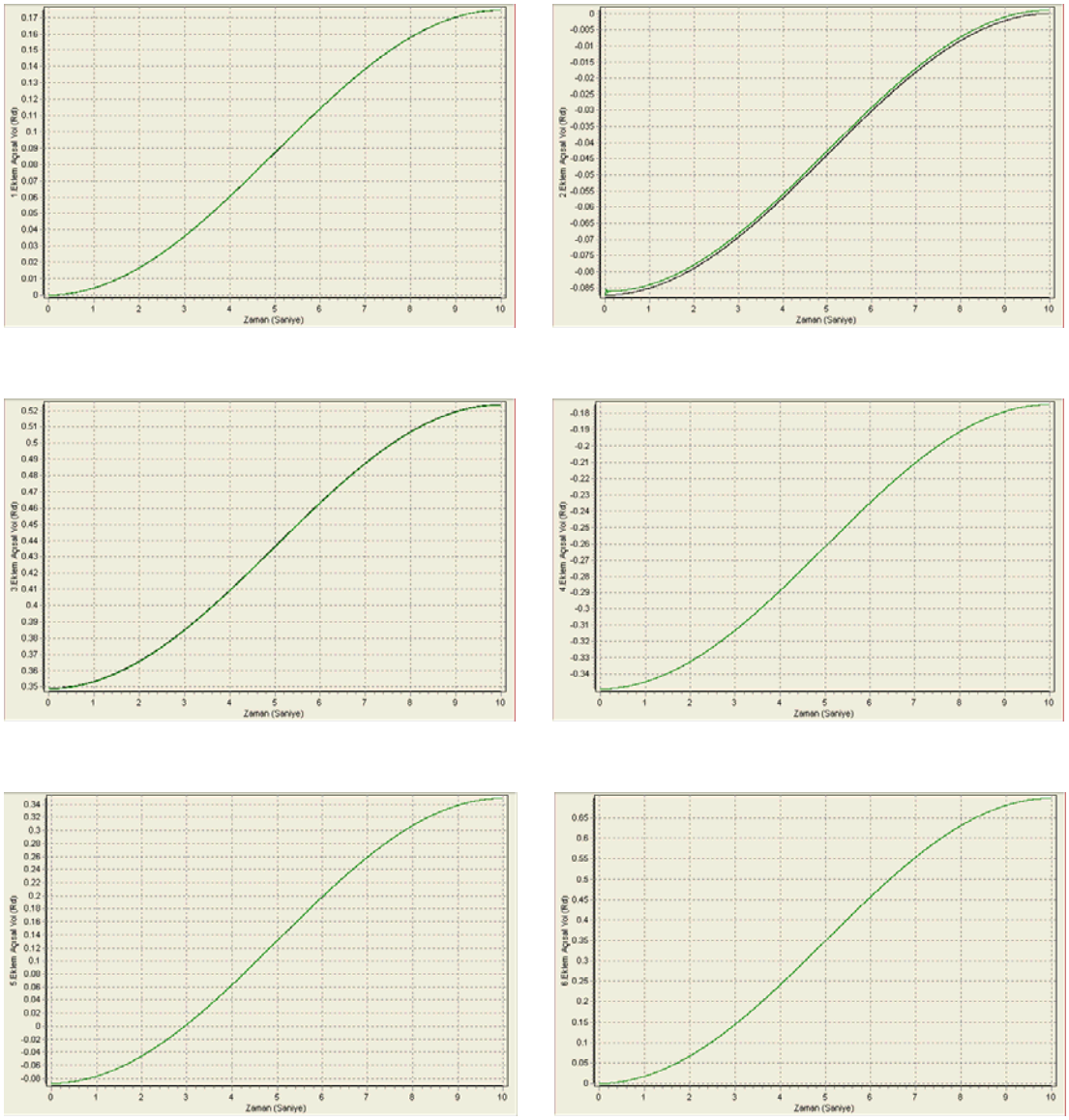
2. Durum ‘da yapılan simülasyon sonuçları Şekil 4.51 ile Şekil 4.80 arasında aşağıda verilmiştir.



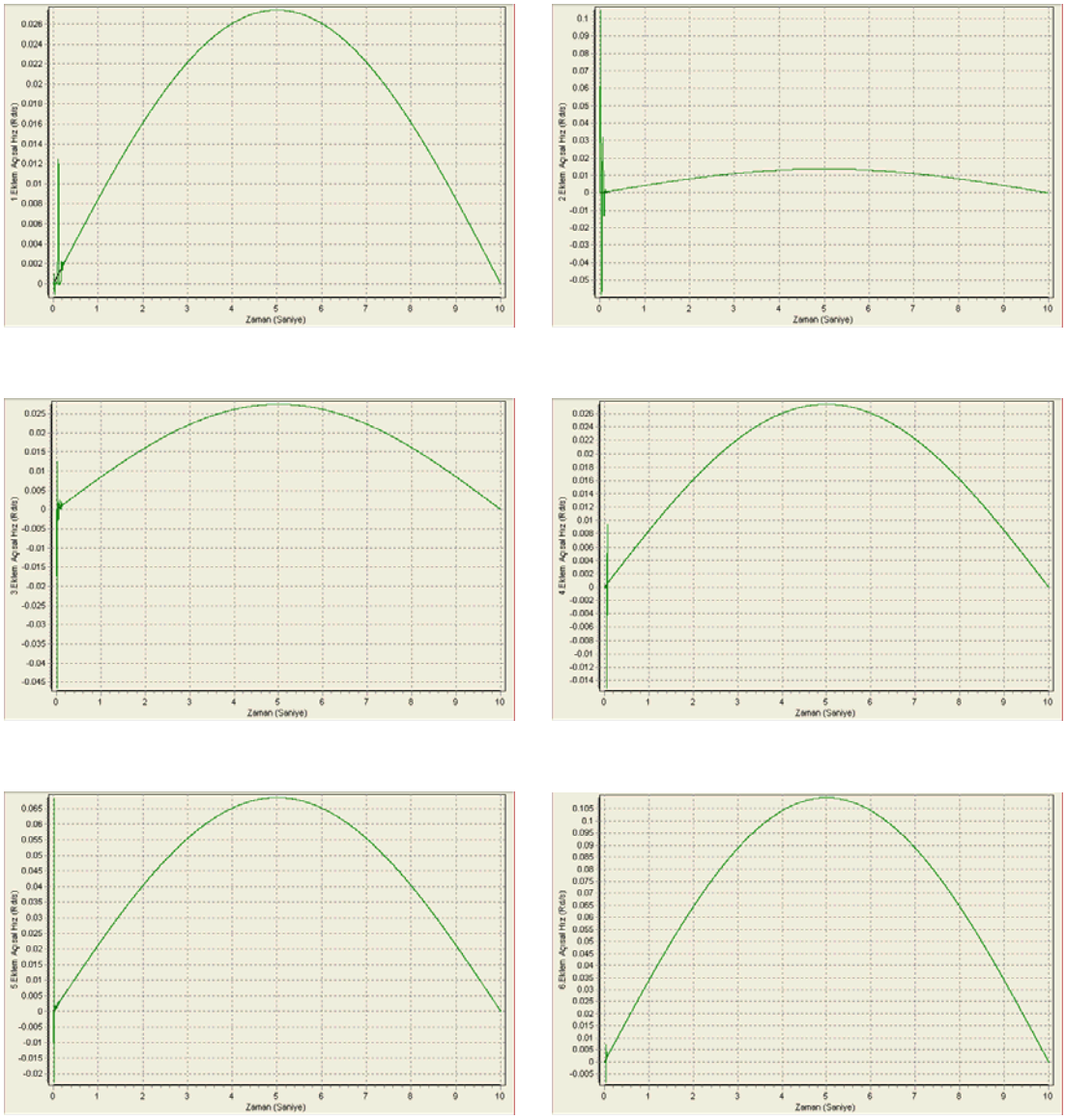

Şekil 4.51. Eklemlere uygulanan gerilim grafikleri (GPC SISO, Örnek 1, Durum 2)




Şekil 4.52. Eklemlere uygulanan tork grafikleri (GPC SISO, Örnek 1, Durum 2)



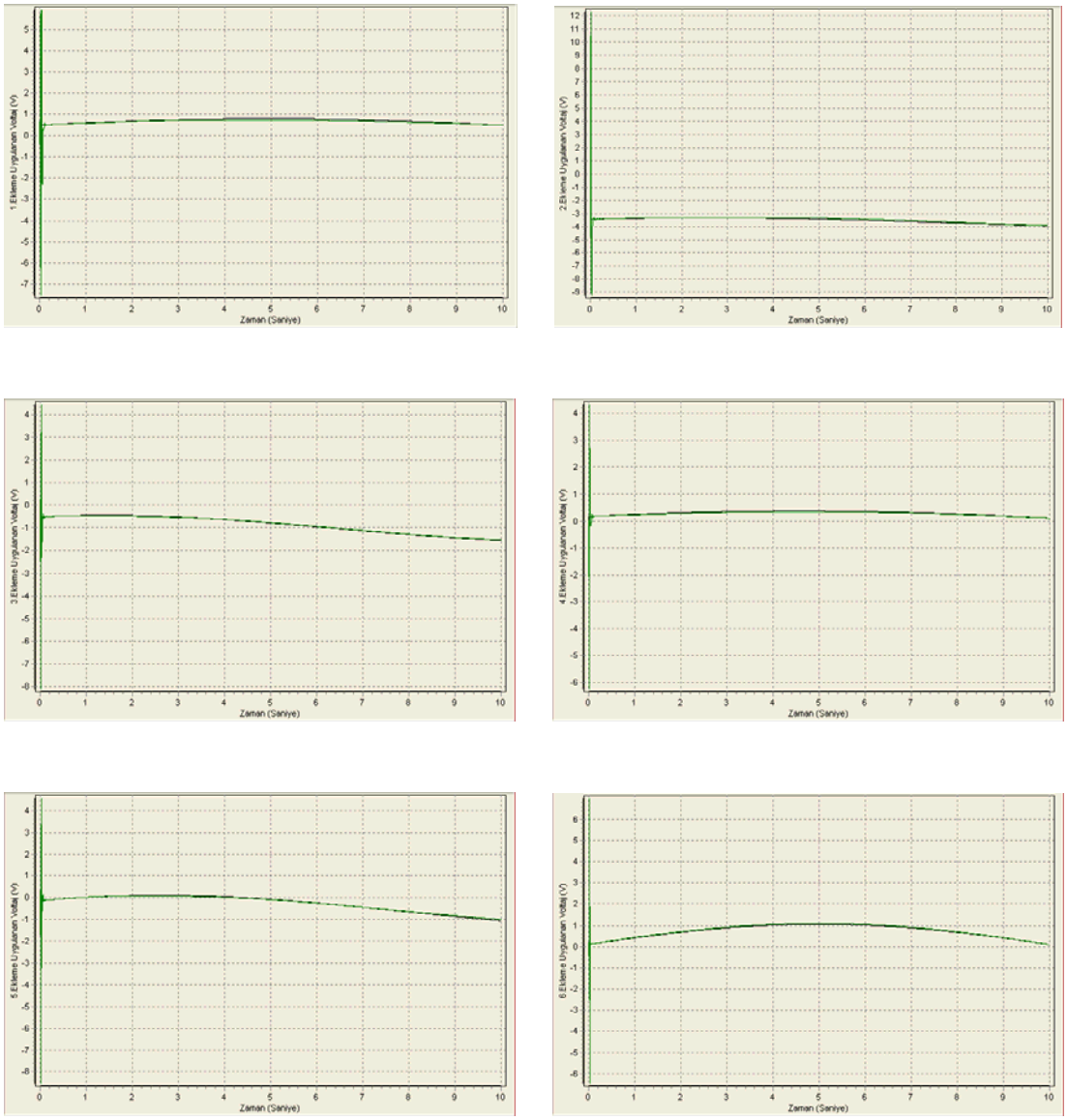

Şekil 4.53. Eklemlerin takip ettiği açısal yol grafikleri (GPC SISO, Örnek 1, Durum 2)



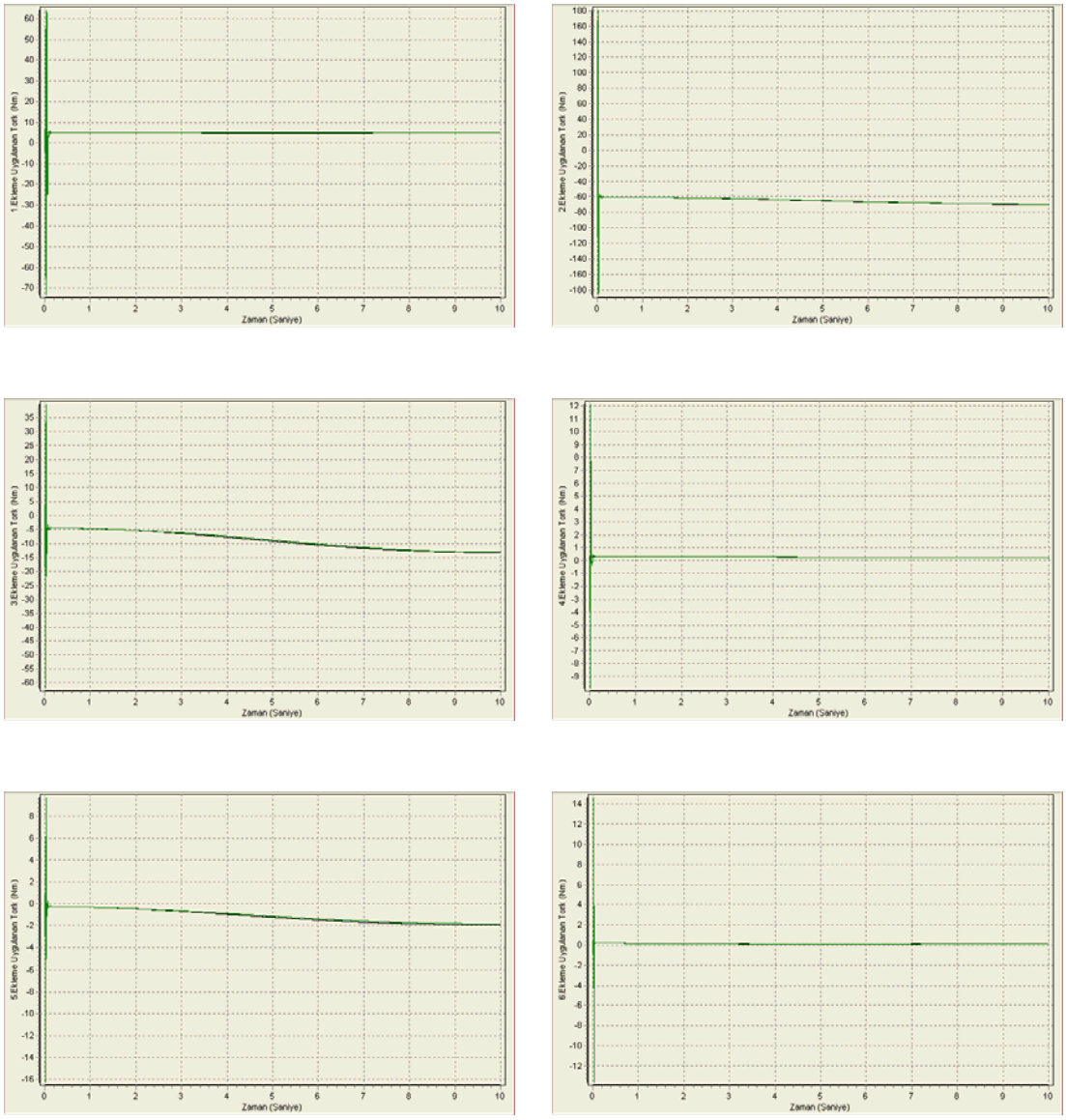

Şekil 4.54. Eklemlerin açısal hız grafikleri (GPC SISO, Örnek 1, Durum 2)


Şekil 4.55. Eklemlerin açısal hız hataları (GPC SISO, Örnek 1, Durum 2)

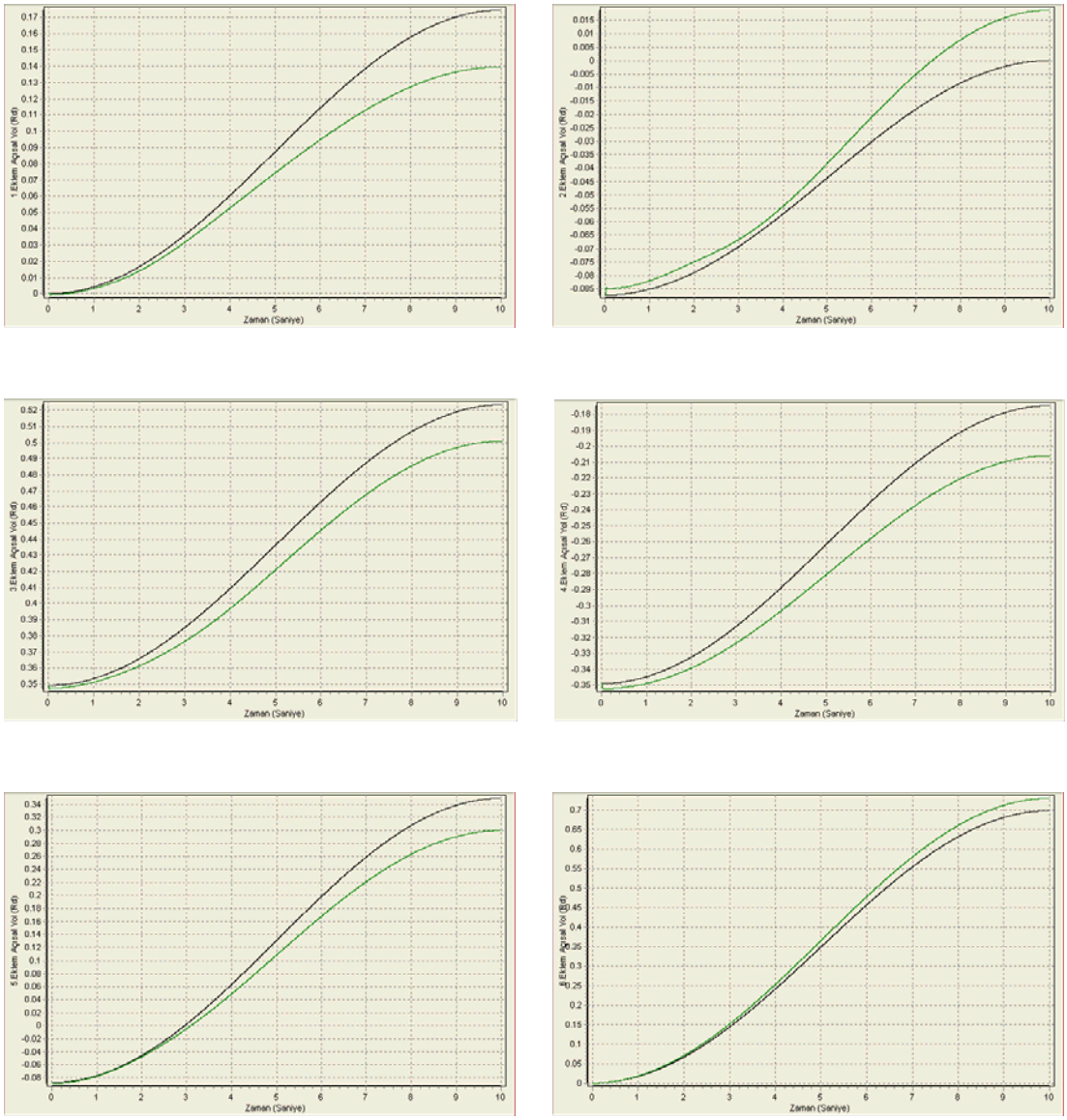
GPC SISO algoritması 2. Durum ‘da (sürtünme ve yük var) 1. Durum ‘da olduğu gibi başarılıdır. 2. Durum ‘da söylenenler burada da geçerlidir. Tork ve gerilim eğrileri düzgündür. Referans yörüngeler oldukça yakın takip edilmiştir ve açısal hız hataları sıfıra yakınsamıştır.



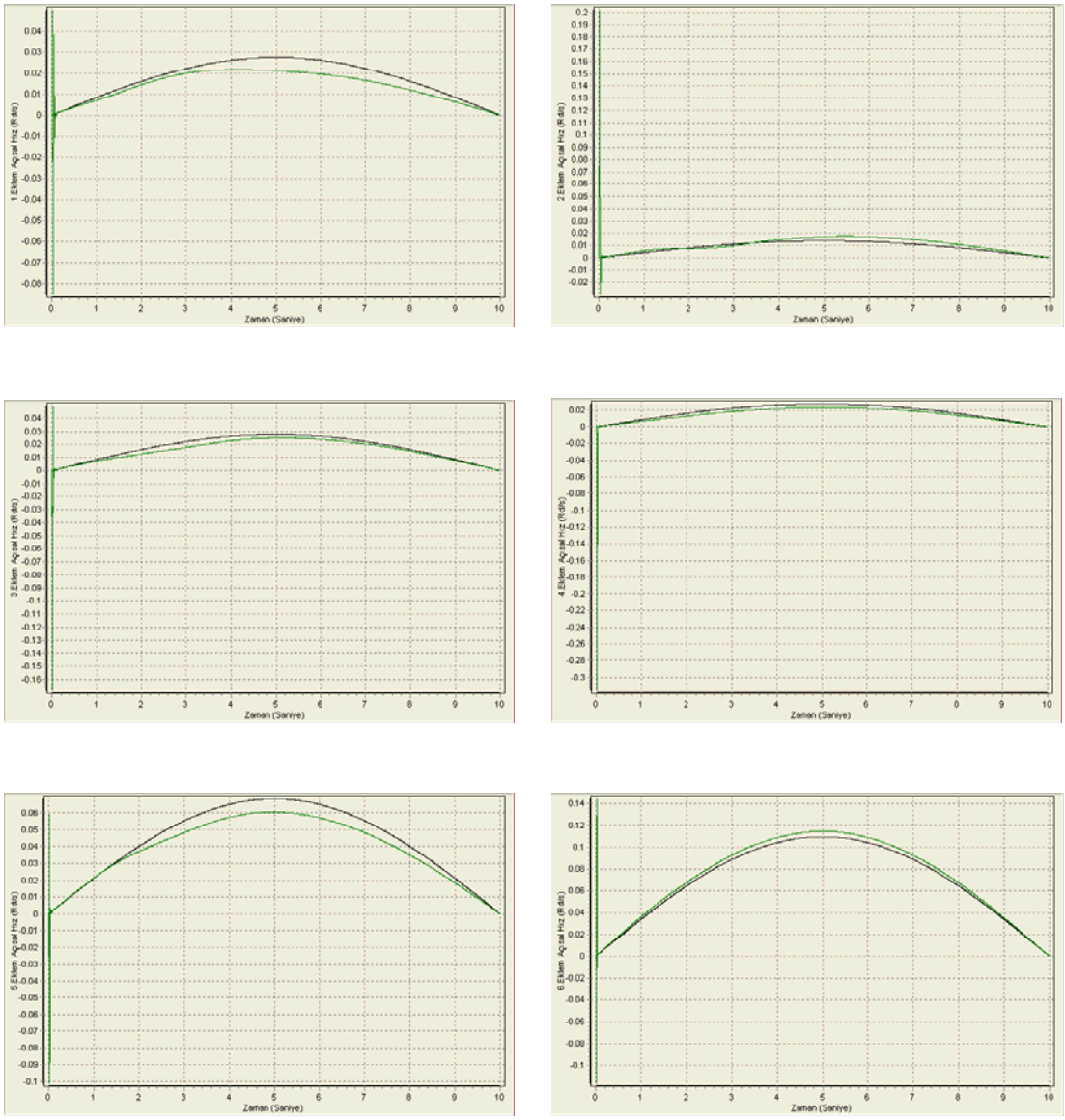

Şekil 4.56. Eklemlere uygulanan gerilim grafikleri (GPC MIMO, Örnek 1, Durum 2)



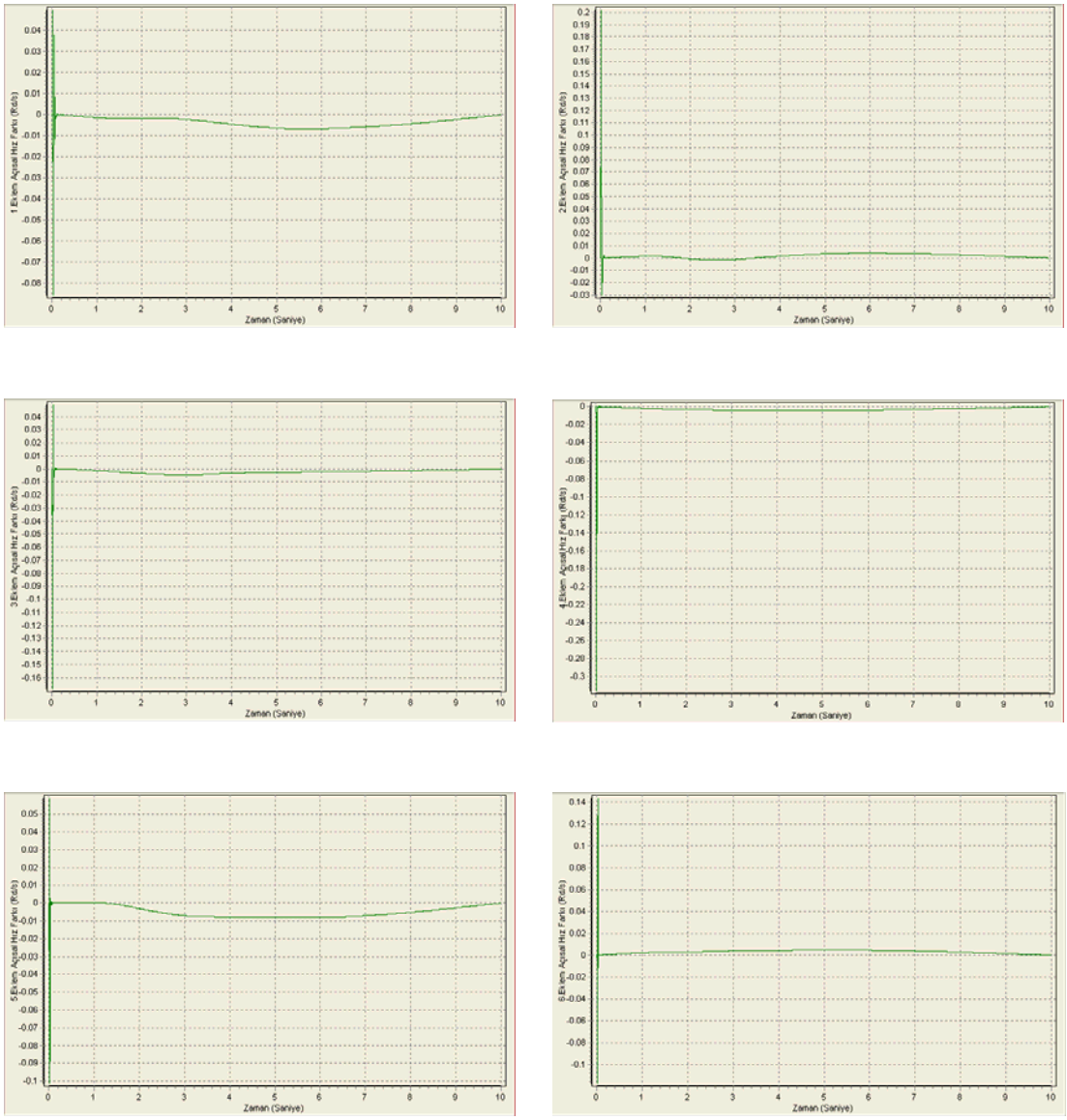

Şekil 4.57. Eklemlere uygulanan tork grafikleri (GPC MIMO, Örnek 1, Durum 2)




Şekil 4.58. Eklemlerin takip ettiği açısal yol grafikleri (GPC MIMO, Örnek 1, Durum 2)

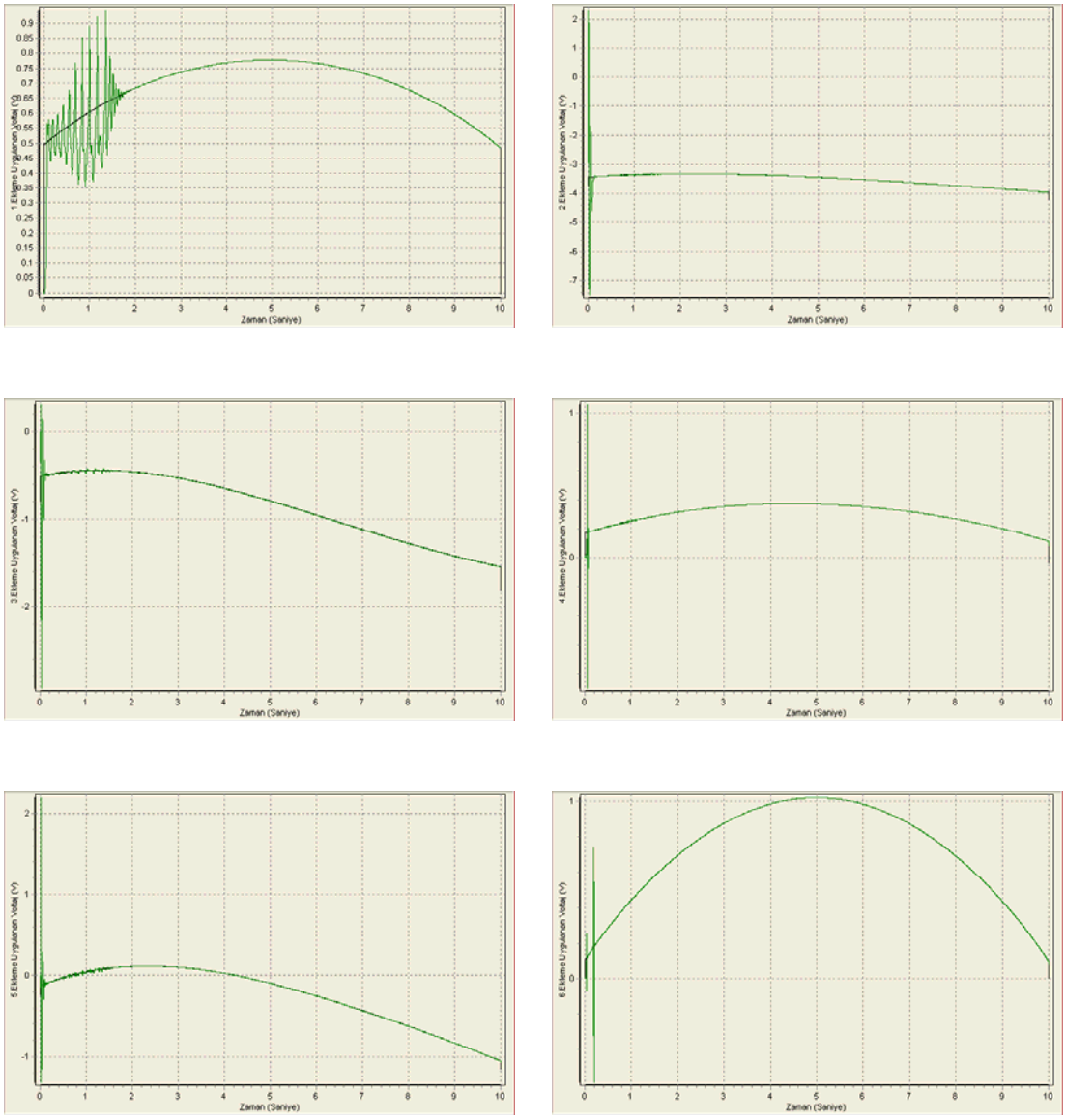



Şekil 4.59. Eklemlerin açısal hız grafikleri (GPC MIMO, Örnek 1, Durum 2)

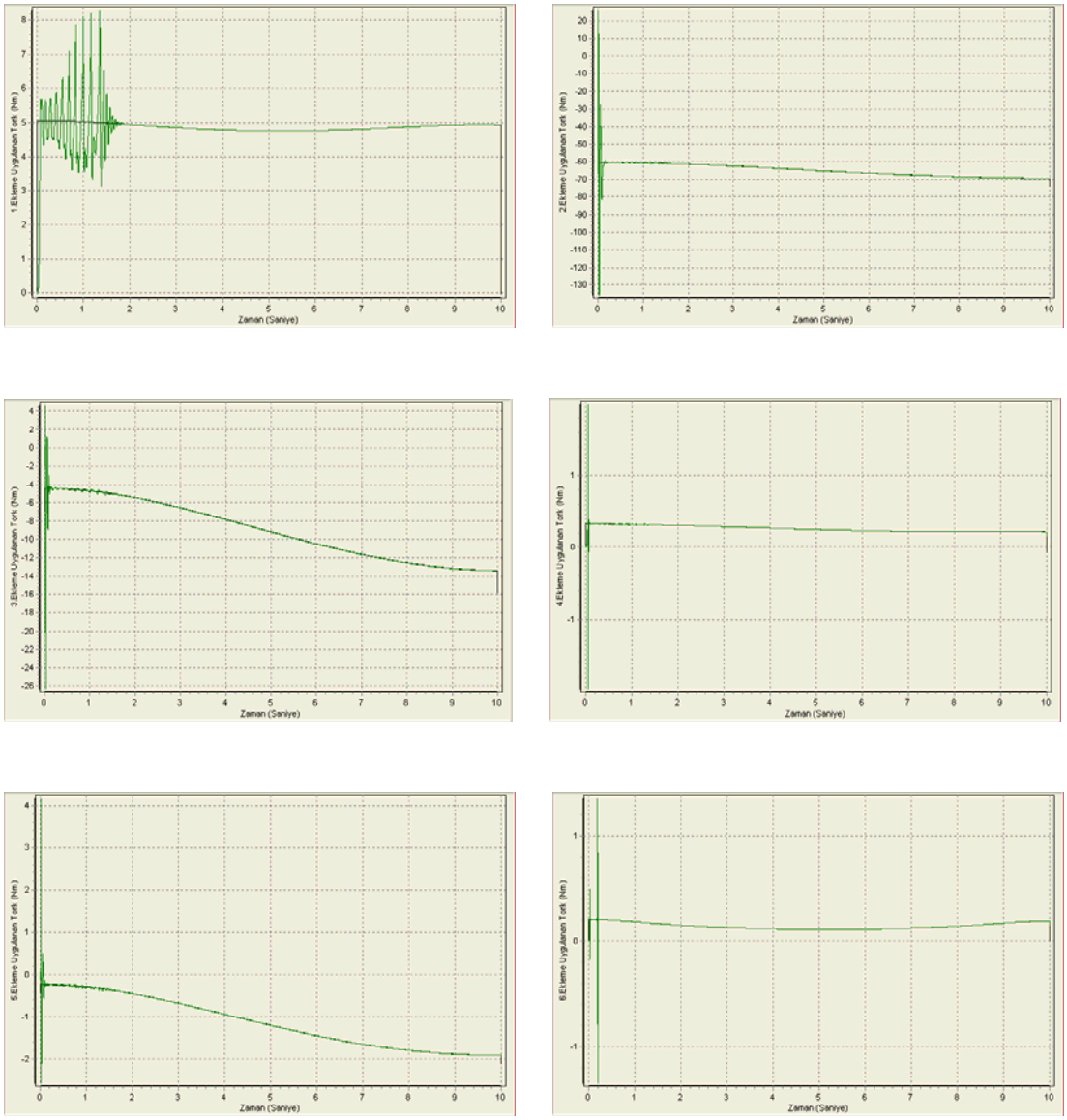



Şekil 4.60. Eklemlerin açısal hız hataları (GPC MIMO, Örnek 1, Durum 2)

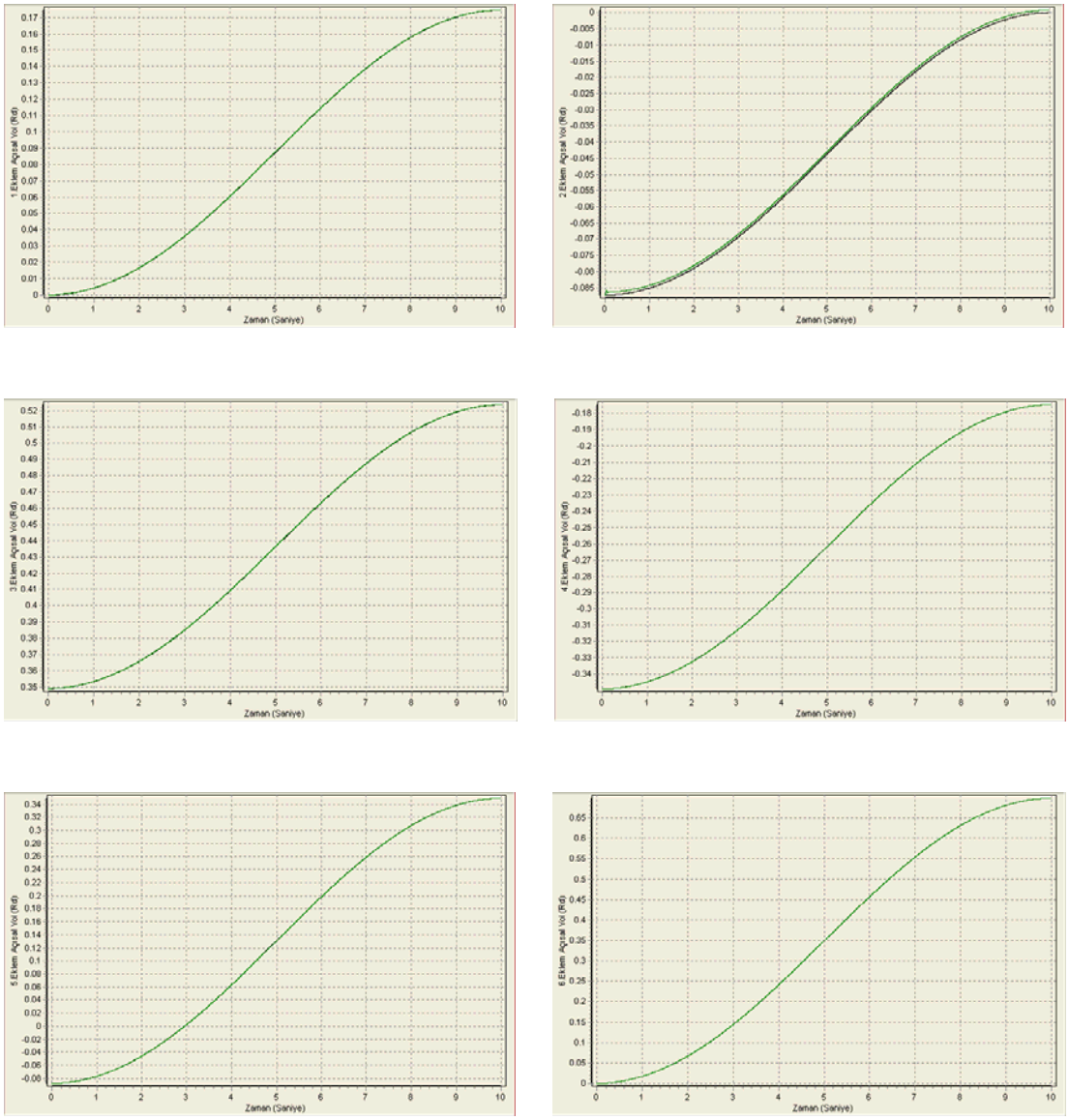
2. Durum için GPC MIMO algoritmasında referans yörüngelerden bir miktar sapmalar meydana gelmiş dolayısıyla bu fark açısal hızlara da yansımıştır ve referans hızlardan bir miktar sapmalar oluşmuştur.



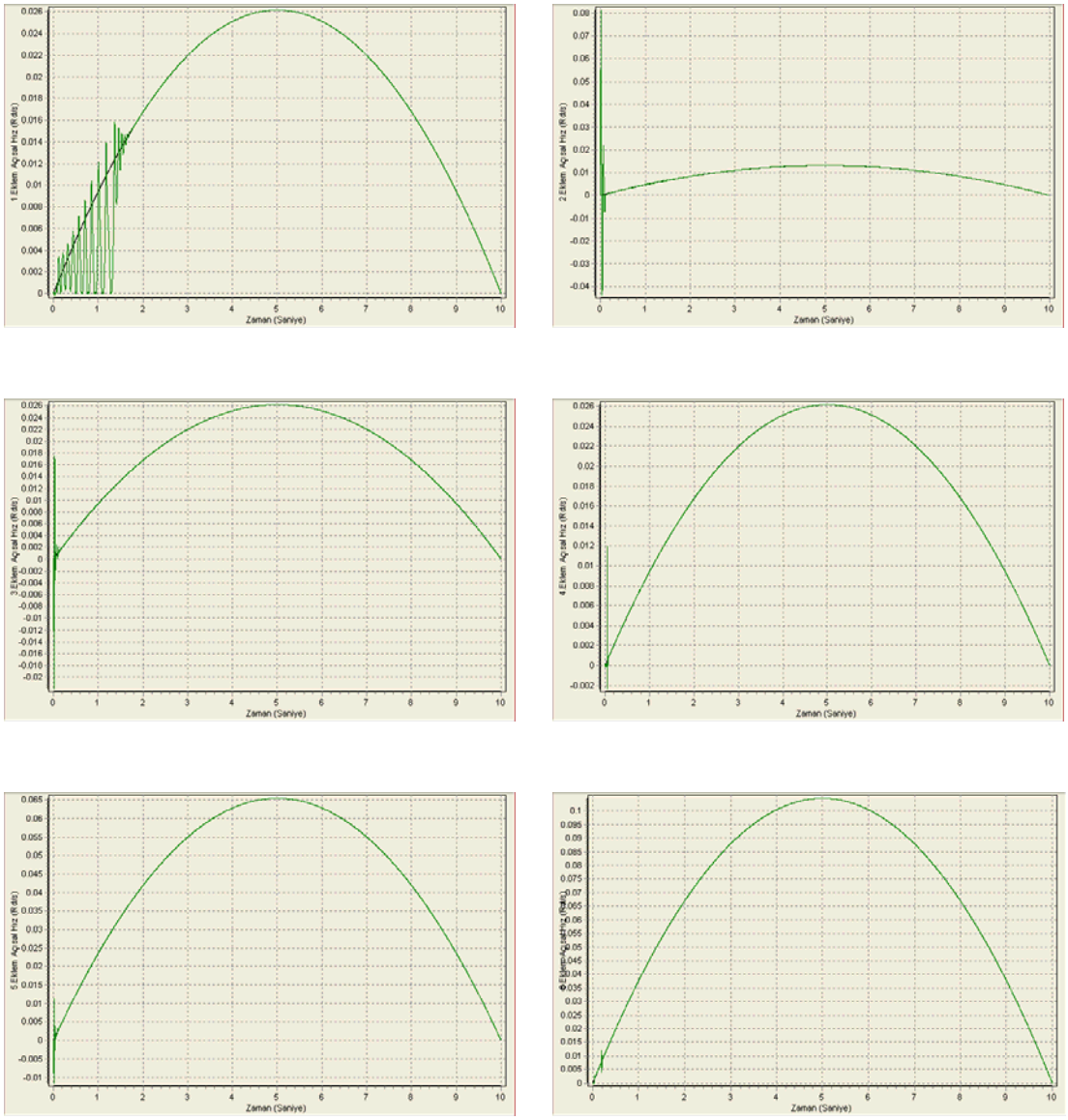

Şekil 4.61. Eklemlere uygulanan gerilim grafikleri (SGA-GPC SISO, Örnek 1, Durum 2)




Şekil 4.62. Eklemlere uygulanan tork grafikleri (SGA-GPC SISO, Örnek 1, Durum 2)



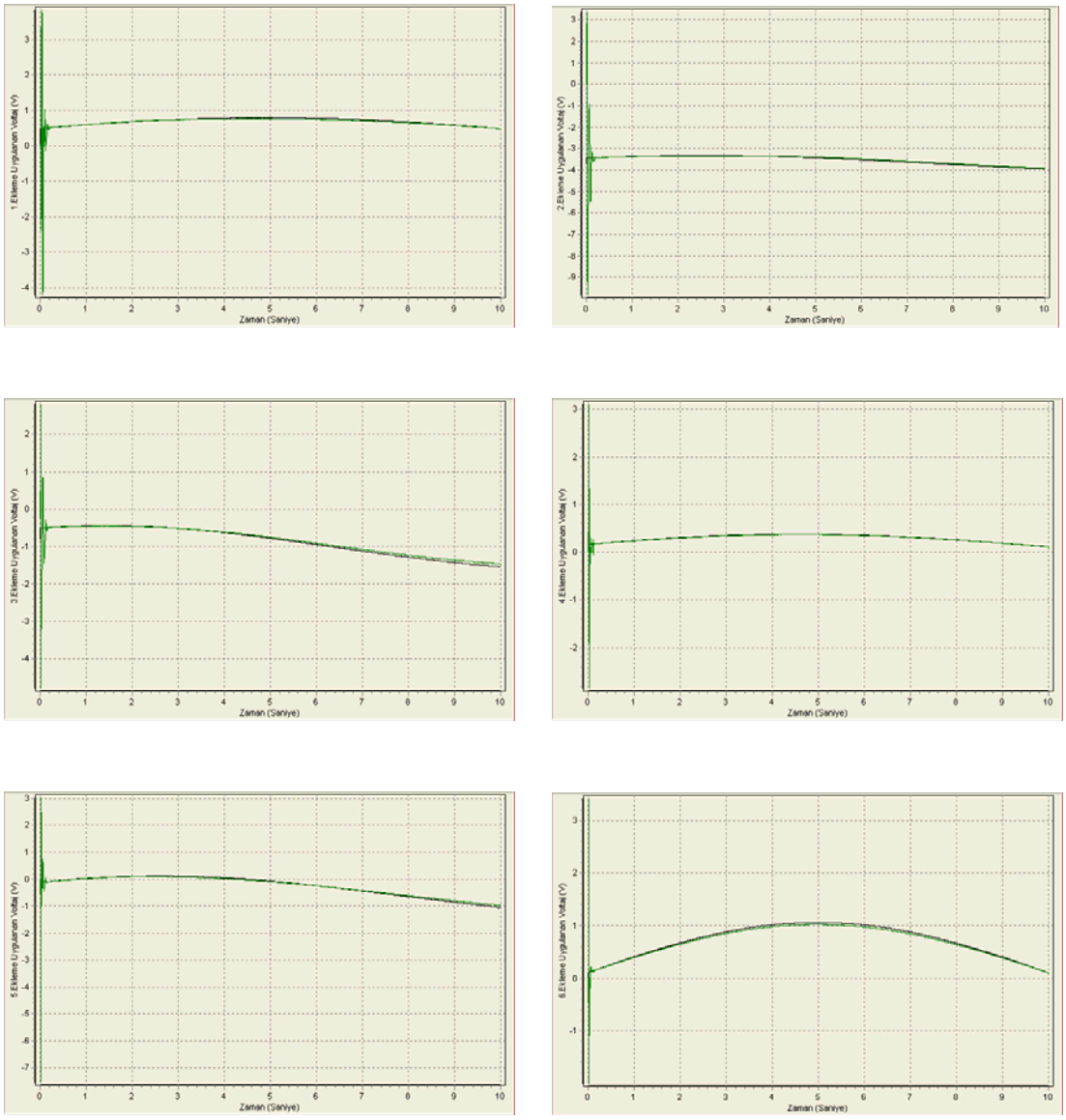

Şekil 4.63. Eklemlerin takip ettiği açısal yol grafikleri (SGA-GPC SISO, Örnek 1, Durum 2)



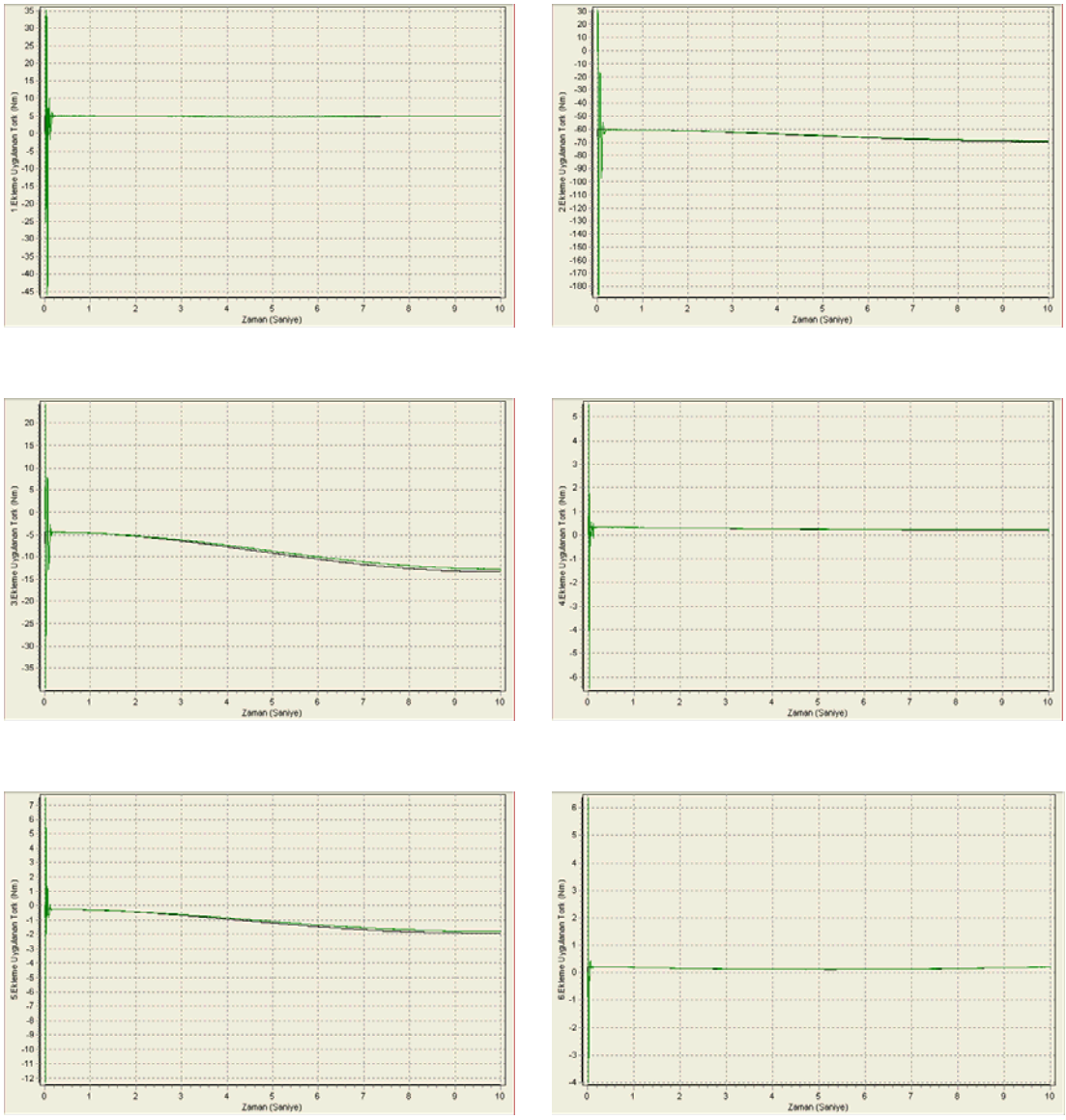

Şekil 4.64. Eklemlerin açısal hız grafikleri (SGA-GPC SISO, Örnek 1, Durum 2)


Şekil 4.65. Eklemlerin açısal hız hataları (SGA-GPC SISO, Örnek 1, Durum 2)

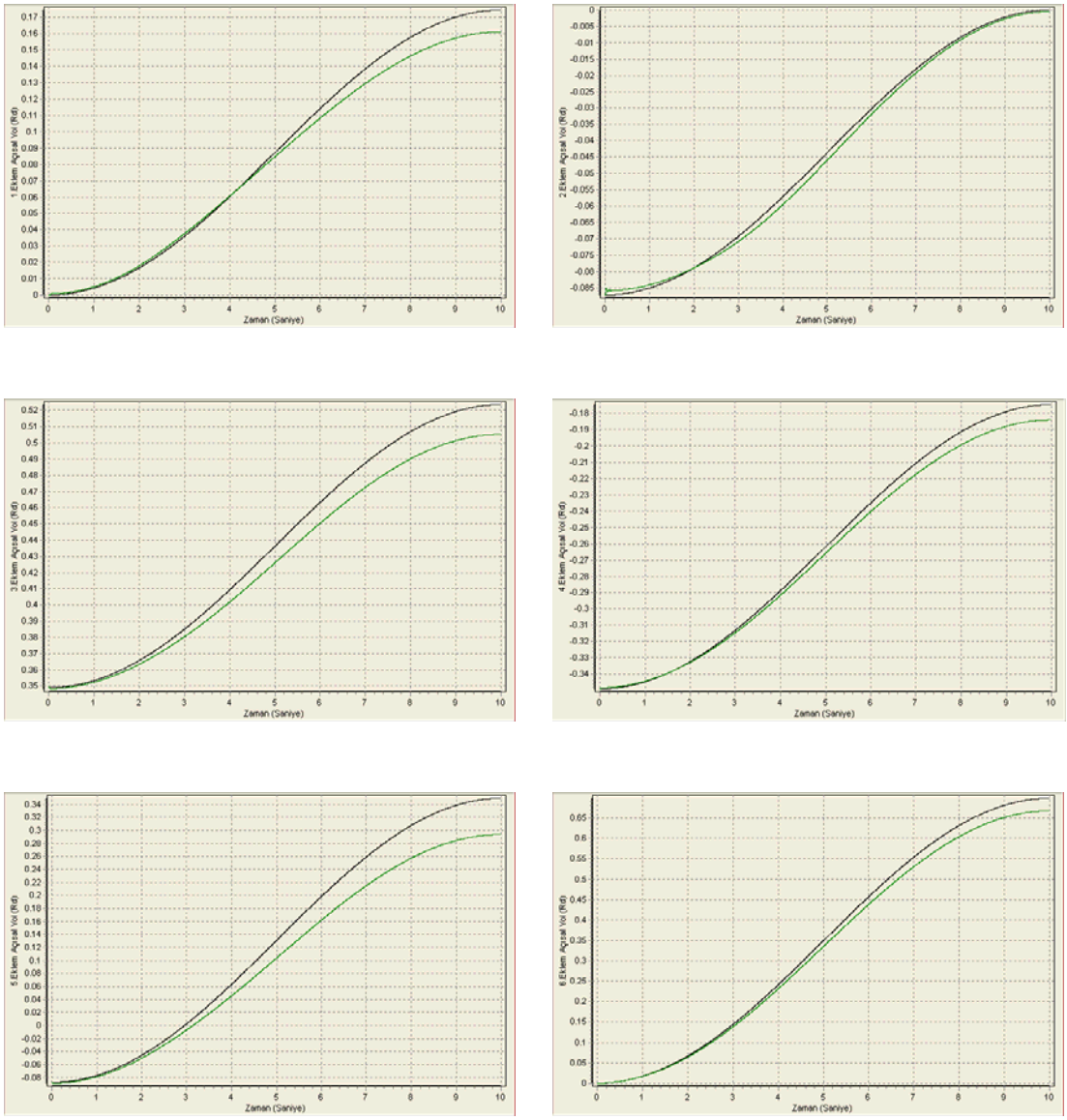
SGA-GPC SISO algoritması ile 2. Durum ‘da genel olarak başarılıdır. 1. Durum ‘da olduğu gibi hareket başlangıcında oluşan farklı genliklerdeki salınımlar, yörünge ve açısal hız hatalarına sebebiyet vermiştir. Salınımların doğurduğu bu sapmalar karşısında kontrolör, durumu dengelemek için referans yörüngeleri yakalayacak uygun tork değerlerini üretmeye çalışmaktadır. Yaklaşık 2 saniye sonunda referans yörüngeler yakalanmış ve hatalar minimize edilmiştir.



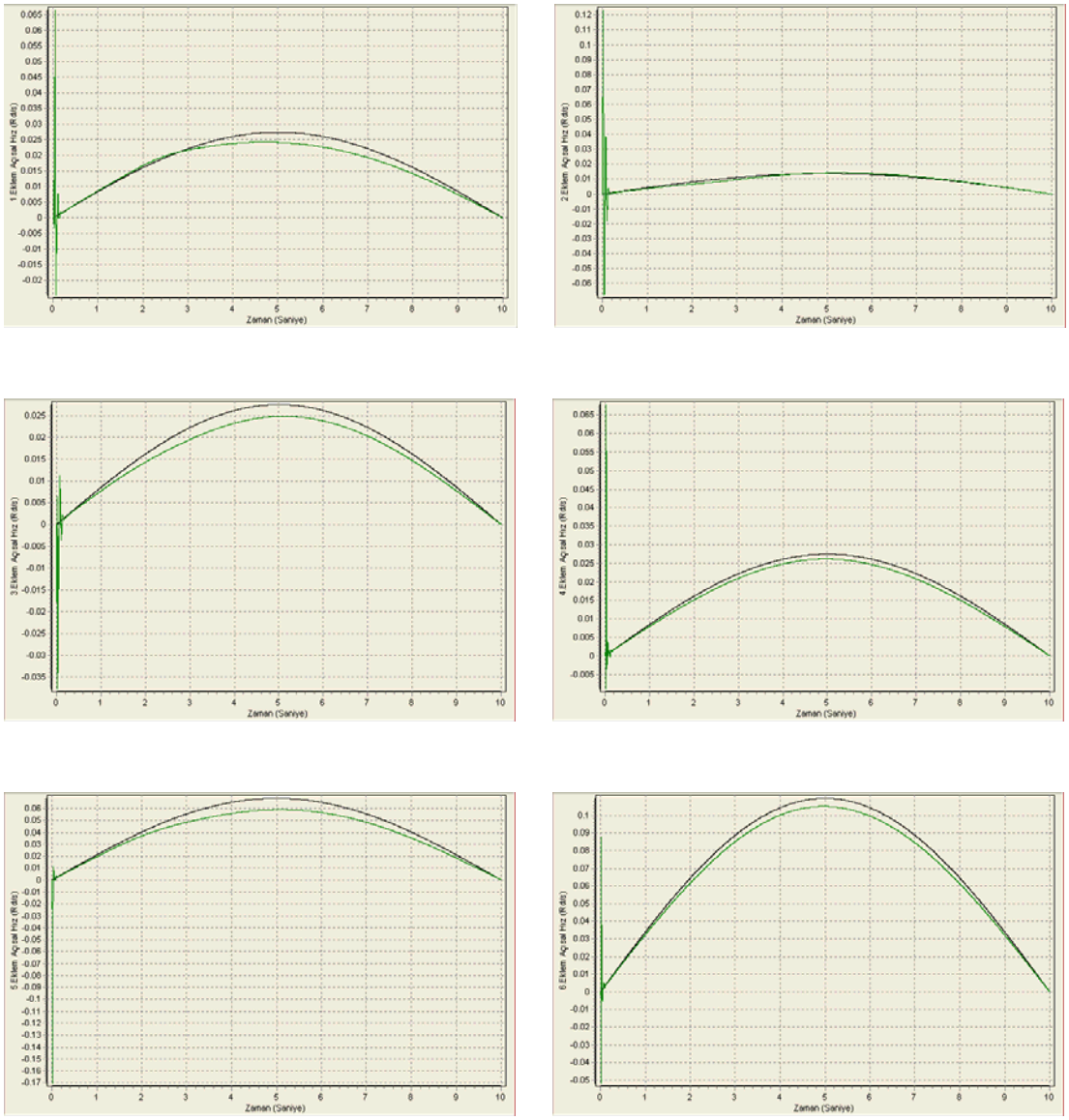

Şekil 4.66. Eklemlere uygulanan gerilim grafikleri (SGA-GPC MIMO, Örnek 1, Durum 2)



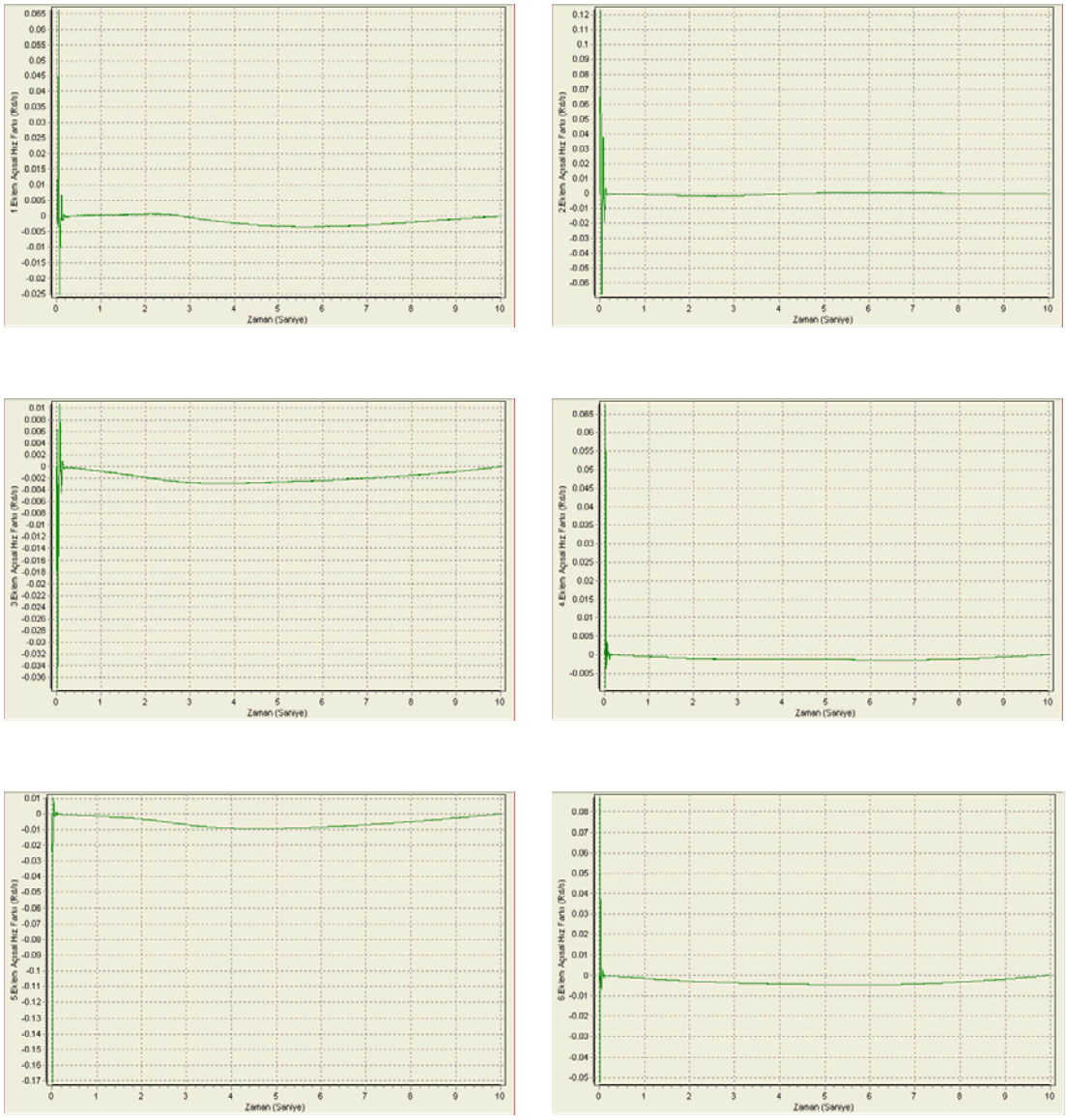

Şekil 4.67. Eklemlere uygulanan tork grafikleri (SGA-GPC MIMO, Örnek 1, Durum 2)




Şekil 4.68. Eklemlerin takip ettiği açısal yol grafikleri (SGA-GPC MIMO, Örnek 1, Durum 2)

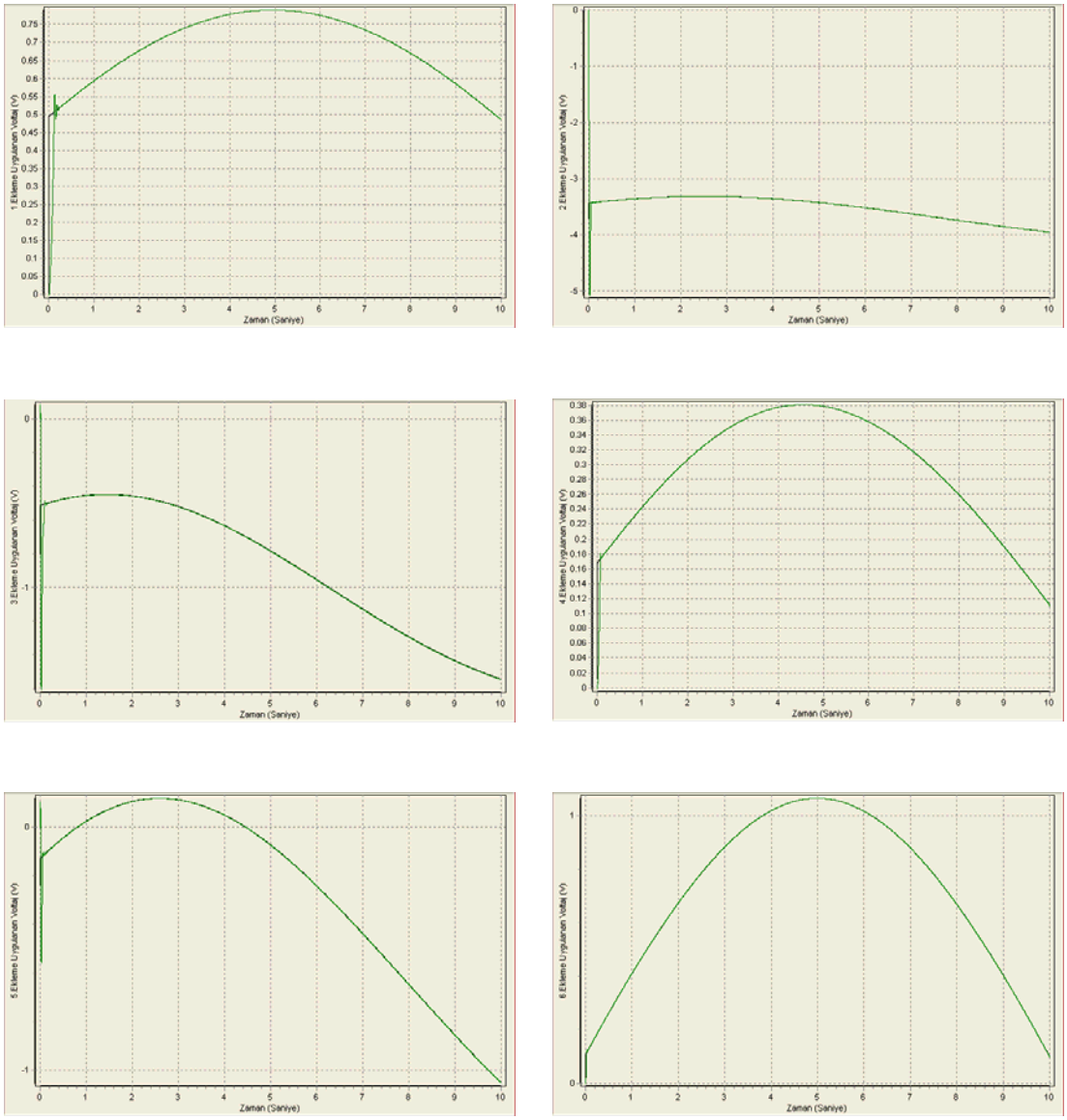



Şekil 4.69. Eklemlerin açısal hız grafikleri (SGA-GPC MIMO, Örnek 1, Durum 2)

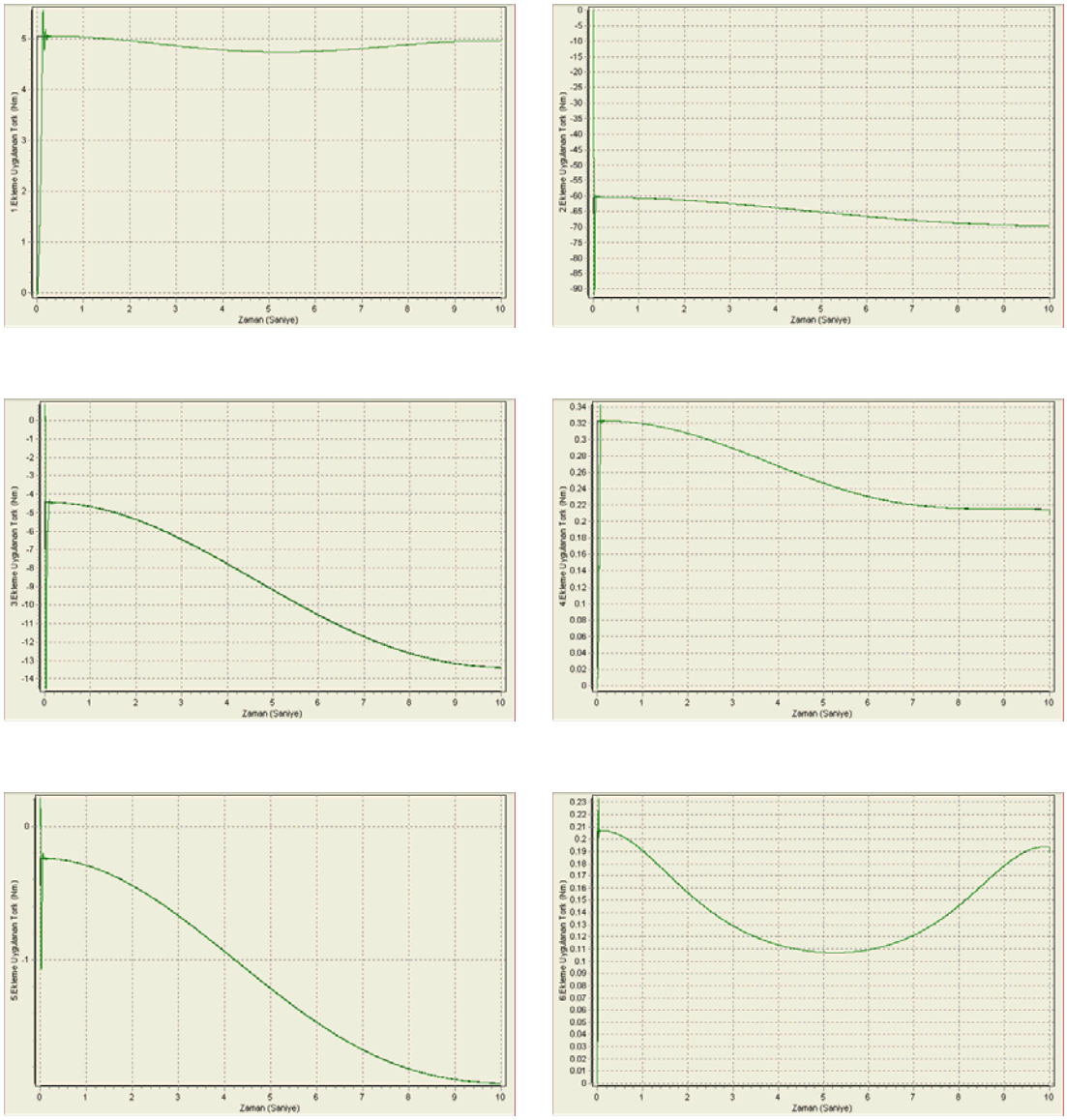



Şekil 4.70. Eklemlerin açısal hız hataları (SGA-GPC MIMO, Örnek 1, Durum 2)

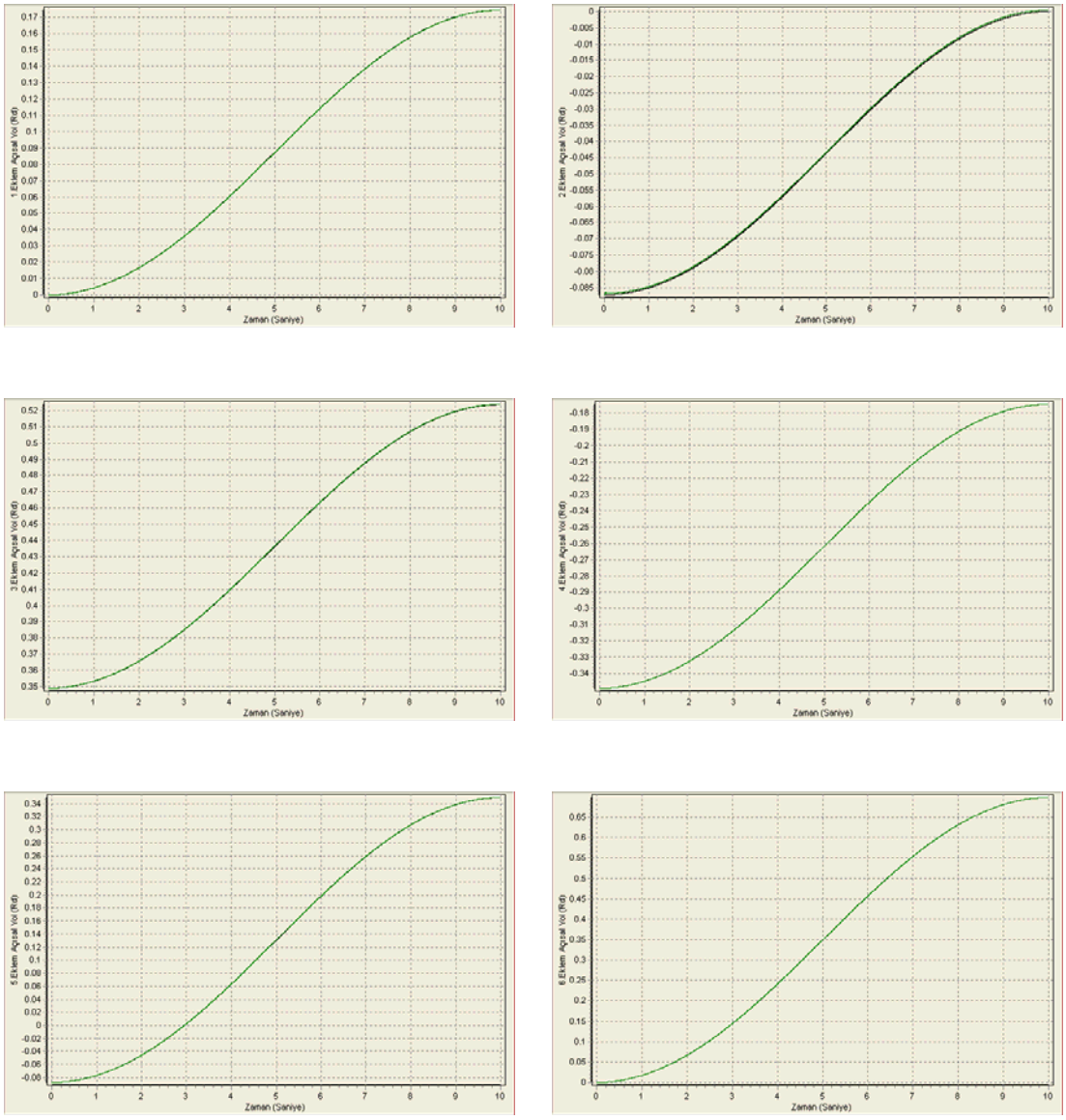
SGA-GPC MIMO algoritması 2. Durum ‘da da gayet düzgün gerilim ve tork eğrilerine sahiptir. Ancak yörünge ve açısal hız grafiklerinde görüldüğü gibi referans eğriler ile farklar oluşmuştur.



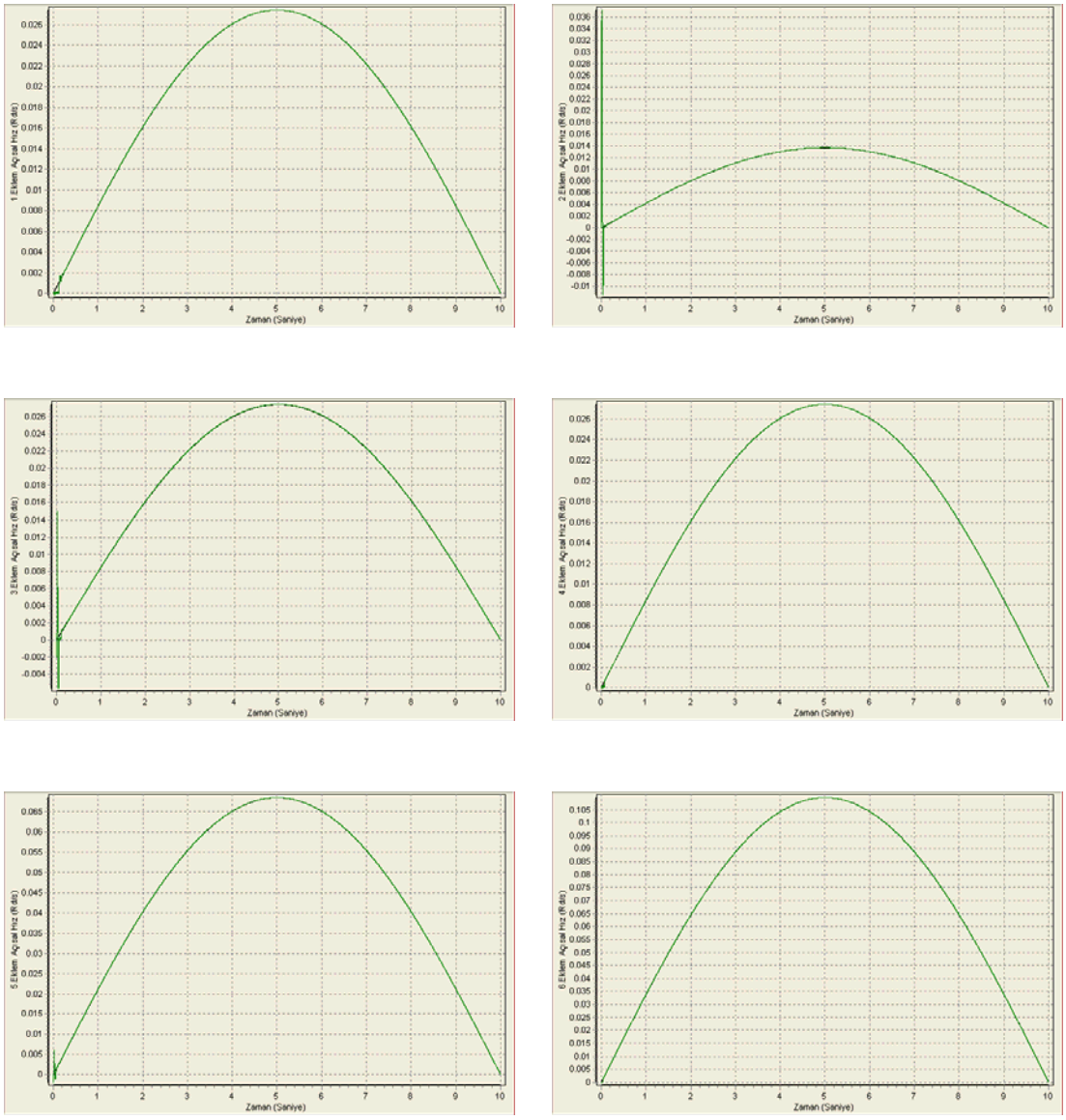

Şekil 4.71. Eklemlere uygulanan gerilim grafikleri (NGPC SISO, Örnek 1, Durum 2)




Şekil 4.72. Eklemlere uygulanan tork grafikleri (NGPC SISO, Örnek 1, Durum 2)



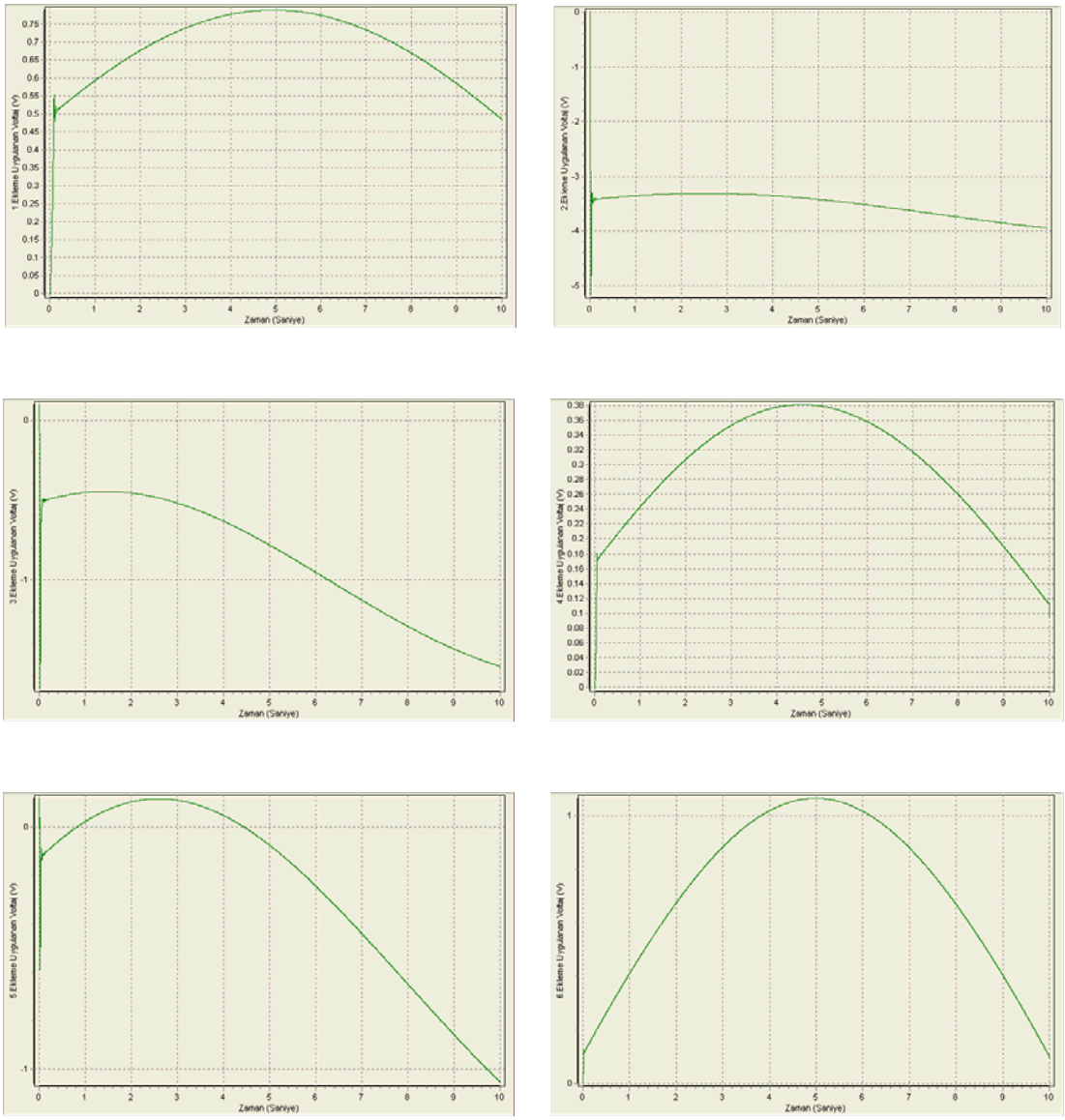

Şekil 4.73. Eklemlerin takip ettiği açısal yol grafikleri (NGPC SISO, Örnek 1, Durum 2)



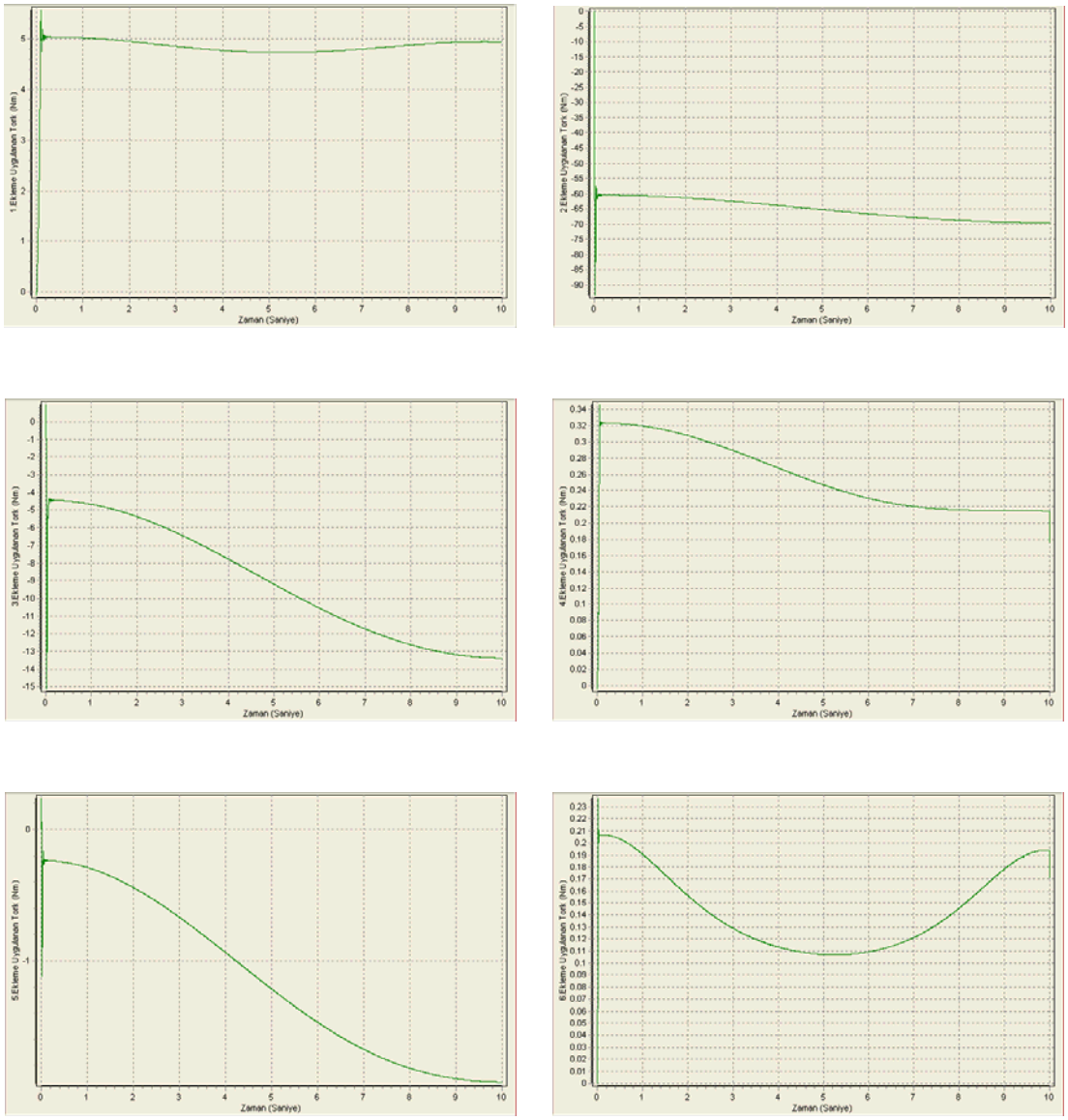

Şekil 4.74. Eklemlerin açısal hız grafikleri (NGPC SISO, Örnek 1, Durum 2)


Şekil 4.75. Eklemlerin açısal hız hataları (NGPC SISO, Örnek 1, Durum 2)

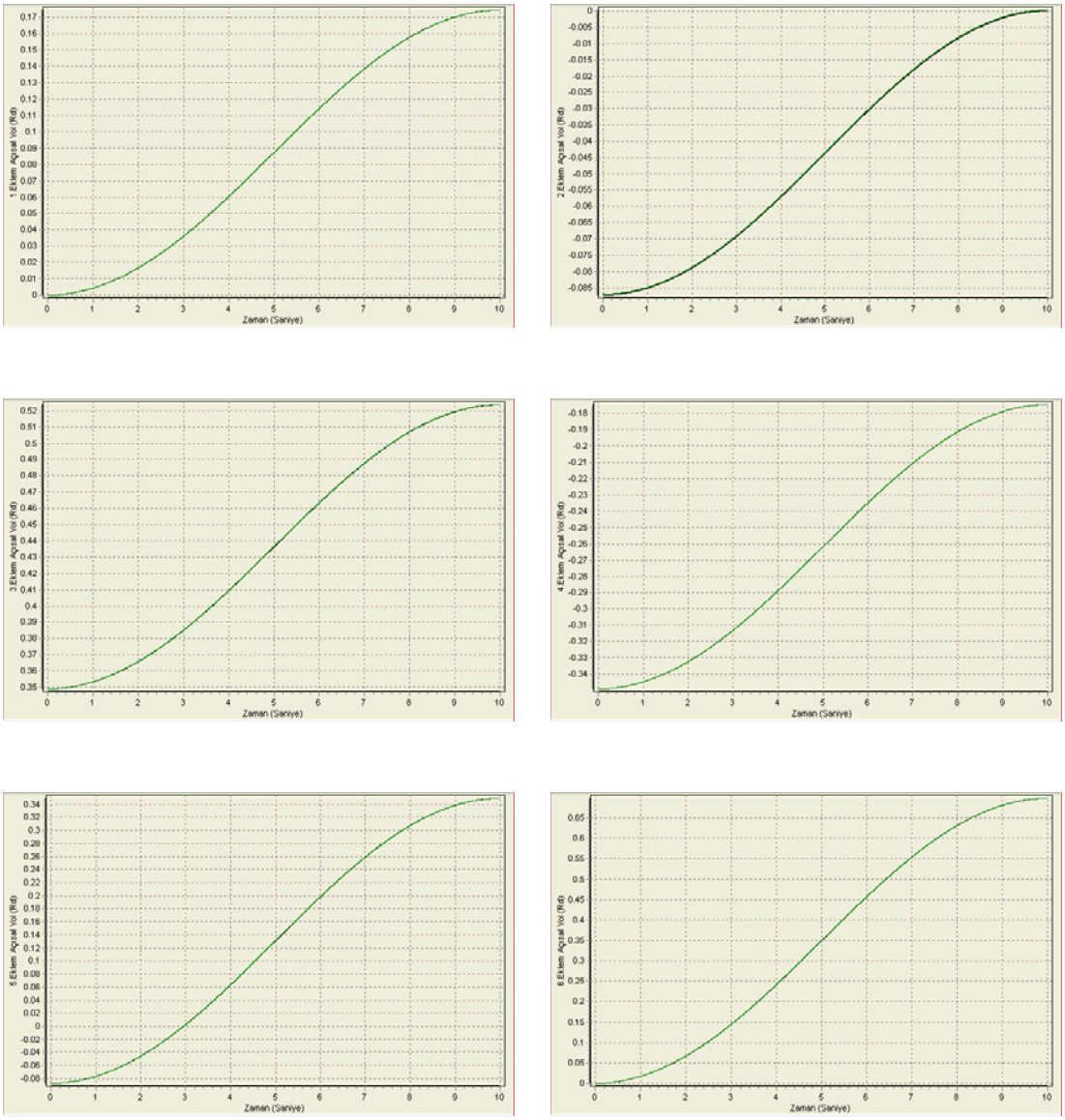
NGPC SISO algoritması 1. Durum ‘da olduğu gibi 2. Durum ‘da da başarılıdır. Tork ve gerilim eğrileri gayet düzgündür. Referans yörüngeler çok yakın takip edilmiştir. Ve açısal hız hataları sıfıra yakınsamıştır.



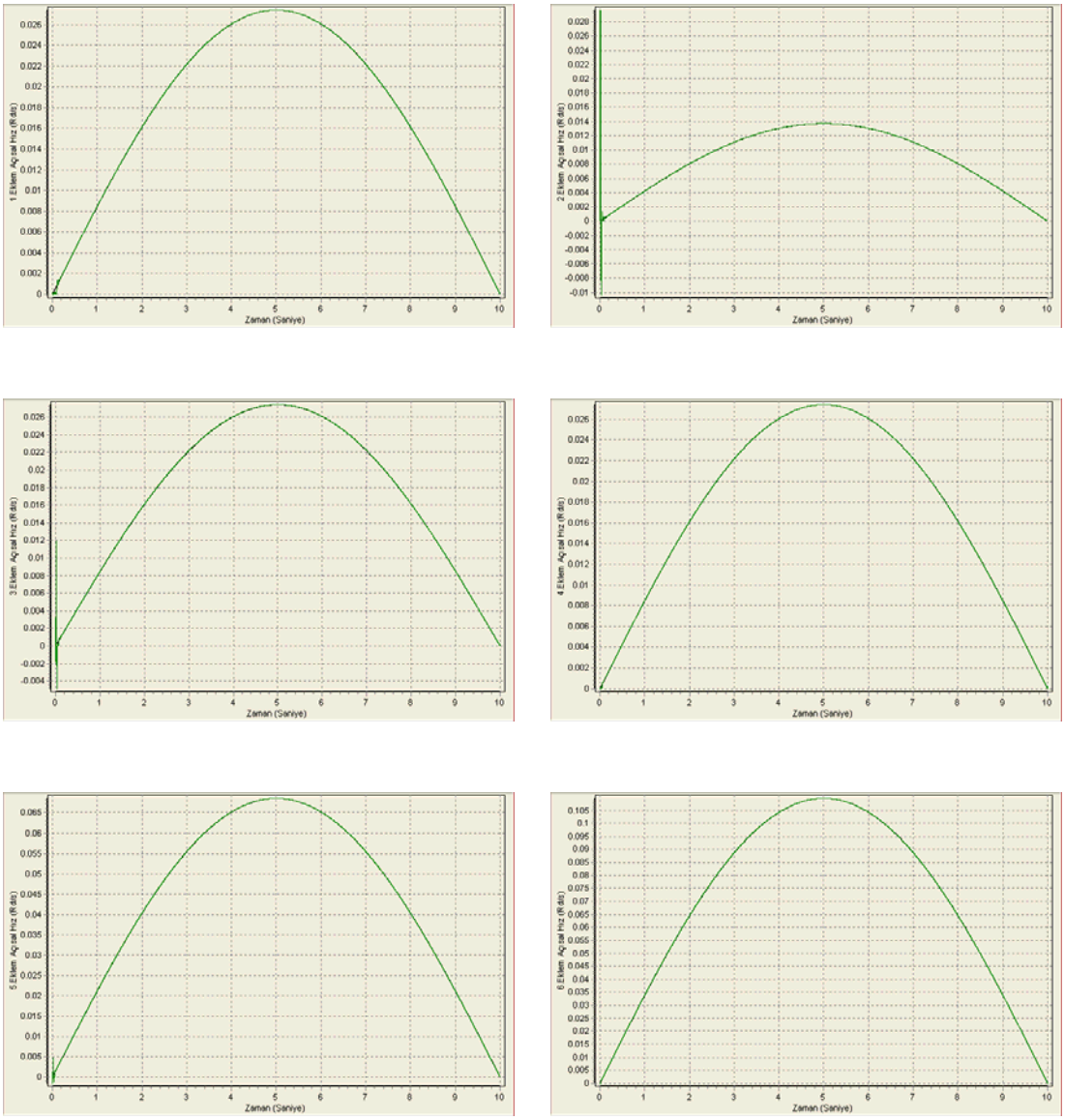

Şekil 4.76. Eklemlere uygulanan gerilim grafikleri (ENGPC SISO, Örnek 1, Durum 2)




Şekil 4.77. Eklemlere uygulanan tork grafikleri (ENGPC SISO, Örnek 1, Durum 2)




Şekil 4.78. Eklemlerin takip ettiği açısal yol grafikleri (ENGPC SISO, Örnek 1, Durum 2)




Şekil 4.79. Eklemlerin açısal hız grafikleri (ENGPC SISO, Örnek 1, Durum 2)

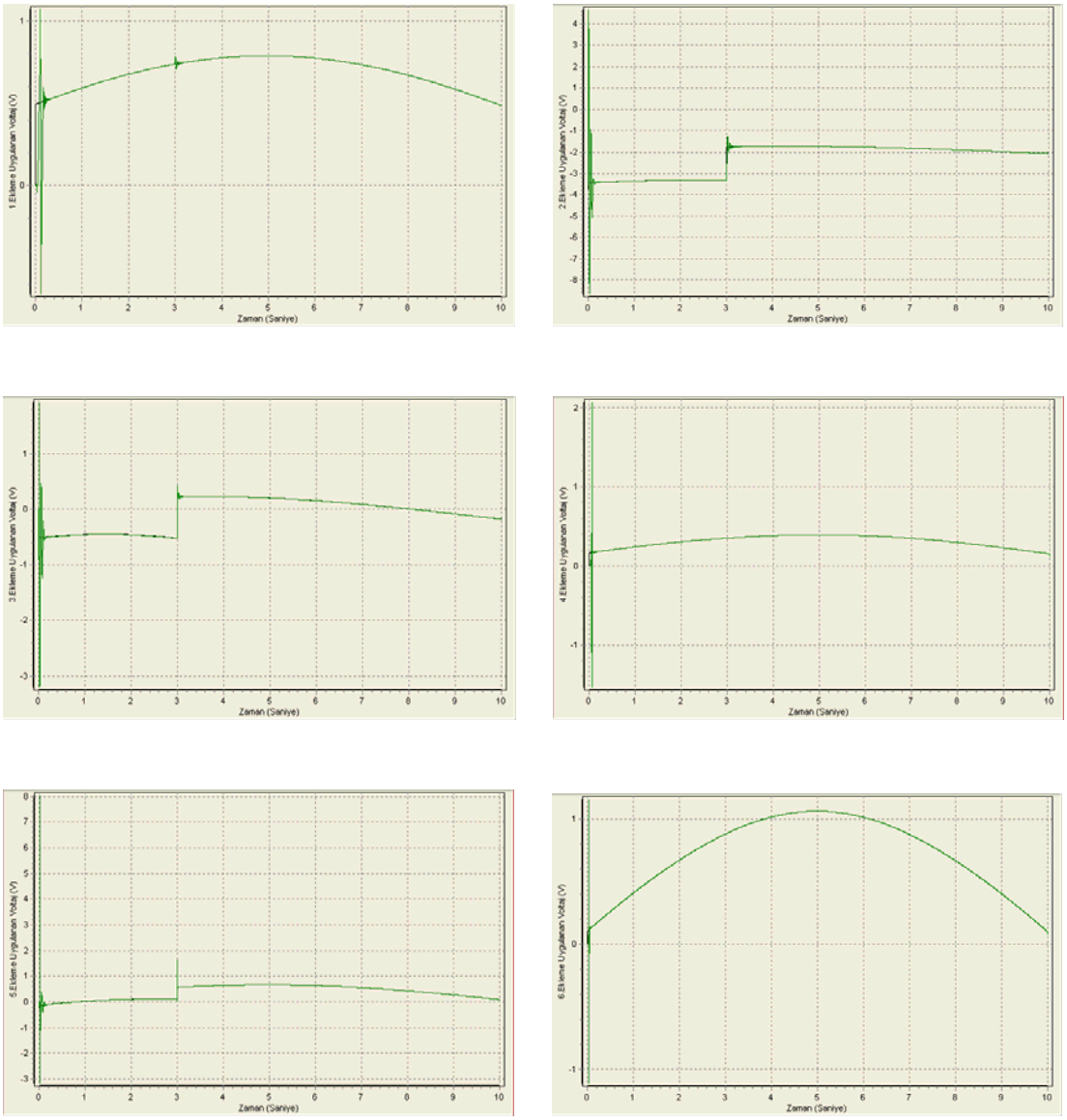

Şekil 4.80. Eklemlerin açısal hız hataları (ENGPC SISO, Örnek 1, Durum 2)

ENGPC SISO algoritması 1. Durum ‘da olduğu gibi 2. Durum ‘da da en başarılı algoritma olmuştur. Tork ve gerilim eğrileri gayet düzgündür. Referans yörüngeler çok yakın takip edilmiştir. Ve açısal hız hataları sıfıra yakınsamıştır.

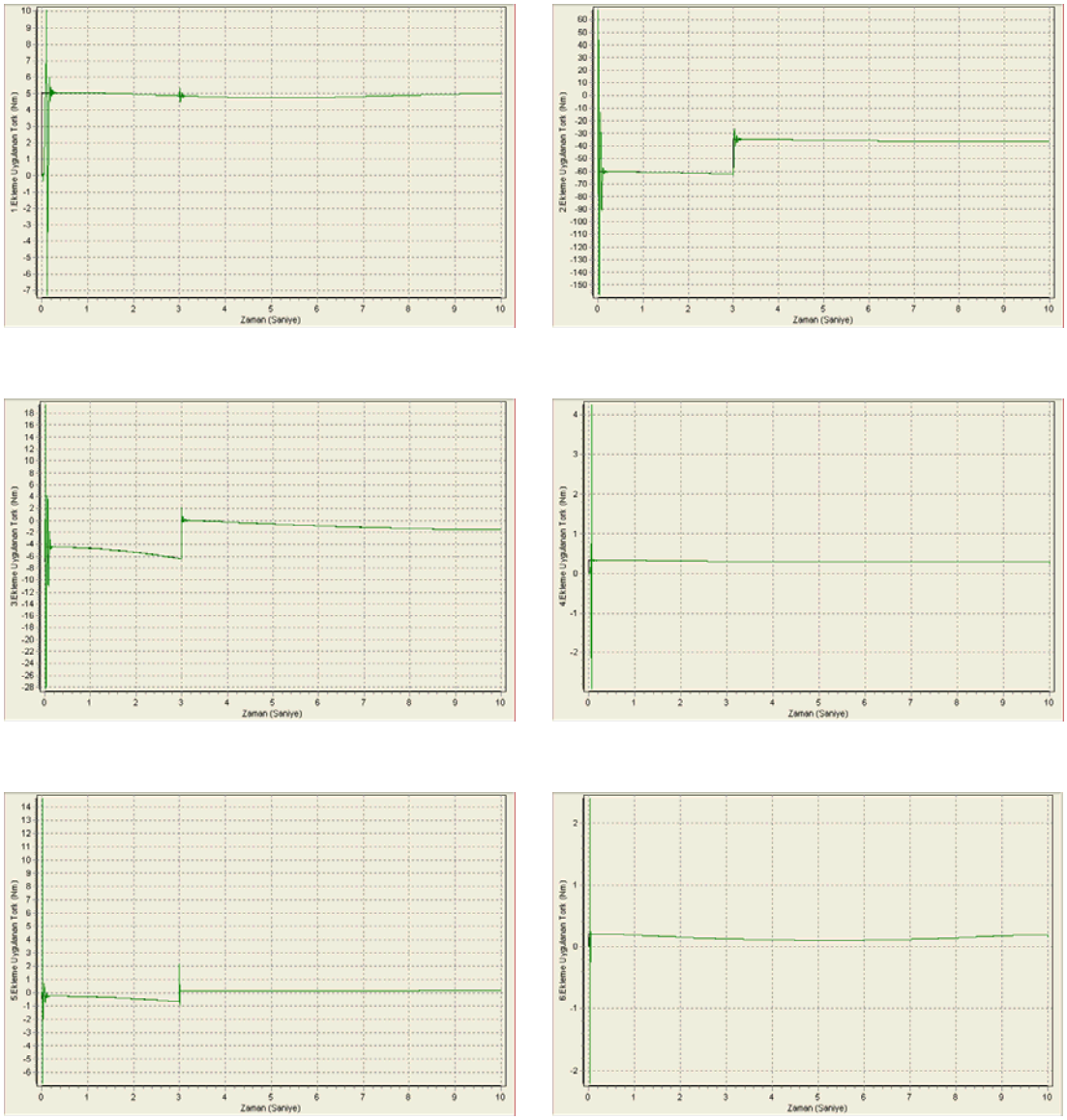
2. Durum için kontrol algoritmaları genel olarak başarılıdır. Sürtünme ve yükün getirdiği ilave zorluk karşısında özellikle SISO yapıdaki algoritmalar referans yörüngeleri çok yakın takip etmişlerdir. Bunun sonucu eklemler çok az açısal hız hataları ile hareketlerini tamamlamışlardır. SGA-GPC algoritmaları hareket başlangıcındaki salınımlar karşısında zorlanmakla birlikte 1-2 saniye içerisinde

durumu dengeleyebilmektedirler. Bu durum, GA ‘nın adaptif özelliğinin diğer algoritmalara göre zayıf olduğu şeklinde yorumlanabilir. MIMO yapılardaki algoritmalar aşağı yukarı referans yörüngeleri takip etmekle birlikte aralarındaki farklar grafiklerde açıkça görülmektedir.

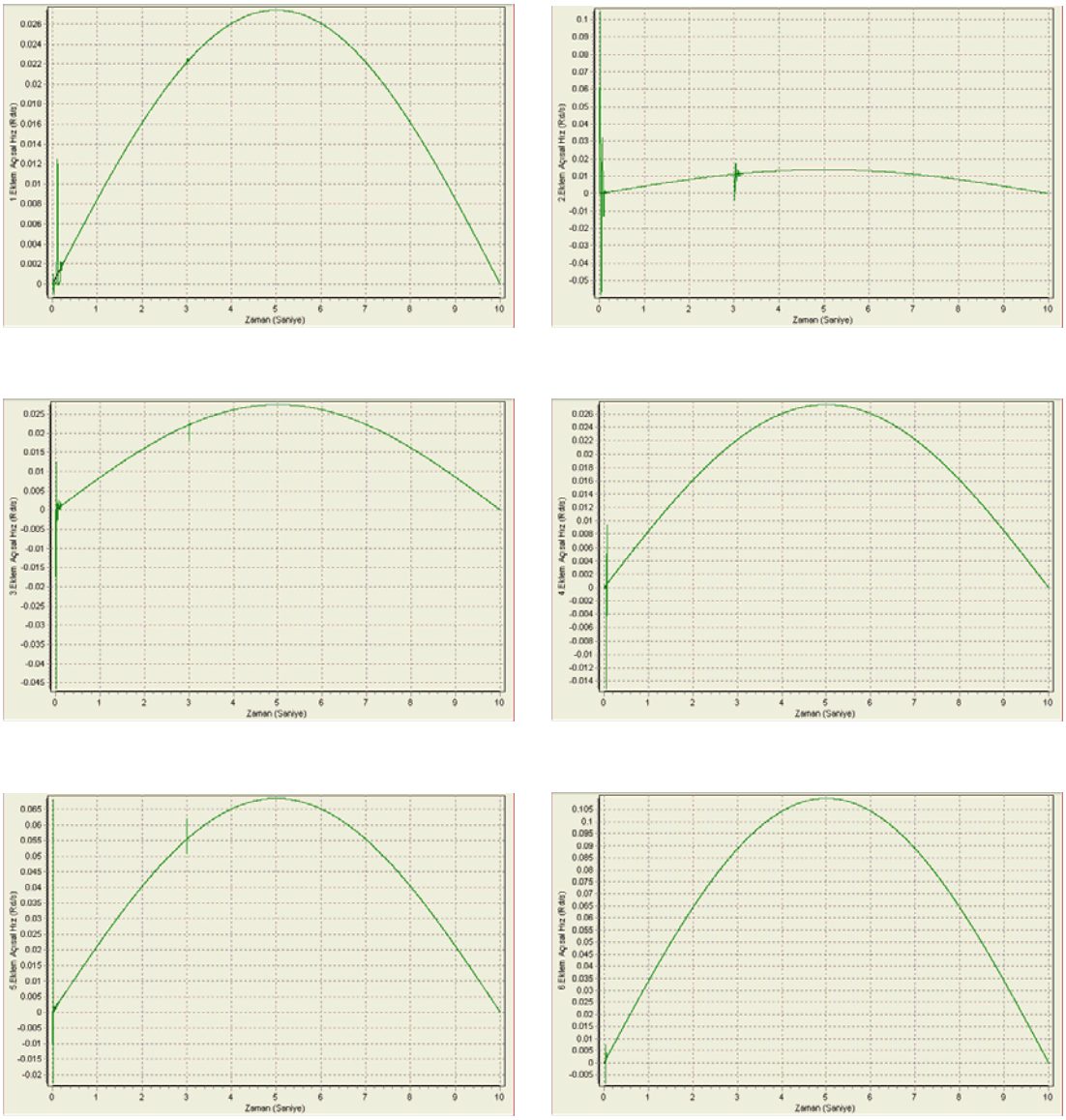
3. Durum ‘da ilave edilen yük düşme durumunda kontrolün 3. sn de yükün düştüğü varsayılmaktadır. Robot kolu bu andan itibaren yüksüz olarak hareketine devam edecektir. Yükün düştüğü anda yeni dinamikler karşısında kontrol algoritmalarının tepkileri incelenmiştir. Yük düşmesi robot kolunun dinamik modelindeki durumu değiştireceğinden oluşan yeni dinamikler karşısında algoritmaların adaptif özelliklerinin sınanması amaçlanmıştır. Bu amaçla 2. eklemin kontrolü diğer eklemlere nazaran daha zor olduğundan bu ekleme ait açısal hız hataları grafikleri daha ayrıntılı şekilde incelenmiş ve yük düşme anında verilen tepkiler gözlemlenmiştir. Oluşan salınımların genlikleri, sayıları ve sistem oturma zamanları ölçülmüştür. 3. Durum ‘da elde edilen simülasyon sonuçları Şekil 4.81 ‘den Şekil 4. 110 ‘a kadar aşağıda verilmiştir.



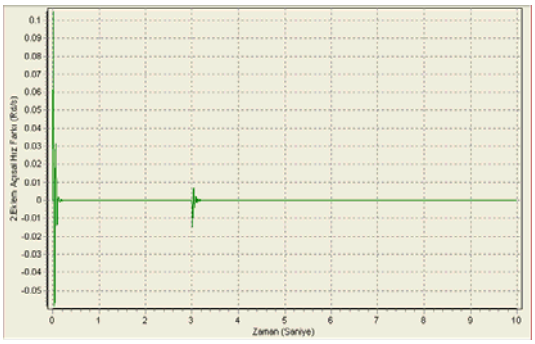

Şekil 4.81. Eklemlere uygulanan gerilim grafikleri (GPC SISO, Örnek 1, Durum 3)



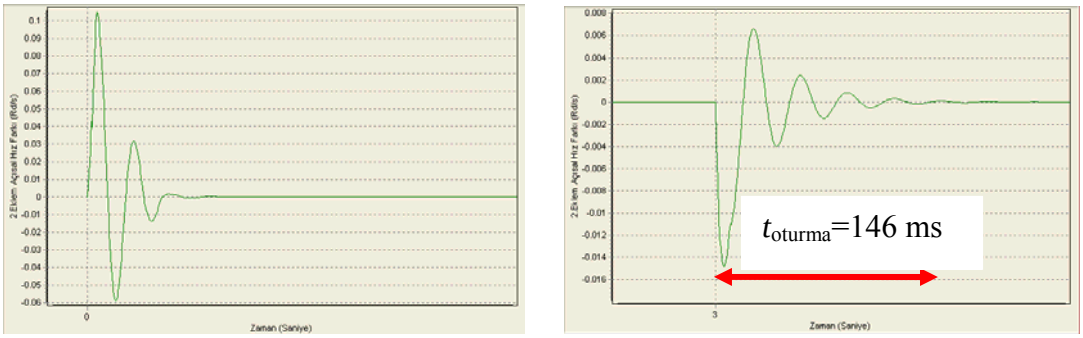

Şekil 4.82. Eklemlere uygulanan tork grafikleri (GPC SISO, Örnek 1, Durum 3)




Şekil 4.83. Eklemlerin açısal hız grafikleri (GPC SISO, Örnek 1, Durum 3)

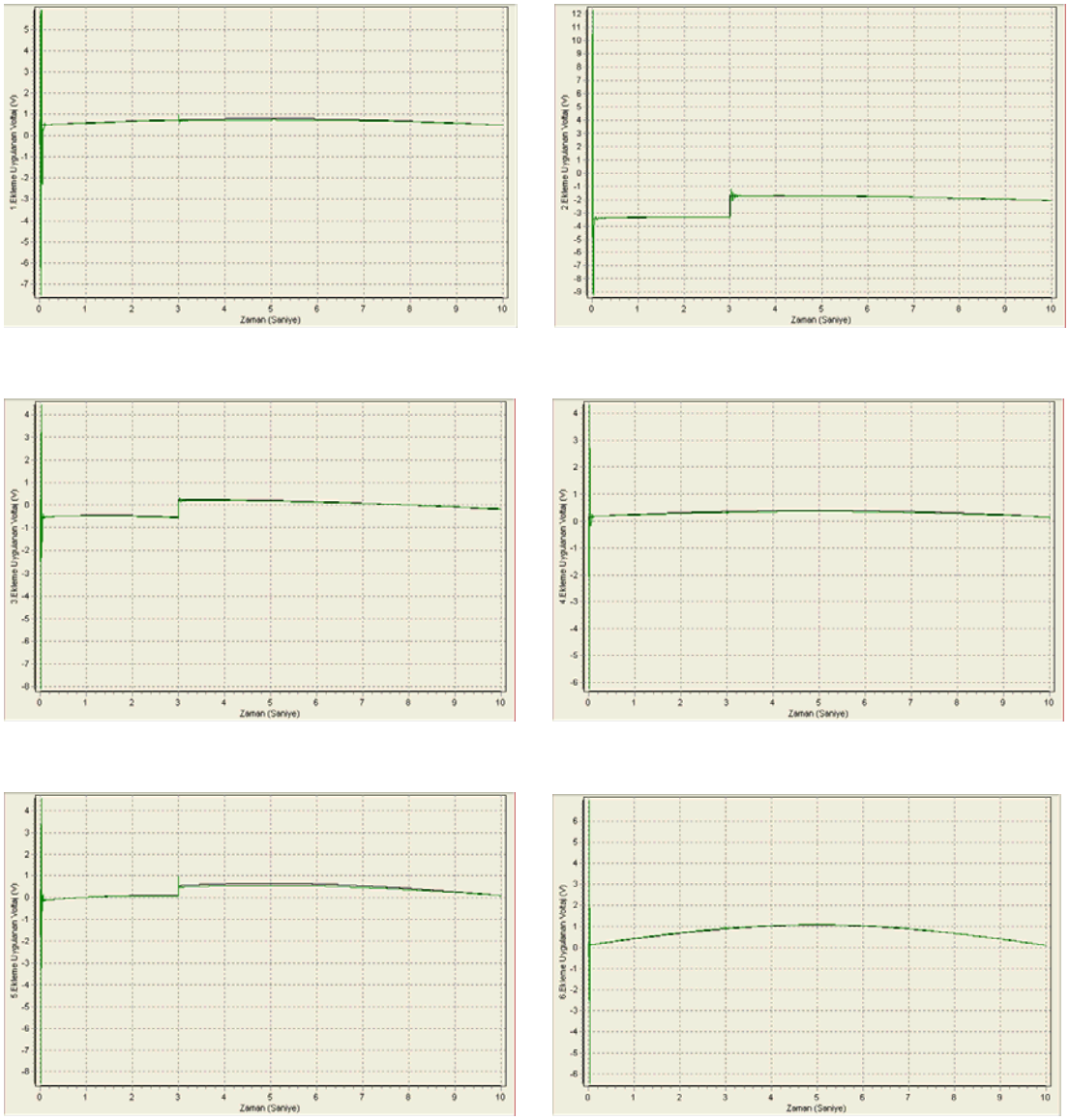



Şekil 4.84. 2. Ekleme ait açısal hız hatası grafiği (GPC SISO, Örnek 1, Durum 3)

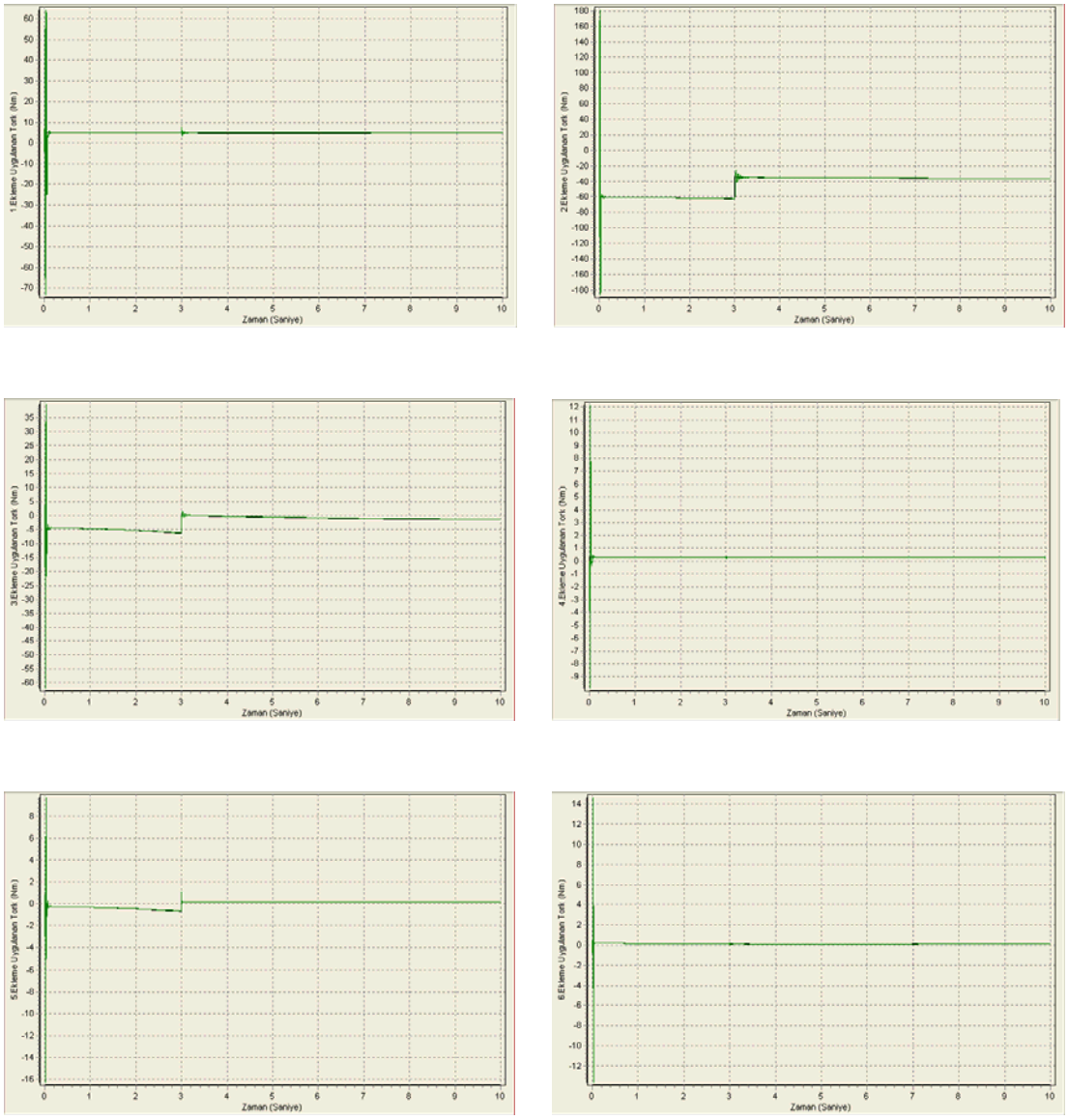



Şekil 4.85. 2. Ekleme ait açısal hız hatası ayrıntı grafiği (GPC SISO, Örnek 1, Durum 3)

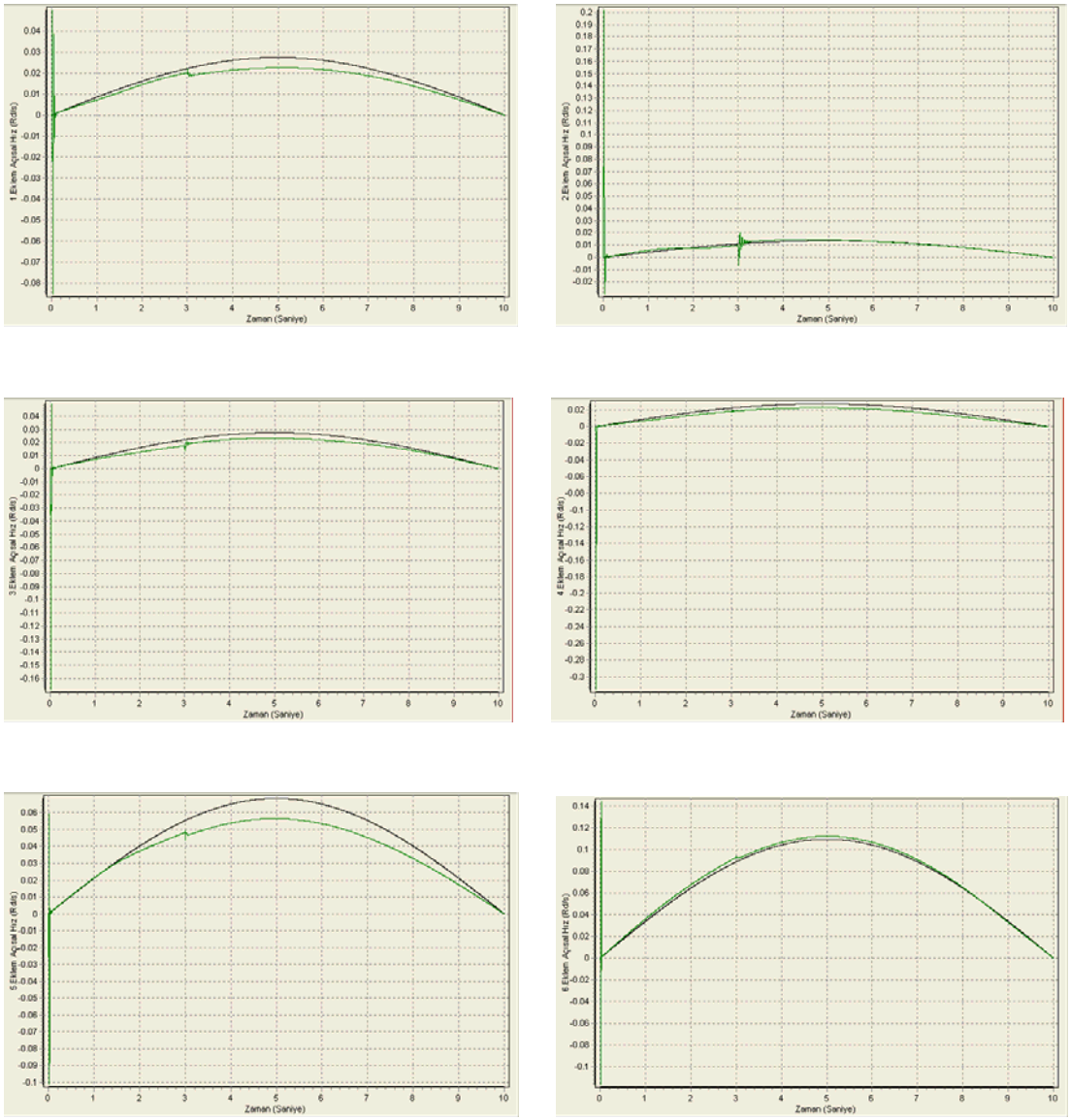
3. Durum ’da GPC SISO algoritmasına ait sonuçlar incelendiğinde yükün düştüğü 3.sn de tork ve gerilim eğrilerinde farklı genliklerde salınımların oluştuğu görülmektedir. Salınımların sonunda algoritma bu yeni duruma kendini adapte ederek kontrole devam etmektedir. Yük düşmesinden sonra eklemlere uygulanan tork ve gerilim eğrileri yeni değerlerine düşerek düzgün bir şeklinde hareketlerini tamamlamaktadırlar. Yük düşme anı dışında referans yörüngeler oldukça yakın takip edilmiştir. Açısal hız hatalarına bakıldığında yük düşme anı dışında hatalar hemen hemen sıfıra yakınsamıştır. Hareket başlangıcı ve yük düşme anında farklı genlikteki hız hataları salınımları hata grafiklerinde açıkça görülmektedir. GPC SISO algoritması ile yük düşme anında 0.006 ile -0.014 (Rd/sn) arası (mutlak fark 0.02 Rd/sn) genliğe sahip yaklaşık 8 adet salınım oluşmuştur. Sistem oturma süresi ise 146 ms olarak ölçülmüştür.



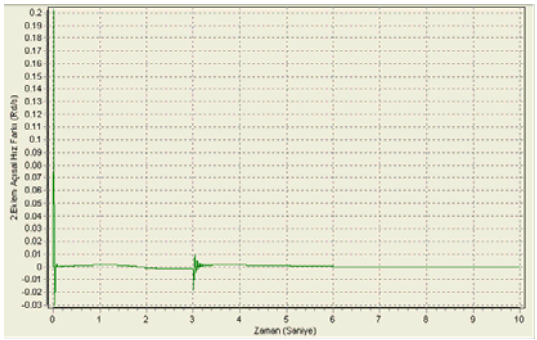

Şekil 4.86. Eklemlere uygulanan gerilim grafikleri (GPC MIMO, Örnek 1, Durum 3)



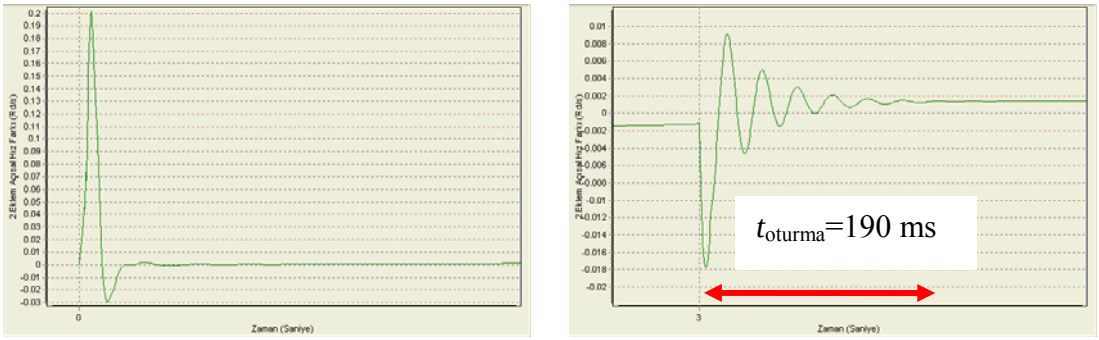

Şekil 4.87. Eklemlere uygulanan tork grafikleri (GPC MIMO, Örnek 1, Durum 3)




Şekil 4.88. Eklemlerin açısal hız grafikleri (GPC MIMO, Örnek 1, Durum 3)

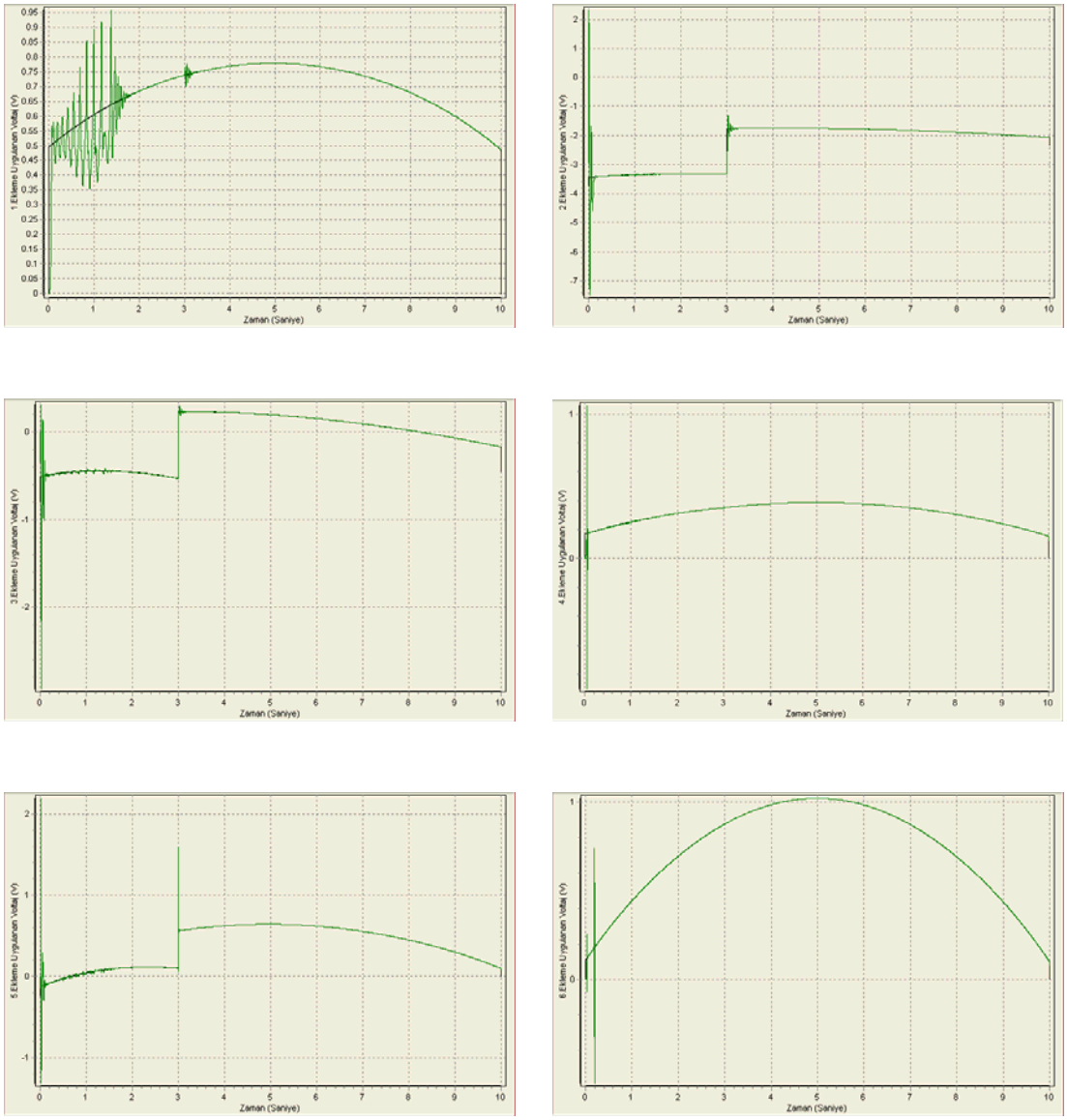



Şekil 4.89. 2. Ekleme ait açısal hız hatası grafiği (GPC MIMO, Örnek 1, Durum 3)

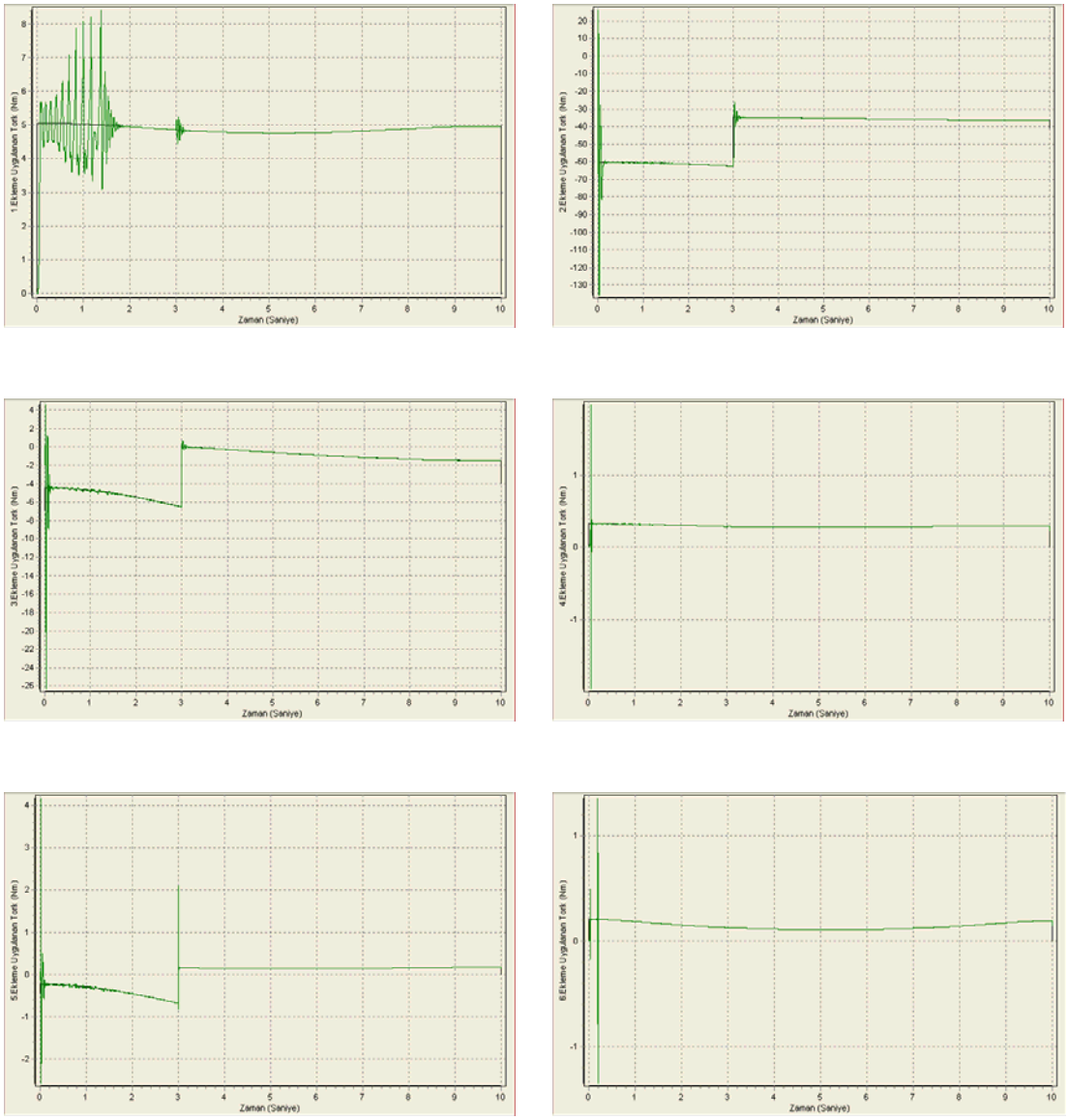



Şekil 4.90. 2. Ekleme ait açısal hız hatası ayrıntı grafiği (GPC MIMO, Örnek 1, Durum 3)

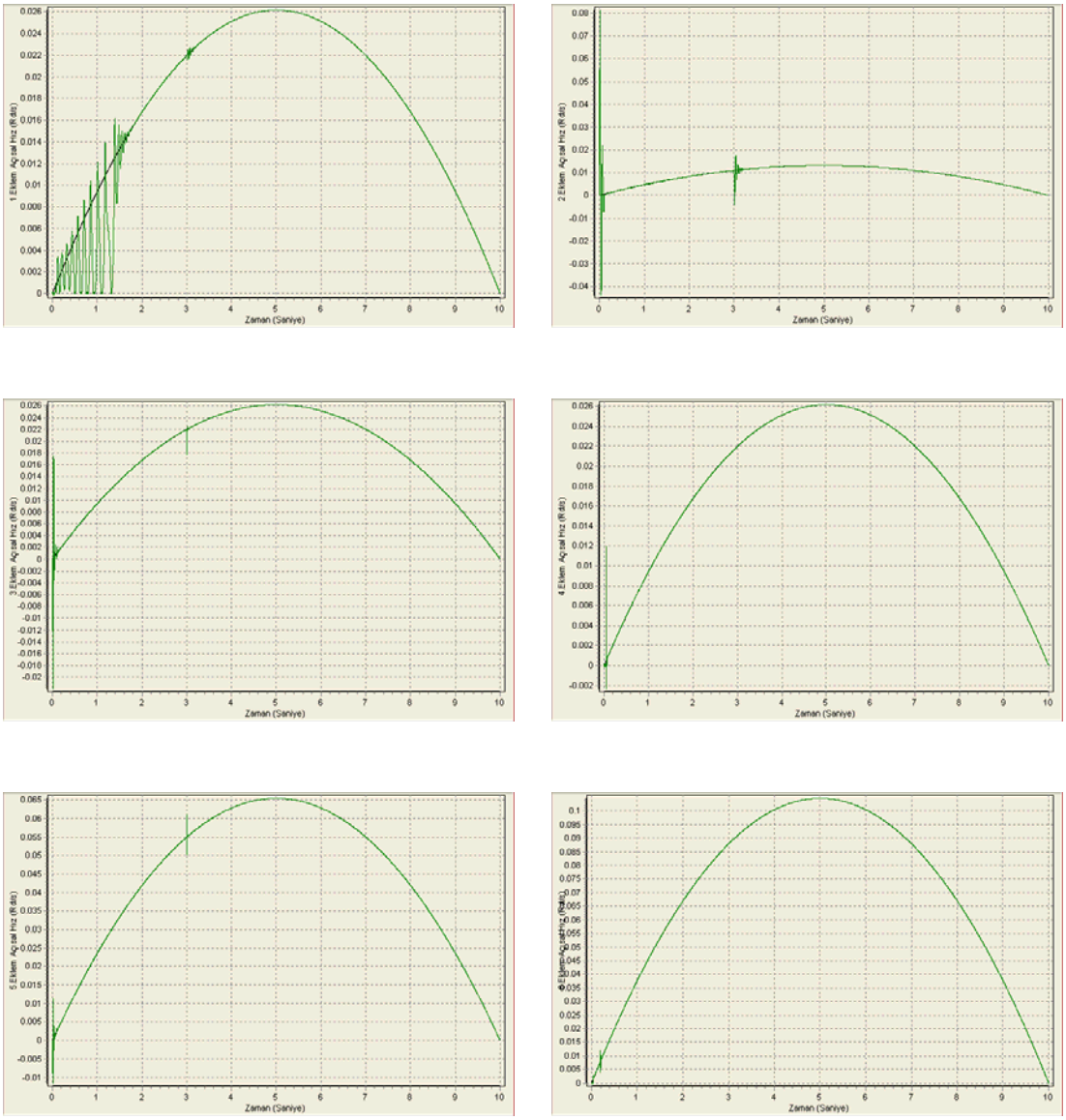
GPC MIMO ‘ya ait 3. Durum sonuçlarında tork ve gerilim eğrilerinde yükün düştüğü an dışında gayet düzgündür. Açısal hızlarda referans yörüngelerden sapmalar görülmektedir. 2. ekleme ait hız hataları incelendiğinde yük düşme anında 0.008 ile -0.018 (Rd/sn) arası (mutlak hata ~0.026) yaklaşık 10 adet salınım oluşmuştur. Oturma süresi 190 ms olarak ölçülmüştür.



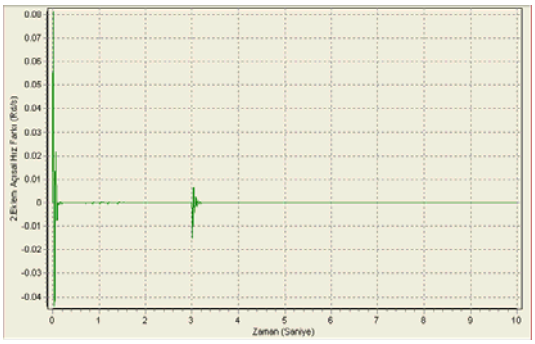

Şekil 4.91. Eklemlere uygulanan gerilim grafikleri (SGA-GPC SISO, Örnek 1, Durum 3)



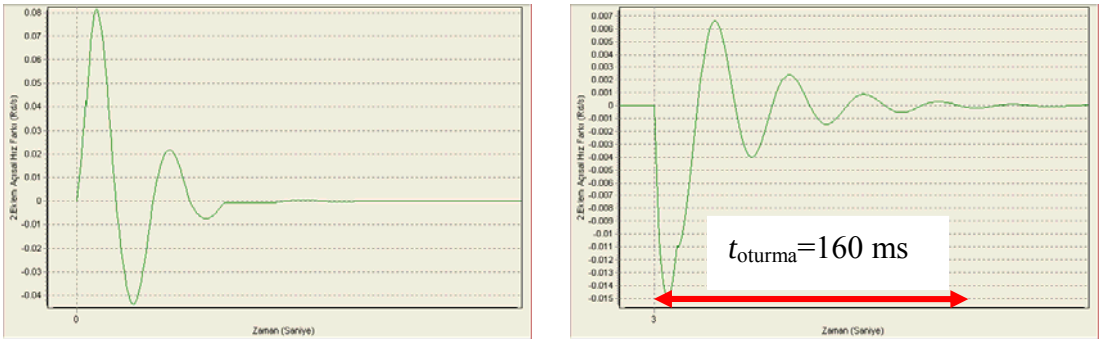

Şekil 4.92. Eklemlere uygulanan tork grafikleri (SGA-GPC SISO, Örnek 1, Durum 3)




Şekil 4.93. Eklemlerin açısal hız grafikleri (SGA-GPC SISO, Örnek 1, Durum 3)

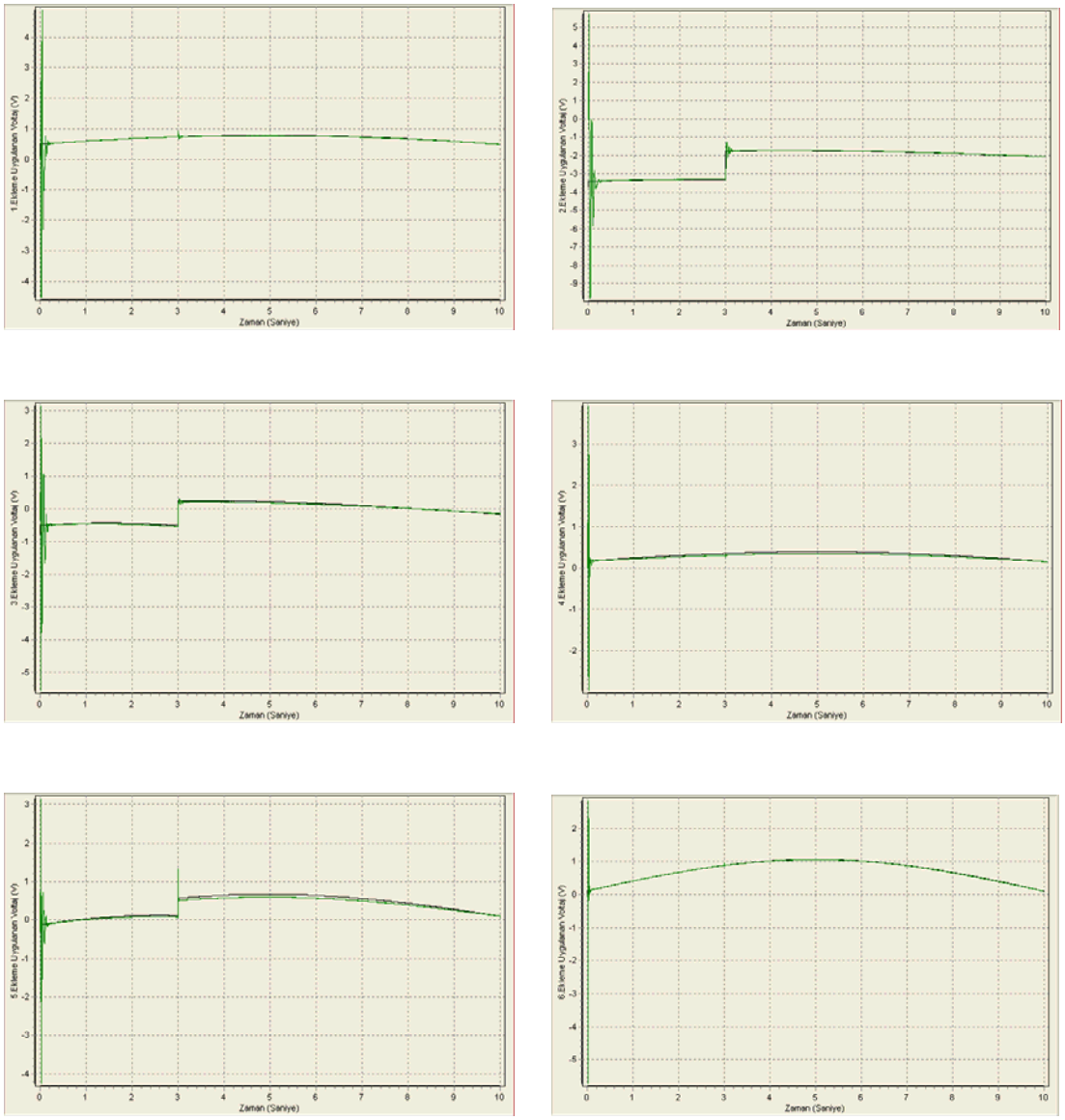



Şekil 4.94. 2. Ekleme ait açısal hız hatası grafiği (SGA-GPC SISO, Örnek 1, Durum 3)

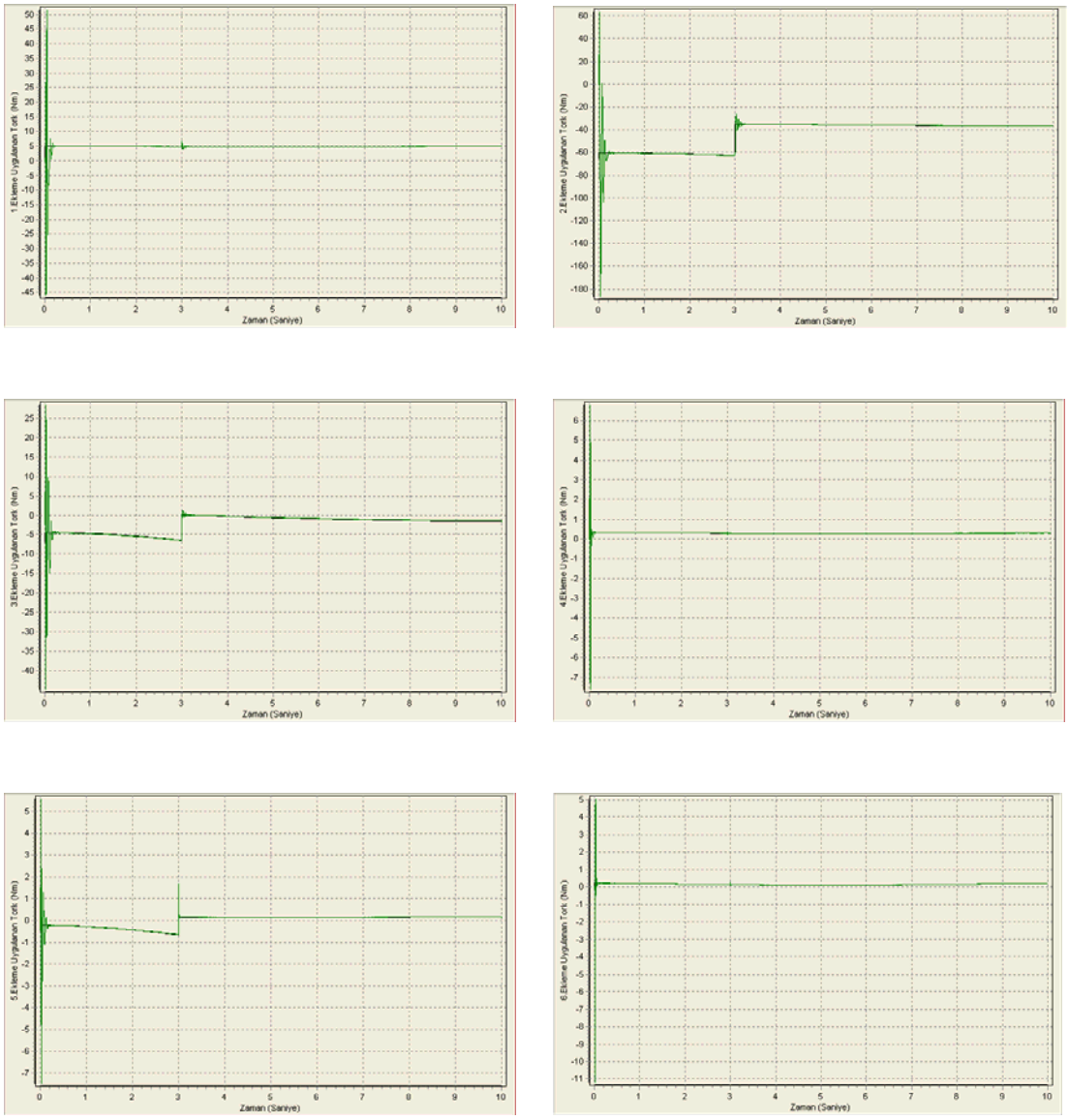



Şekil 4.95. 2. Ekleme ait açısal hız hatası ayrıntı grafiği (SGA-GPC SISO, Örnek 1, Durum 3)

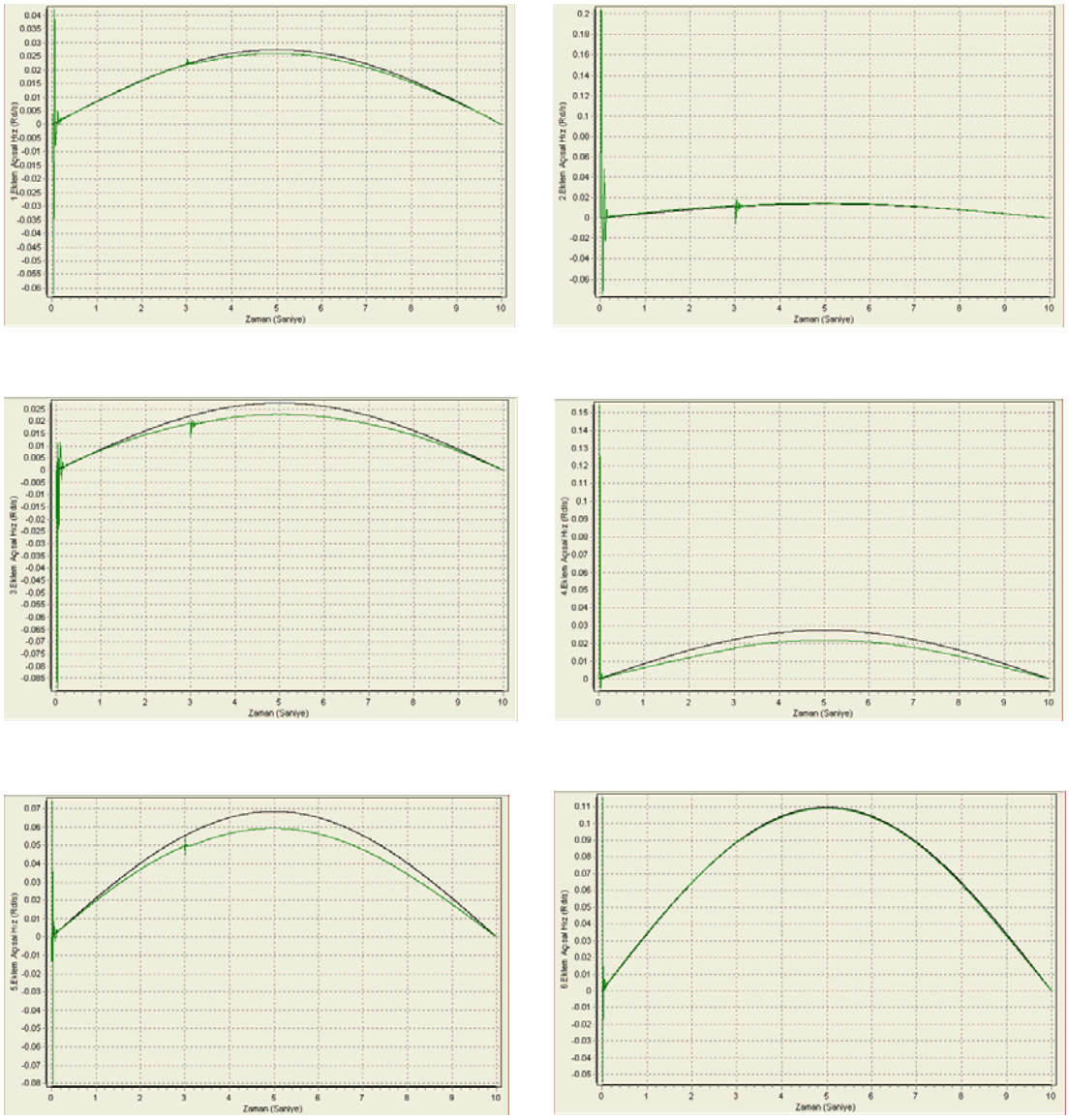
SGA-GPC SISO algoritmasına ait tork ve gerilim eğrilerinde yük düşme anı dışında hareket başlangıcında da ani değişimler görülmektedir. Bu değişimler hız eğrilerine ve hız hatalarına da yansımıştır. Yük düşme anına bakıldığında ise 0.006 ile -0.015(Rd/sn) arası (mutlak hata 0.021) yaklaşık 7 adet salınım oluşmuştur. Oturma süresi 160 ms ölçülmüştür.



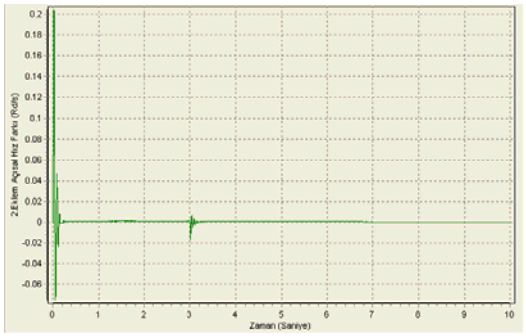

Şekil 4.96. Eklemlere uygulanan gerilim grafikleri (SGA-GPC MIMO, Örnek 1, Durum 3)



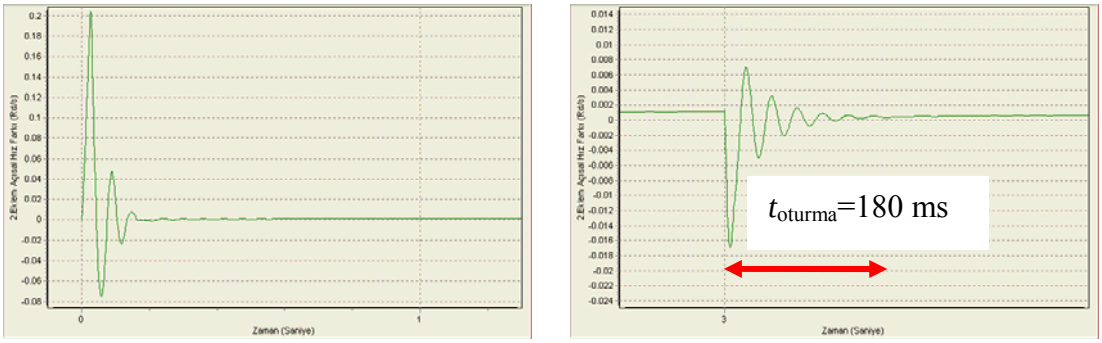

Şekil 4.97. Eklemlere uygulanan tork grafikleri (SGA-GPC MIMO, Örnek 1, Durum 3)




Şekil 4.98. Eklemlerin açısal hız grafikleri (SGA-GPC MIMO, Örnek 1, Durum 3)

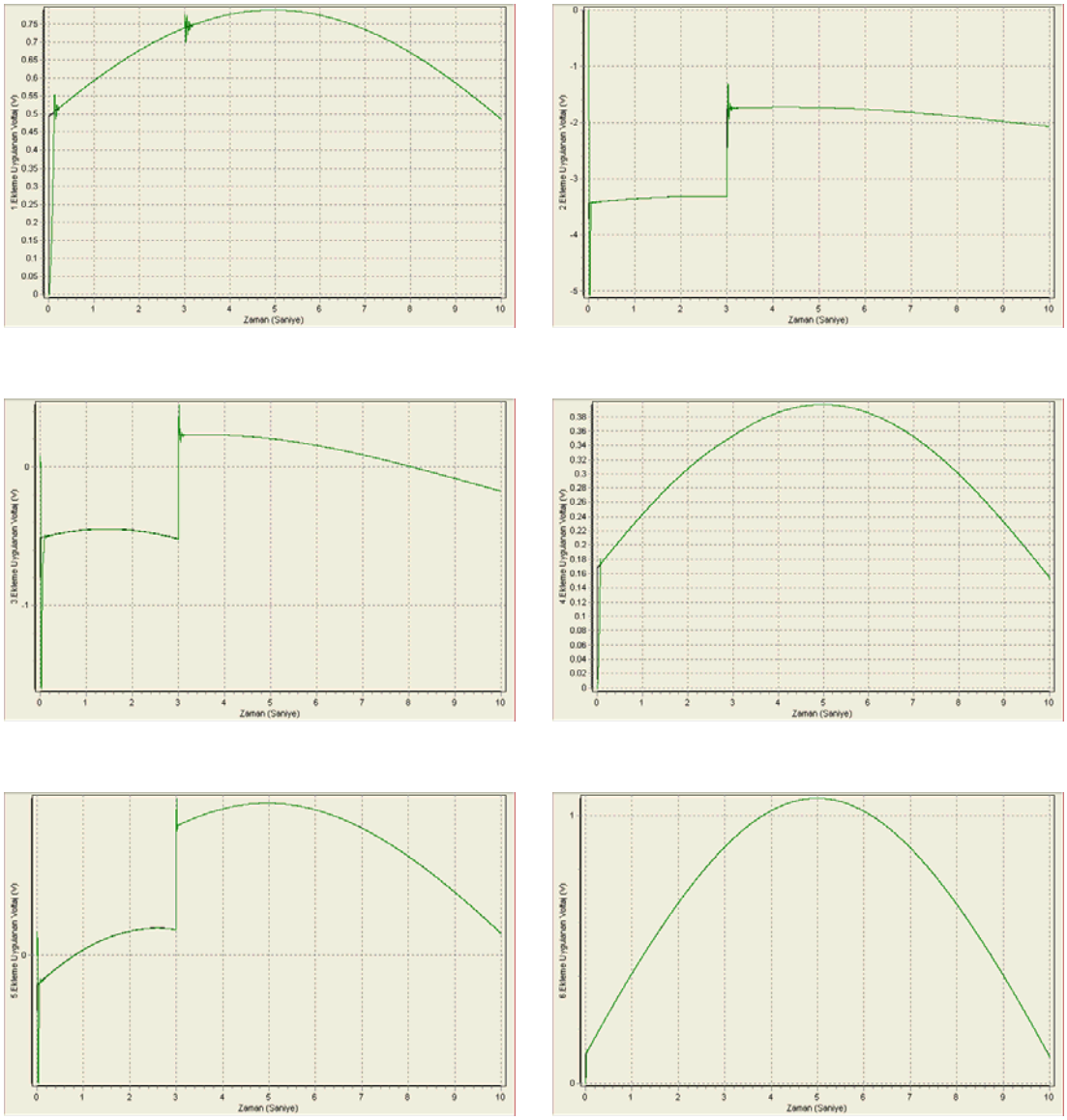



Şekil 4.99. 2. Ekleme ait açısal hız hatası grafiği (SGA-GPC MIMO, Örnek 1, Durum 3)

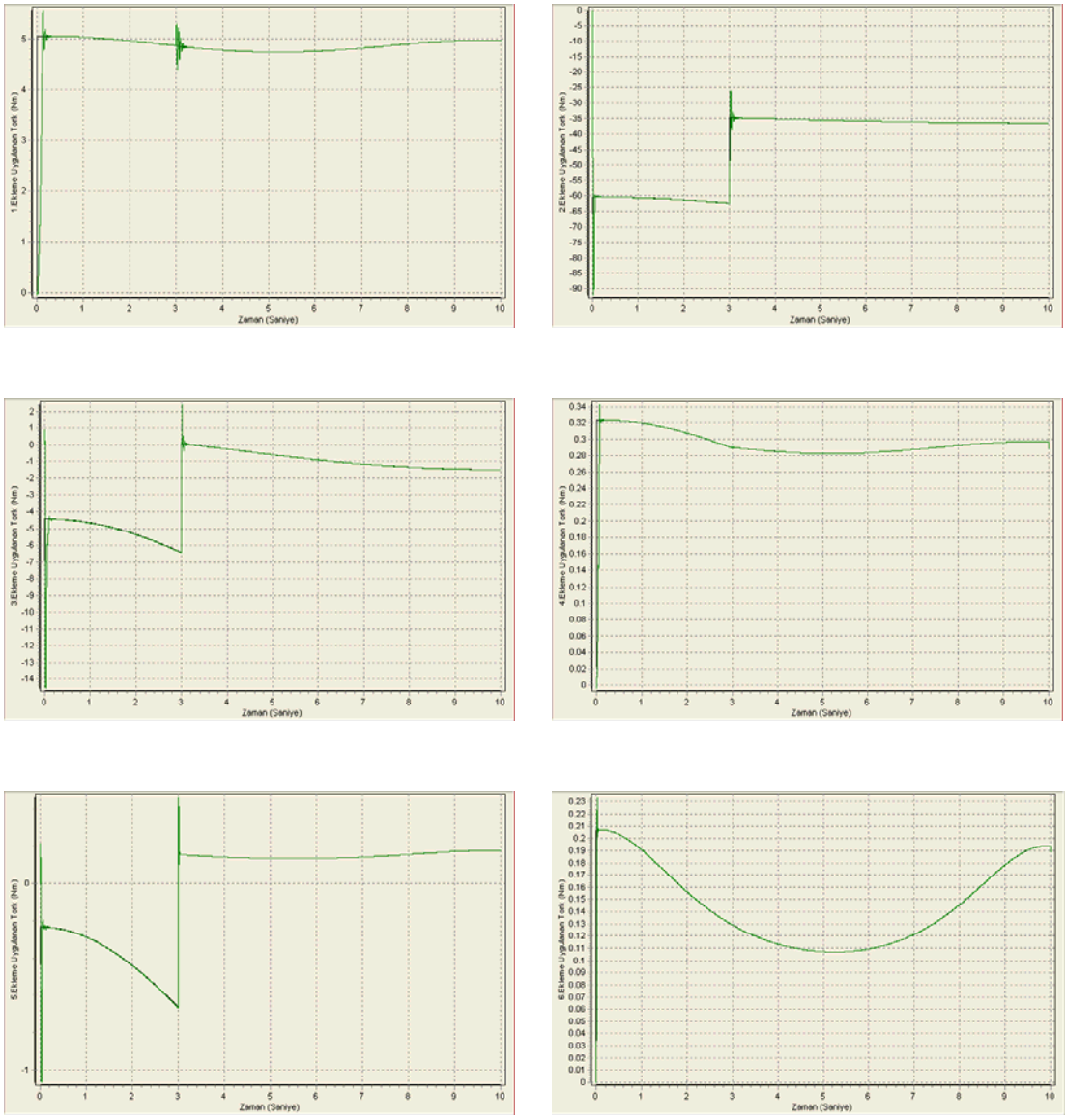



Şekil 4.100. 2. Ekleme ait açısal hız hatası ayrıntı grafiği (SGA-GPC MIMO, Örnek 1, Durum 3)

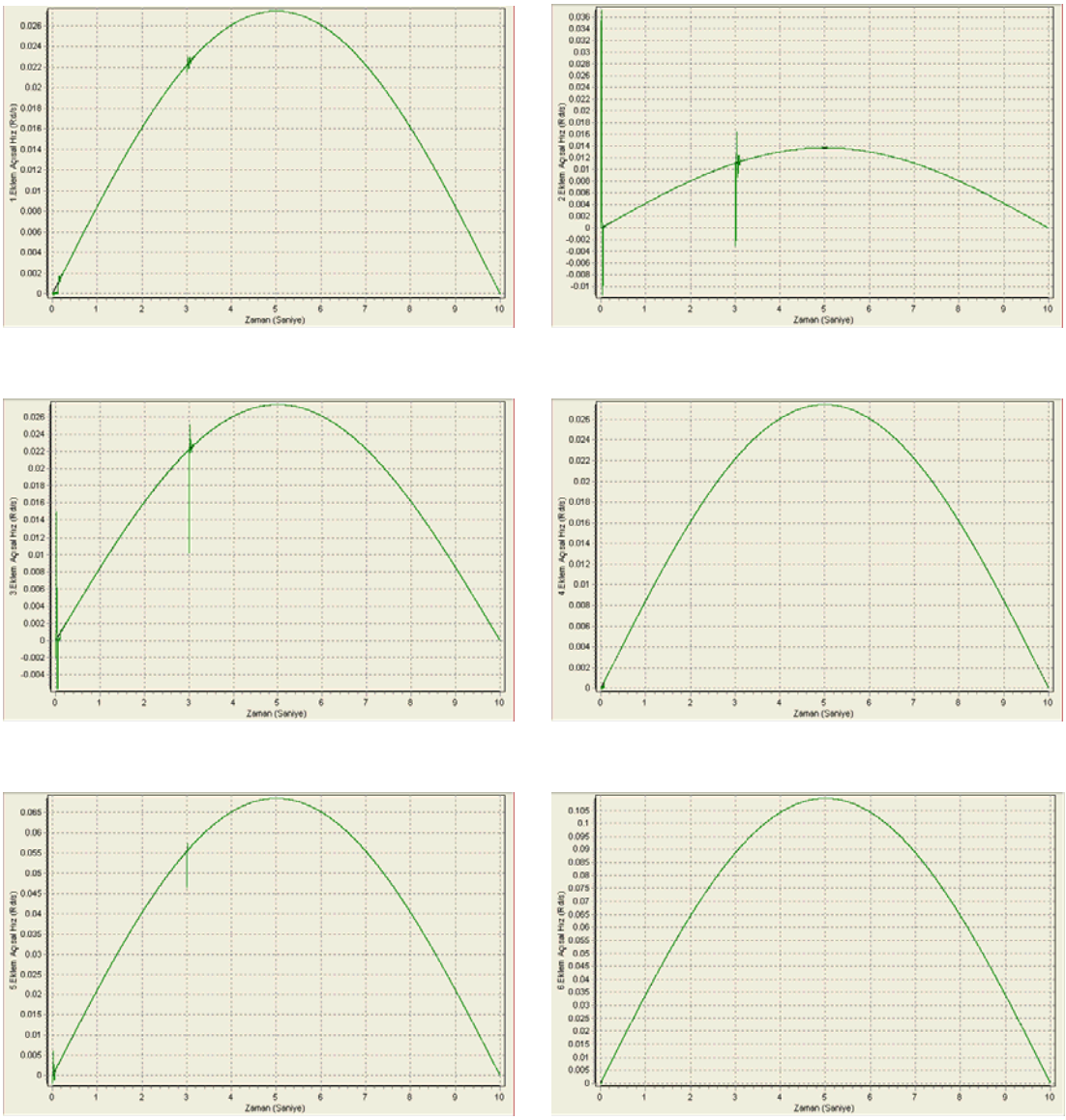
3. Durum ‘da SGA-GPC MIMO algoritması genel olarak başarılı olmuştur. Yükün düştüğü an dışında tork ve gerilim eğrileri gayet düzgündür. Referans yörüngelerde ise bir miktar sapmalar görülmüştür. Yük düşmesi anında 0.006 ile -0.016 (Rd/sn) arası (mutlak hata ~0.022 Rd/sn) yaklaşık 10 adet salınım oluşmuştur. Oturma süresi 180 ms ölçülmüştür.



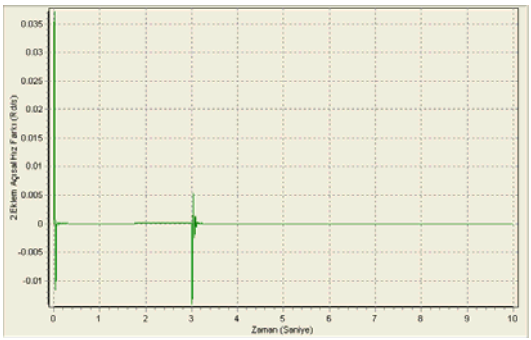

Şekil 4.101. Eklemlere uygulanan gerilim grafikleri (NGPC SISO, Örnek 1, Durum 3)



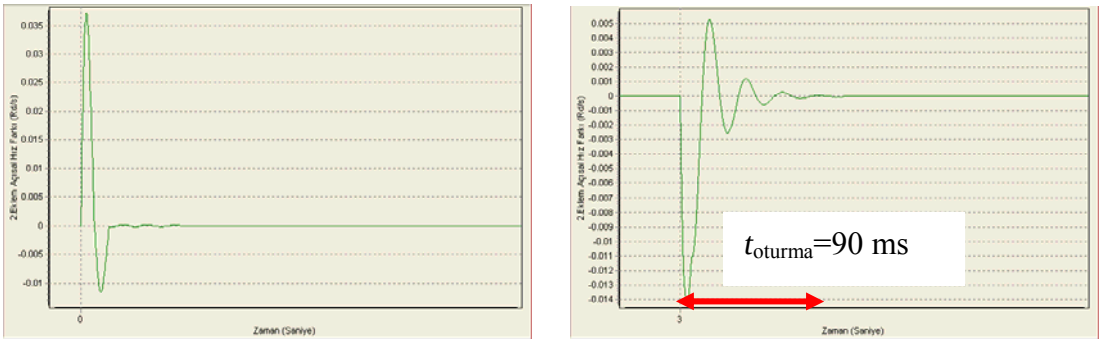

Şekil 4.102. Eklemlere uygulanan tork grafikleri (NGPC SISO, Örnek 1, Durum 3)




Şekil 4.103. Eklemlerin açısal hız grafikleri (NGPC SISO, Örnek 1, Durum 3)

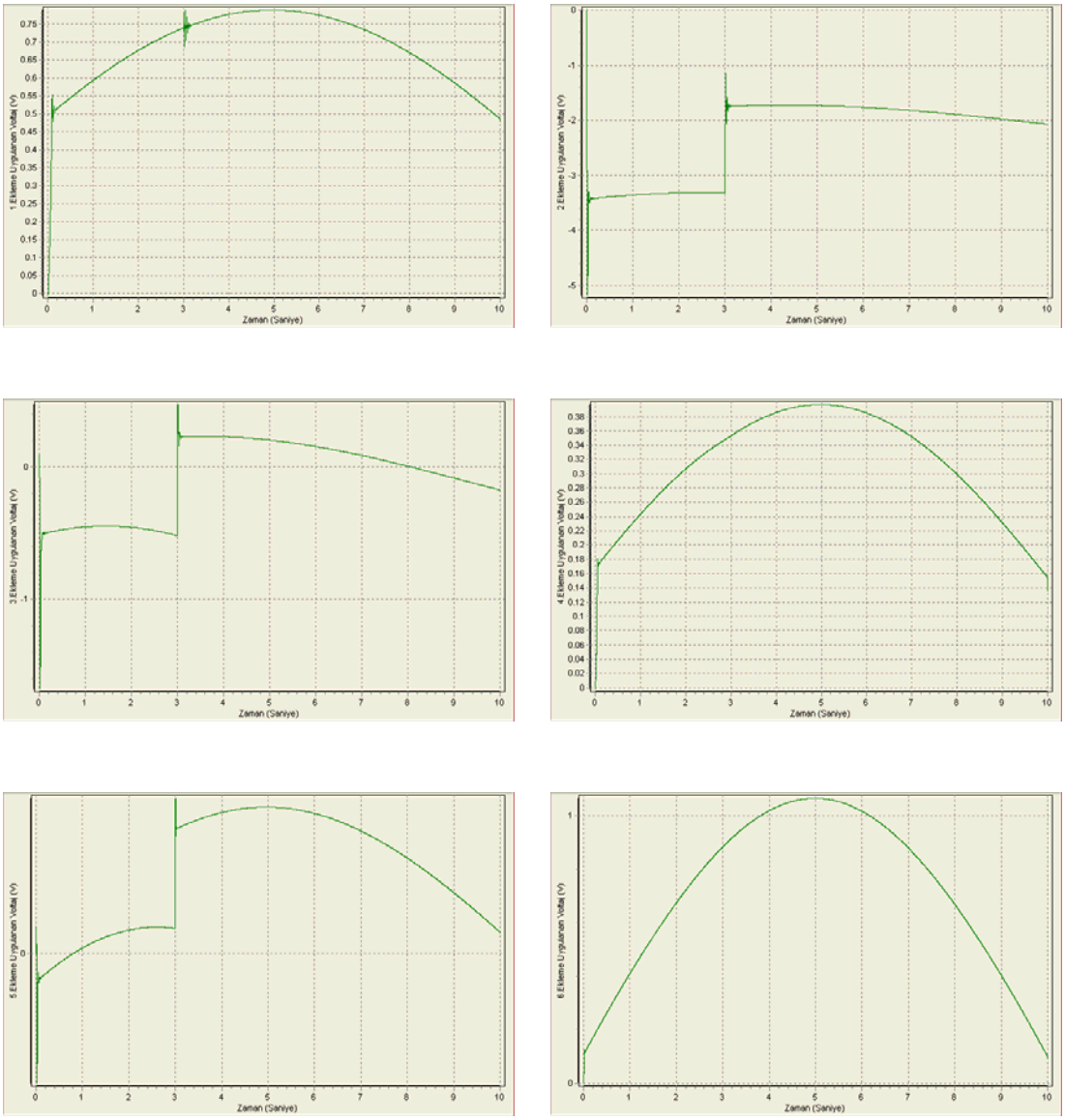



Şekil 4.104. 2. Ekleme ait açısal hız hatası grafiği (NGPC SISO, Örnek 1, Durum 3)

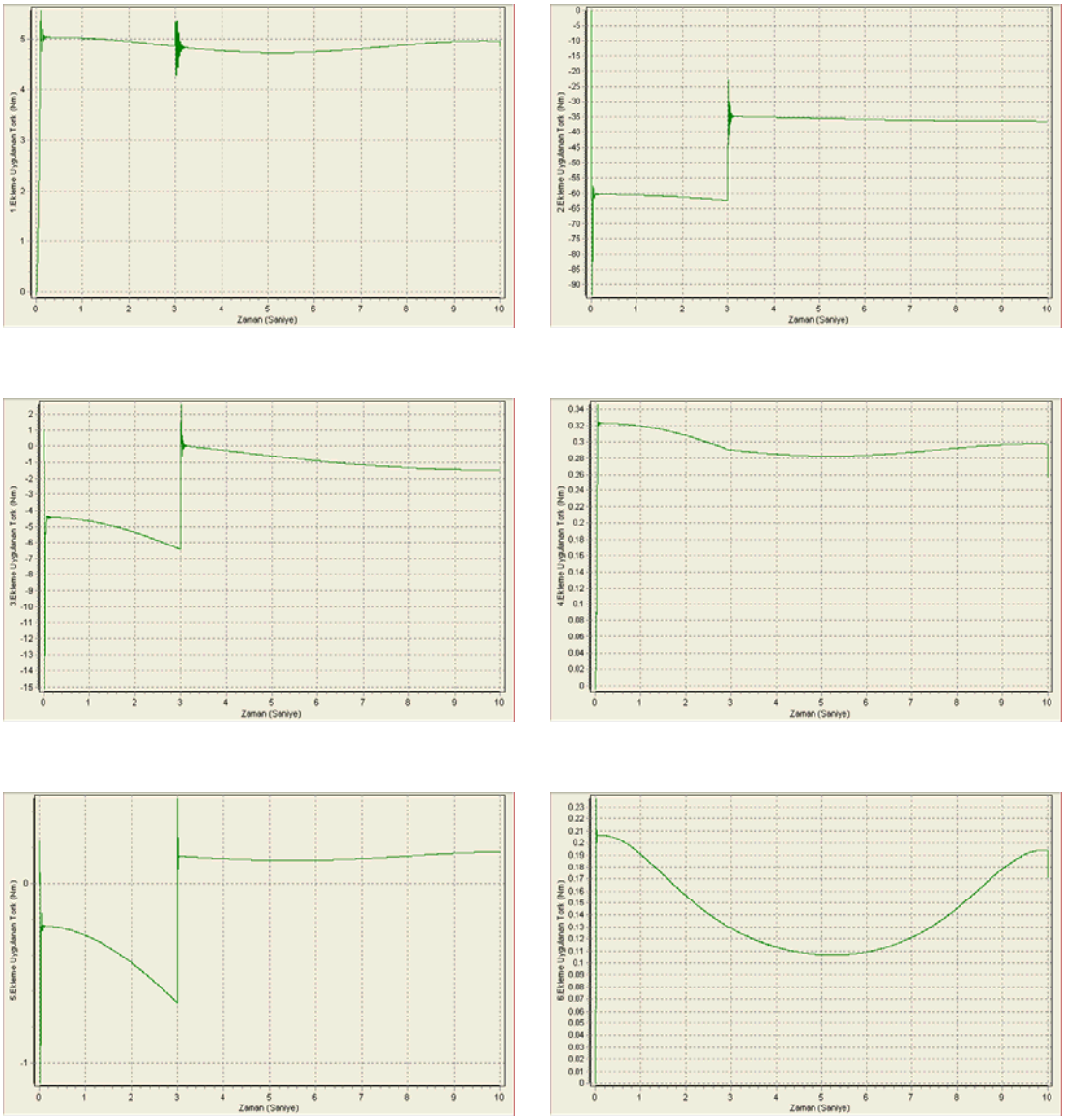



Şekil 4.105. 2. Ekleme ait açısal hız hatası ayrıntı grafiği (NGPC SISO, Örnek 1, Durum 3)

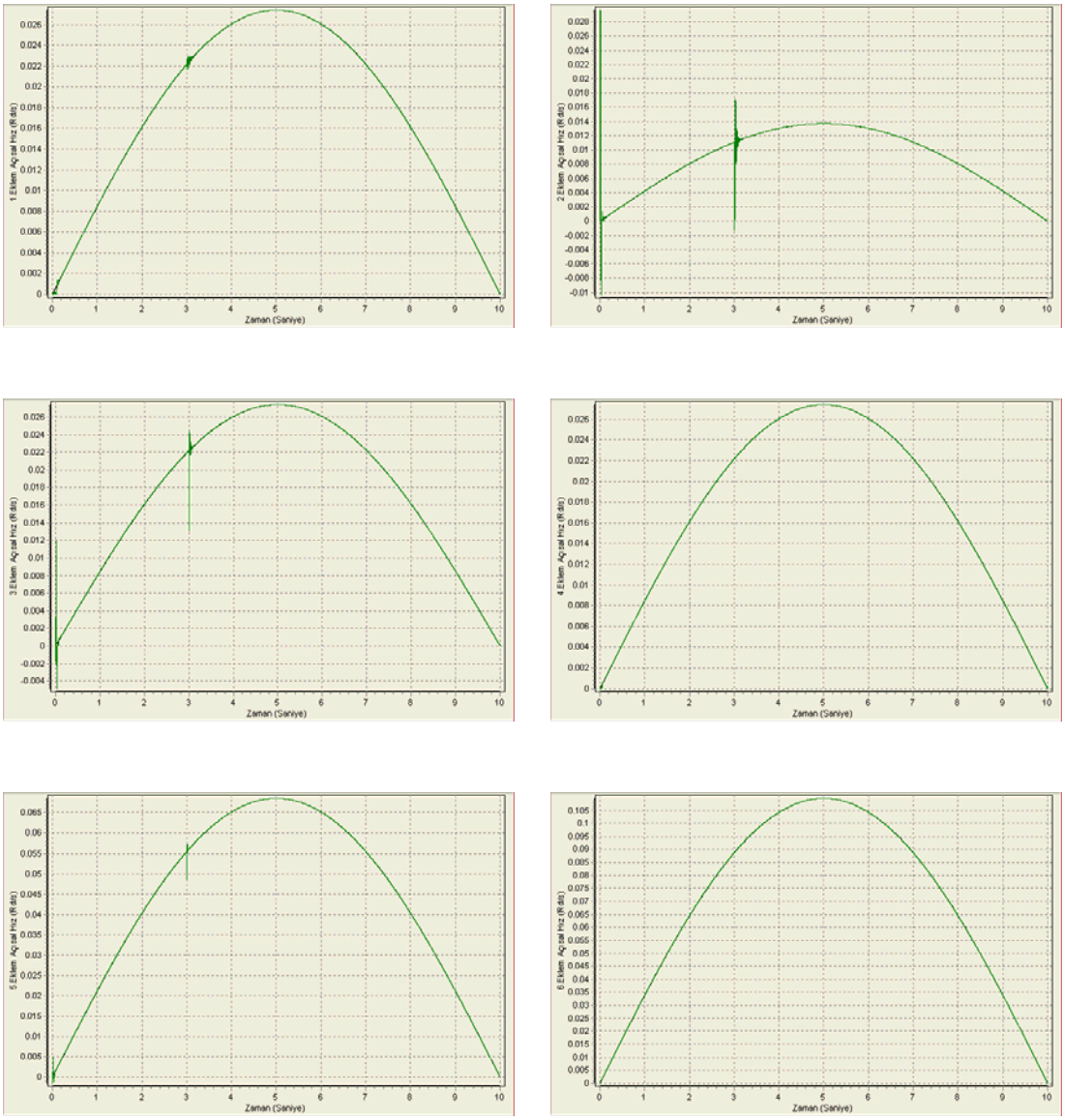
NGPC SISO algoritması ile 3. Durum ‘da yük düşme anı dışında tork ve gerilim eğrileri oldukça düzgündür. Referans yörüngeler oldukça yakın takip edilmiştir. Yük düşme anında 0.005 ile -0.014 (Rd/sn) arası (mutlak hata ~019 Rd/sn) yaklaşık 6 adet salınım meydana gelmiştir. Oturma süresi 90 ms ölçülmüştür.



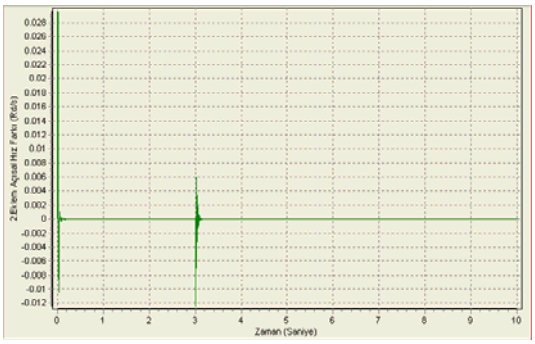

Şekil 4.106. Eklemlere uygulanan gerilim grafikleri (ENGPC SISO, Örnek 1, Durum 3)



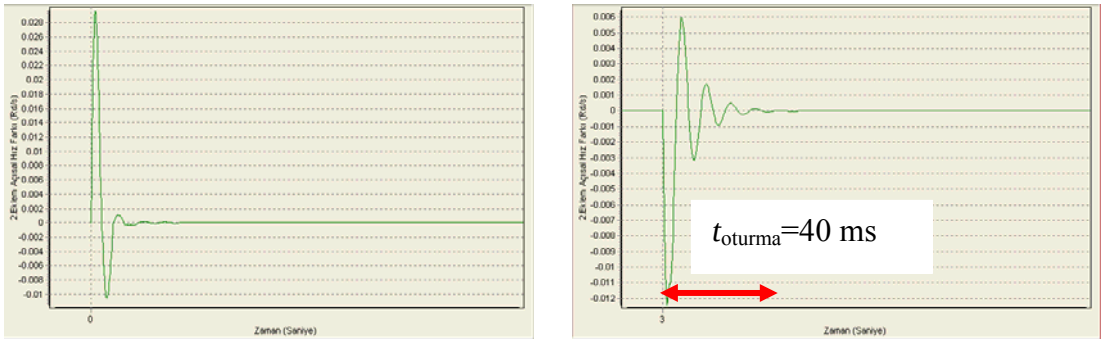

Şekil 4.107. Eklemlere uygulanan tork grafikleri (ENGPC SISO, Örnek 1, Durum 3)




Şekil 4.108. Eklemlerin açısal hız grafikleri (ENGPC SISO, Örnek 1, Durum 3)




Şekil 4.109. 2. Ekleme ait açısal hız hatası grafiği (ENGPC SISO, Örnek 1, Durum 3)




Şekil 4.110. 2. Ekleme ait açısal hız hatası ayrıntı grafiği (ENGPC SISO, Örnek 1, Durum 3)

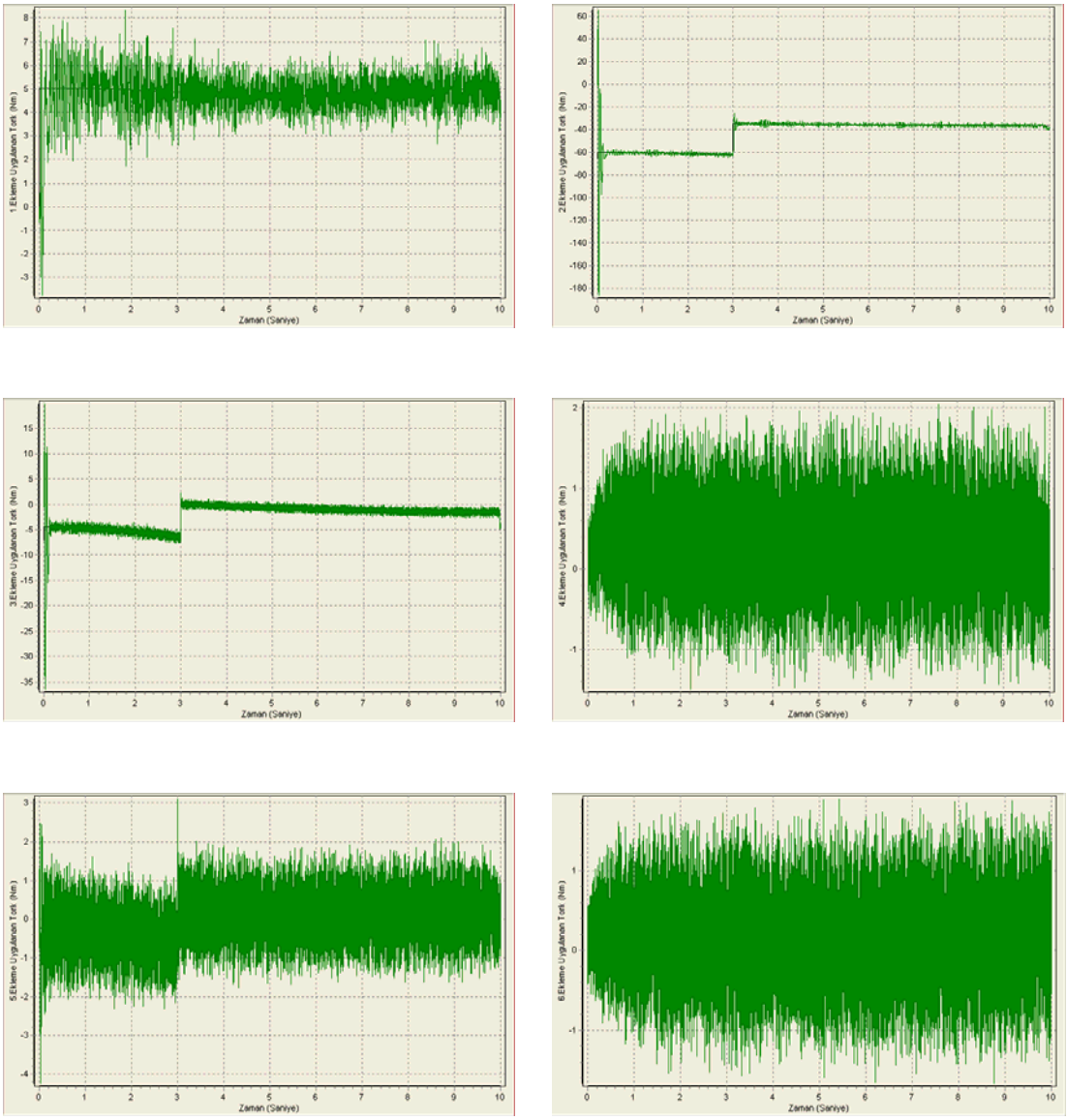
ENGPC SISO algoritması ile 3. Durum ‘da yük düşme anı dışında tork ve gerilim eğrileri oldukça düzgündür. Referans yörüngeler oldukça yakın takip edilmiştir. Yük düşme anında 0.006 ile -0.012 (Rd/sn) arası (mutlak hata ~018 Rd/sn) yaklaşık 6 adet salınım meydana gelmiştir. Oturma süresi 40 ms ölçülmüştür.

3. Durum ‘da genel olarak bir değerlendirme yapıldığında; hareket başlangıcında ve yük düşme anında farklı genlikte ve sayılarda salınımlar oluşmuştur. Yükün düşmesi ile robot kolunun dinamik modelinde oluşan değişimle meydana gelen bu salınımlar kısa sürede sönümlendikten sonra kontrol algoritmaları bu yeni duruma kendilerini adapte ederek kolun kontrolüne devam ettirmektedirler.

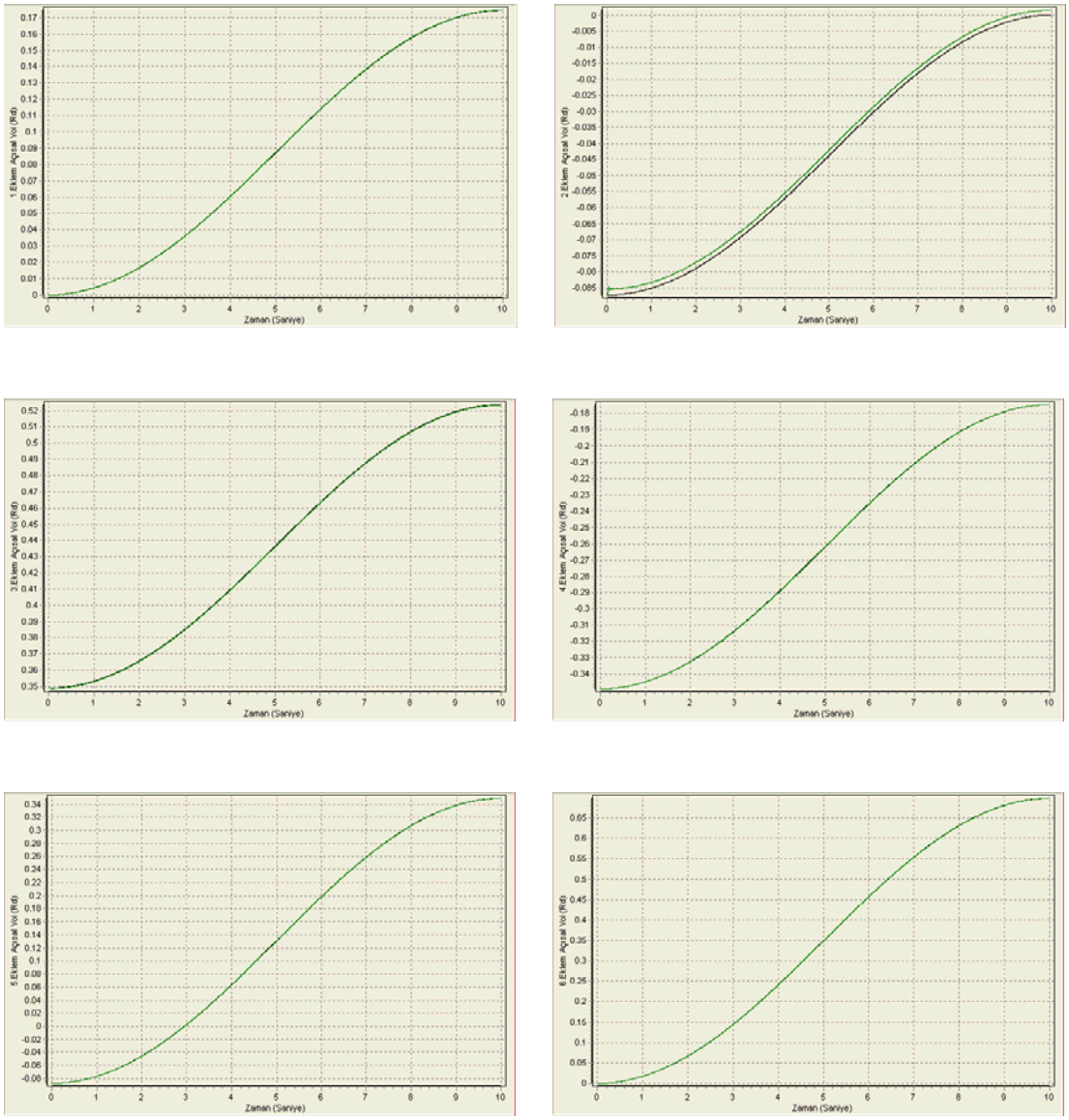
Yük düşme anındaki salınımların ayrıntılı bir şekilde incelenmesi için kontrolü en zor eklem olan 2. ekleme ait hız hataları grafikleri üzerinden değerlendirmeler yapılmıştır. Bu değerlendirmelere göre ENGPC SISO 0.018 Rd/sn lik mutlak hata ile en az genliğe sahip algoritmadır. Dolayısıyla değişken yük karşısında en dirençli algoritmadır. Bunu sırası ile NGPC SISO 0.019 Rd/sn, GPC SISO 0.02 Rd/sn, SGA-GPC SISO 0.021 Rd/sn, SGA-GPC MIMO 0.022 Rd/sn ve GPC MIMO 0.026Rd/sn ile izlemektedir. Öte yandan yük düşme anındaki sistem oturma süreleri incelendiğinde ENGPC SISO algoritmasında 40 ms ‘lik sürede istenilen değere ulaşılmaktadır. Buda yeni durum karşısında en hızlı adapte olan algoritma anlamına gelir ve sonuç olarak ENGPC SISO algoritması diğer algoritmalara kıyasla daha adaptif bir davranış sergilemiştir. Bunu NGPC SISO 90 ms, GPC SISO 146 ms, GPC SGA-GPC SISO 160 ms, SGA-GPC MIMO 180 ms ve GPC MIMO 190 ms ile

izlemektedir. Genetik algoritmanın adaptif özelliği zayıf olmasına rağmen MIMO yapıda iyileşme sağlamıştır.

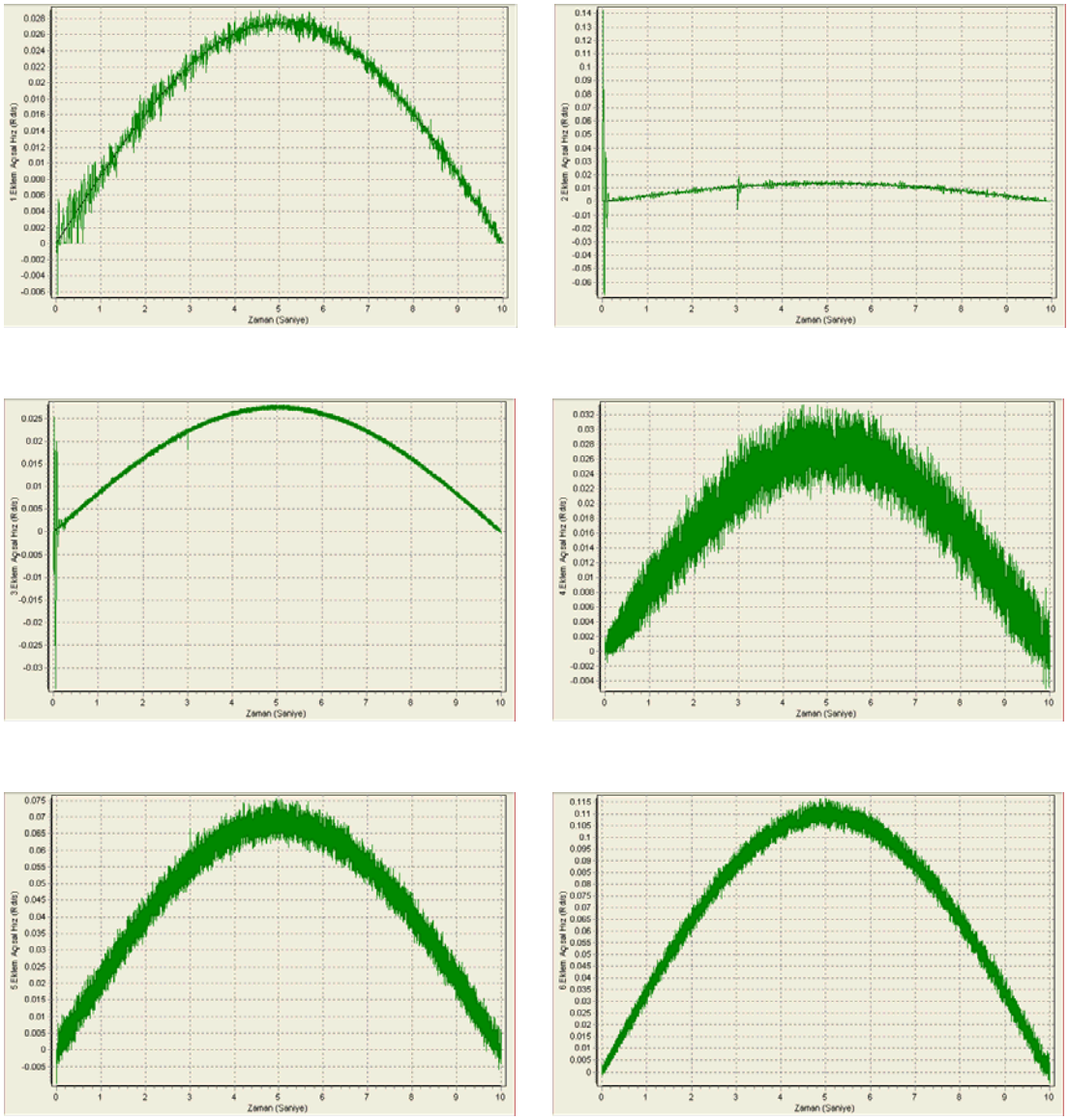
4. Durum ‘da her kontrol adımında eklemlere uygulanan tork değerine $-0.5\ Nm$ ile $+0.5\ Nm$ arasında rasgele değerde ilave bozucular uygulanarak kontrol güçleştirilmektedir. Kontrol şartları zorlaştırılan bu durumda kontrol algoritmalarının başarımının test edilmesi amaçlanmıştır. Algoritmalara ait grafikler Şekil 4.111 ‘den 4.134 ‘e kadar aşağıda sunulmuştur.



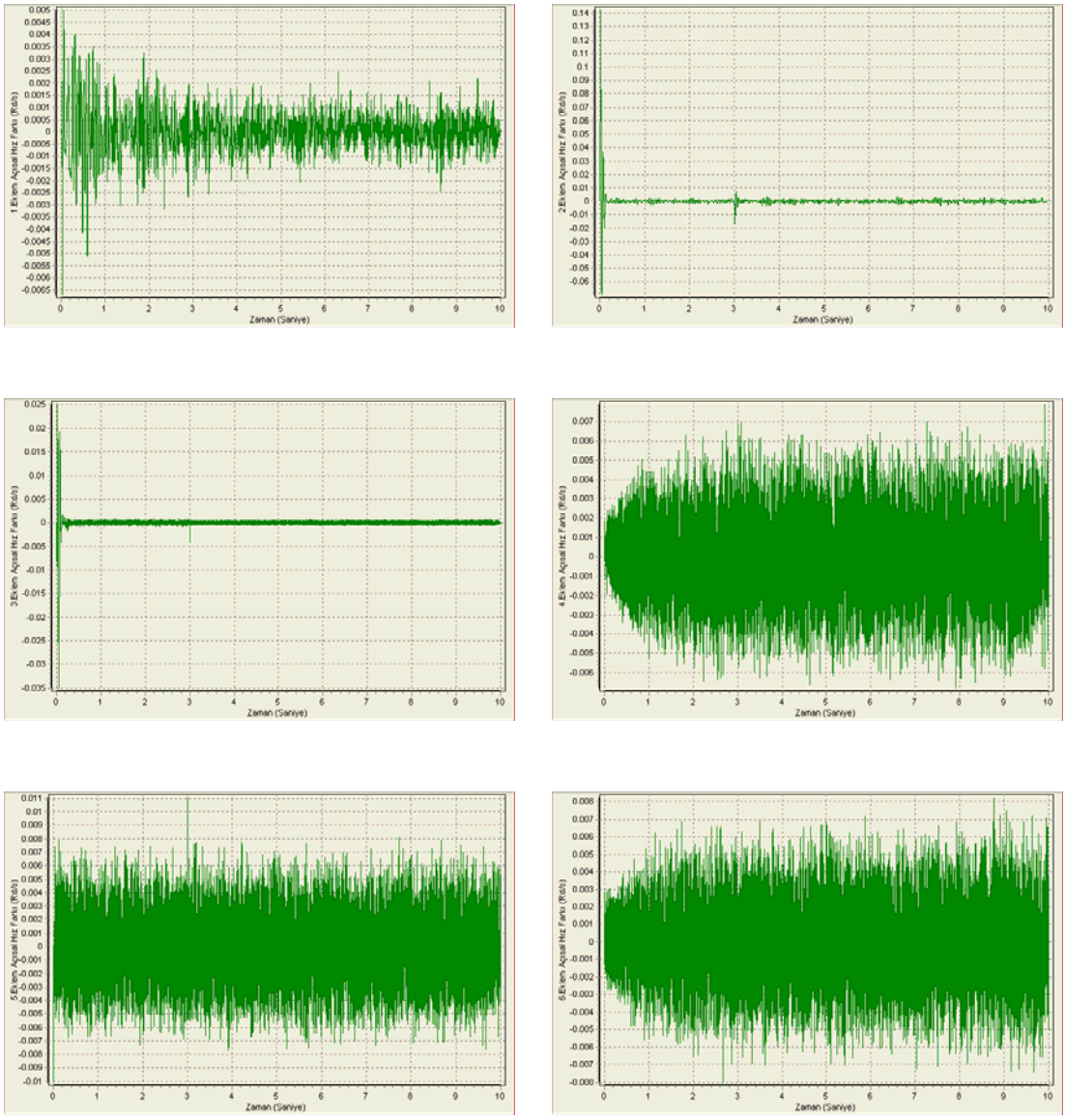

Şekil 4.111. Eklemlere uygulanan tork grafikleri (GPC SISO, Örnek 1, Durum 4)




Şekil 4.112. Eklemlerin takip ettiği açısal yol grafikleri (GPC SISO, Örnek 1, Durum 4)

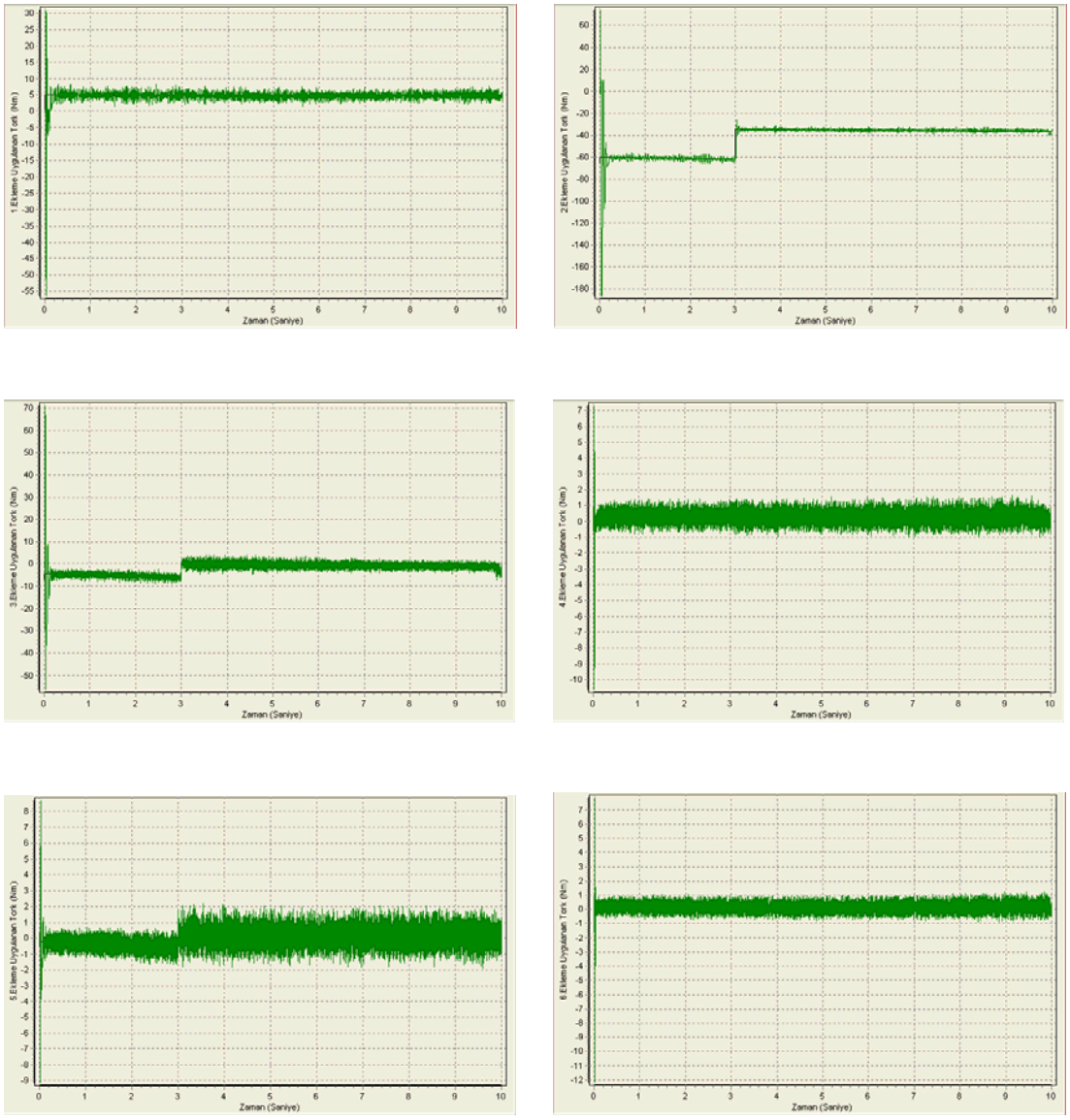



Şekil 4.113. Eklemlerin açısal hız grafikleri (GPC SISO, Örnek 1, Durum 4)

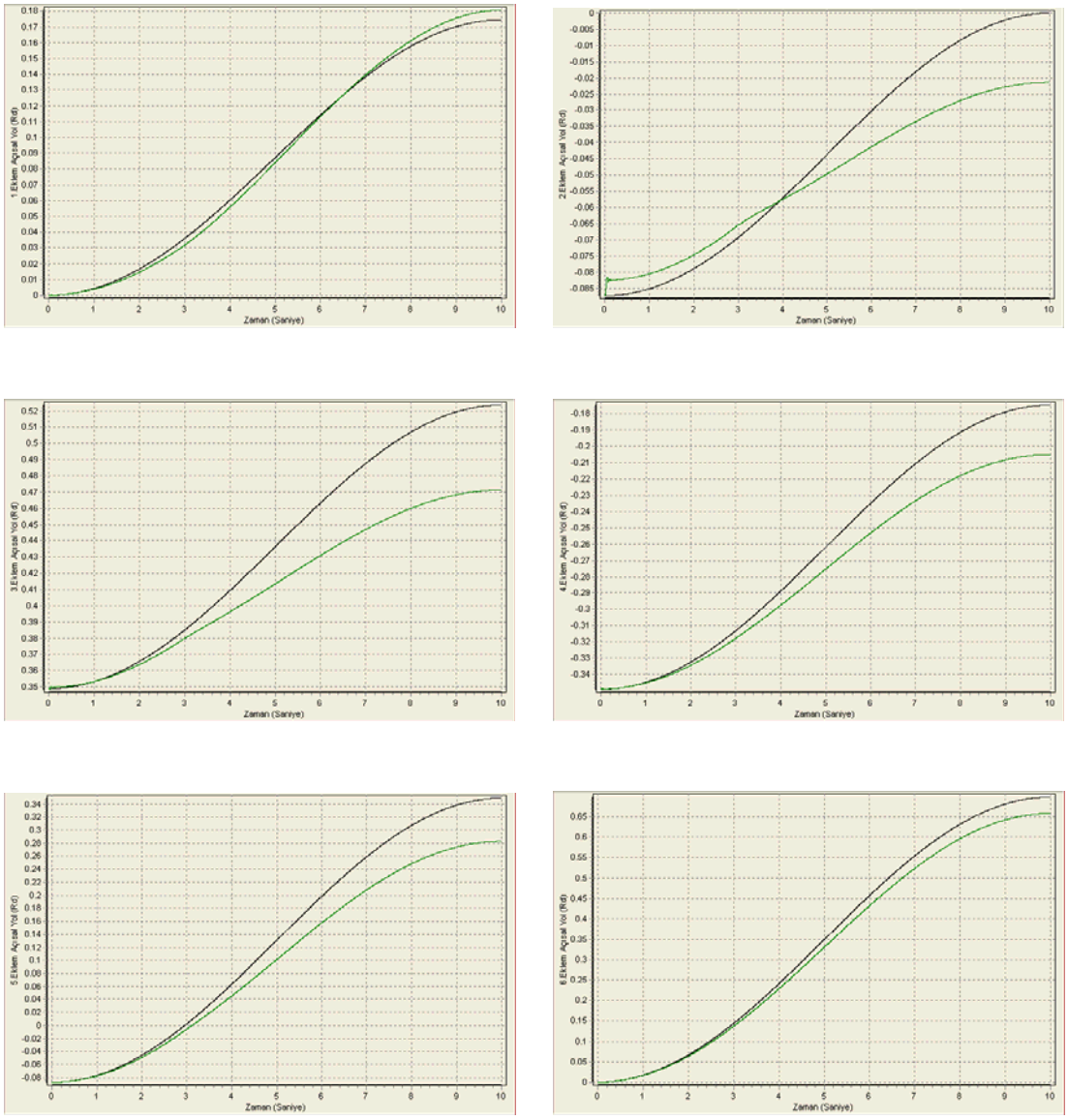



Şekil 4.114. Eklemlerin açısal hız hataları (GPC SISO, Örnek 1, Durum 4)

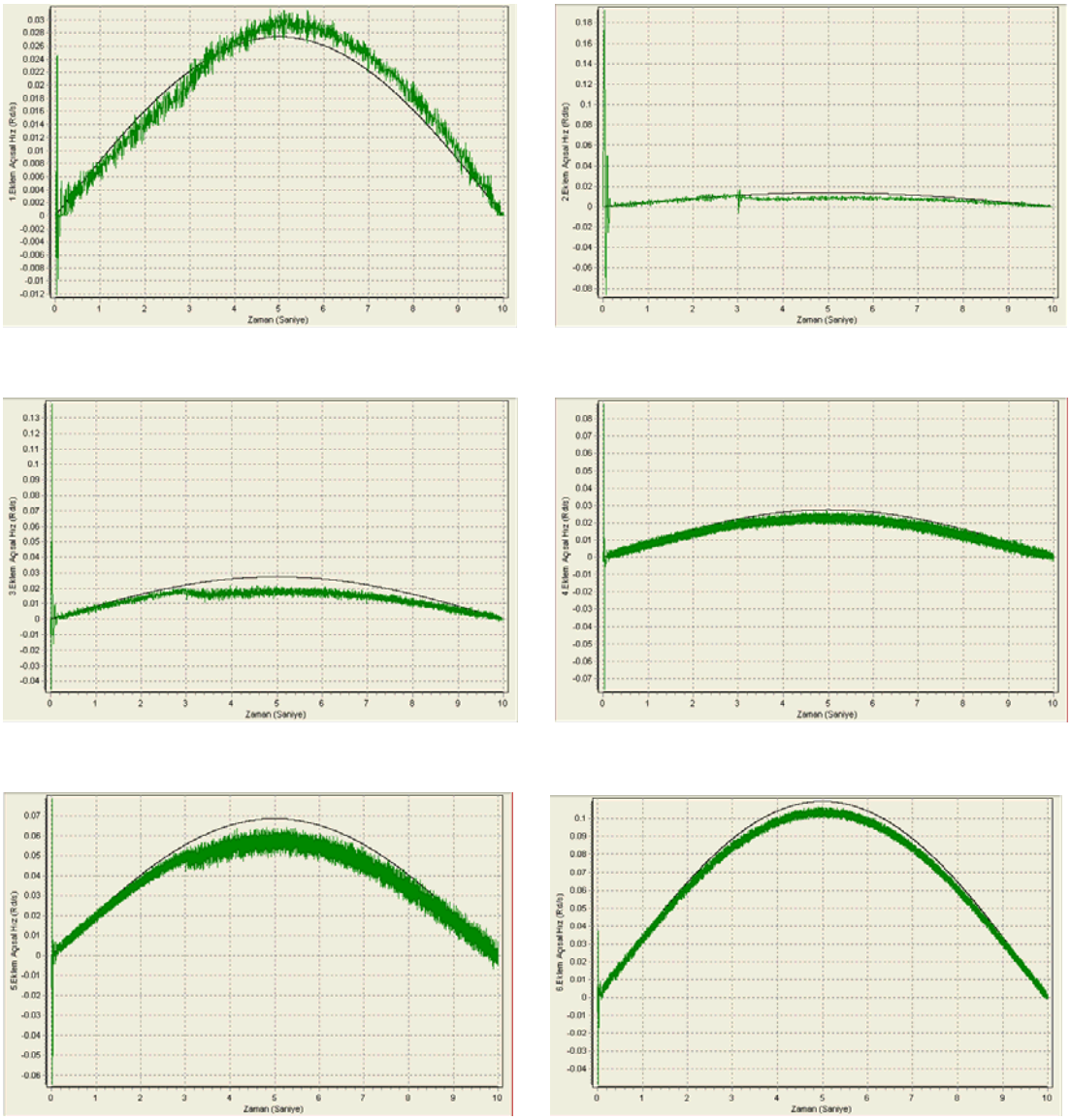
4. Durum ‘da GPC-SISO algoritmasına ait açısal yol grafiklerine bakıldığında eklemler referans yörüngelerini yakın takip etmişlerdir. İlave bozuculardan kaynaklanan salınımlar tork, açısal hız ve açısal hız hataları grafiklerinde görülmektedir.



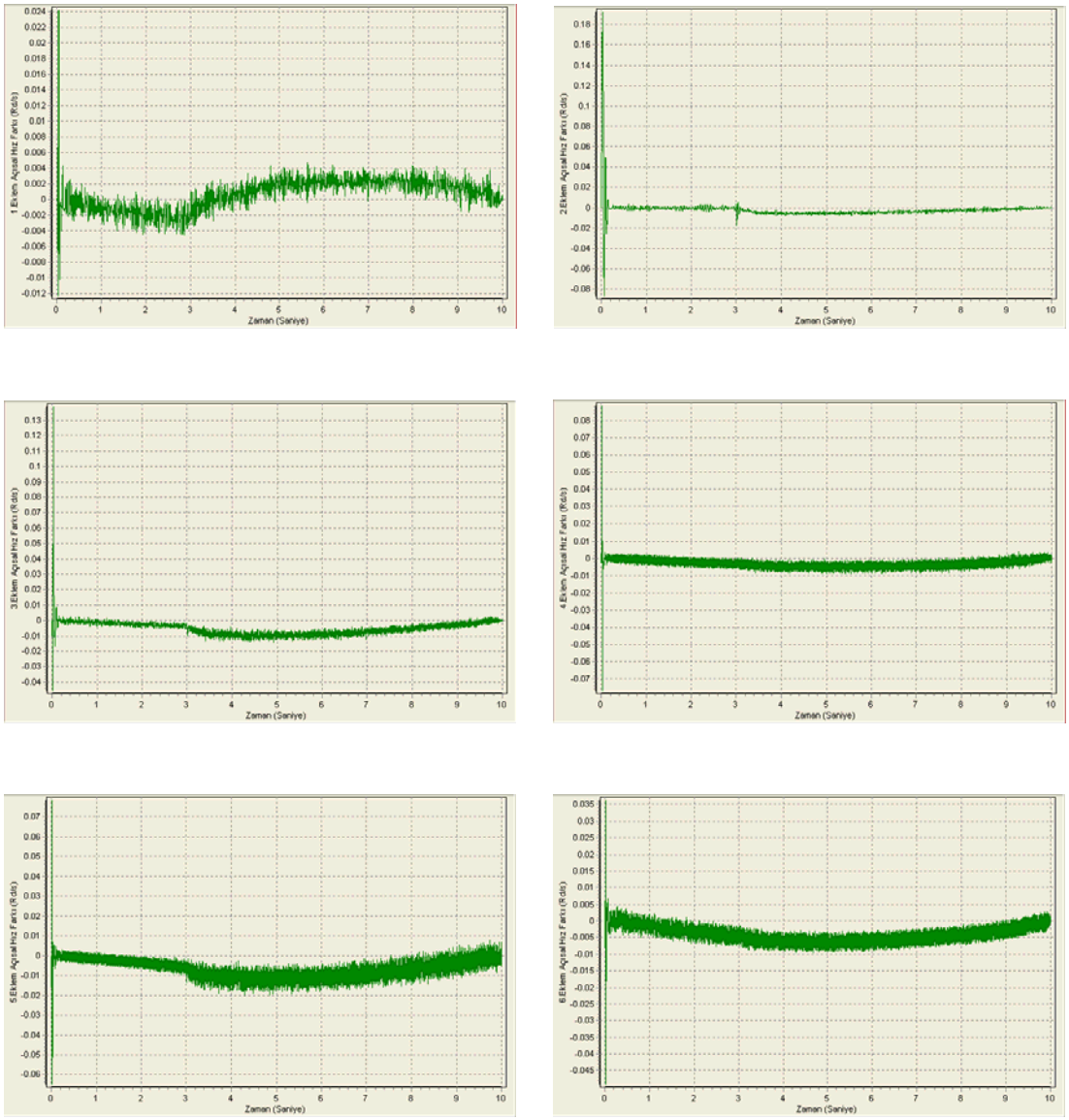

Şekil 4.115. Eklemlere uygulanan tork grafikleri (GPC MIMO, Örnek 1, Durum 4)




Şekil 4.116. Eklemlerin takip ettiği açısal yol grafikleri (GPC MIMO, Örnek 1, Durum 4)

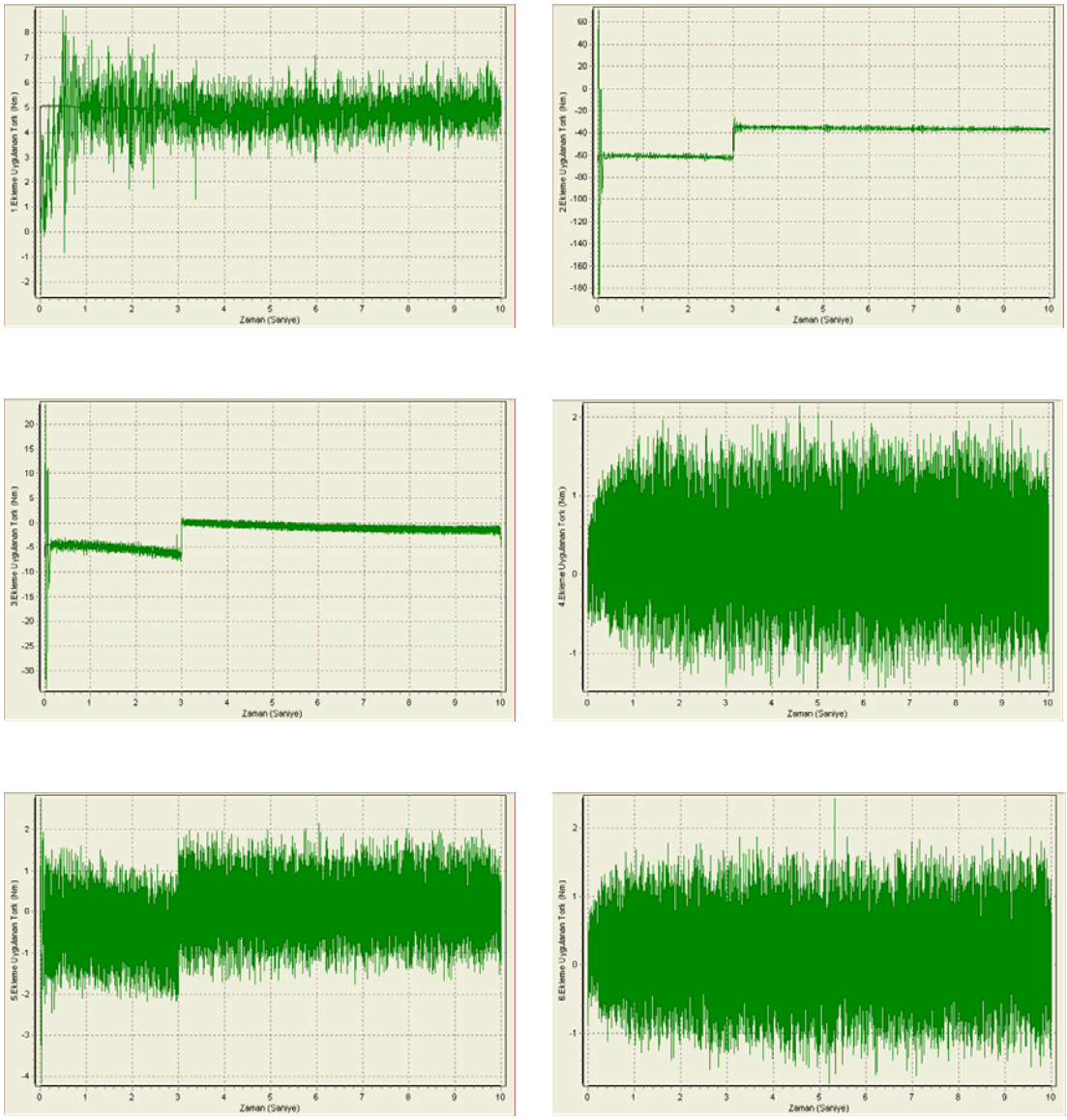



Şekil 4.117. Eklemlerin açısal hız grafikleri (GPC MIMO, Örnek 1, Durum 4)

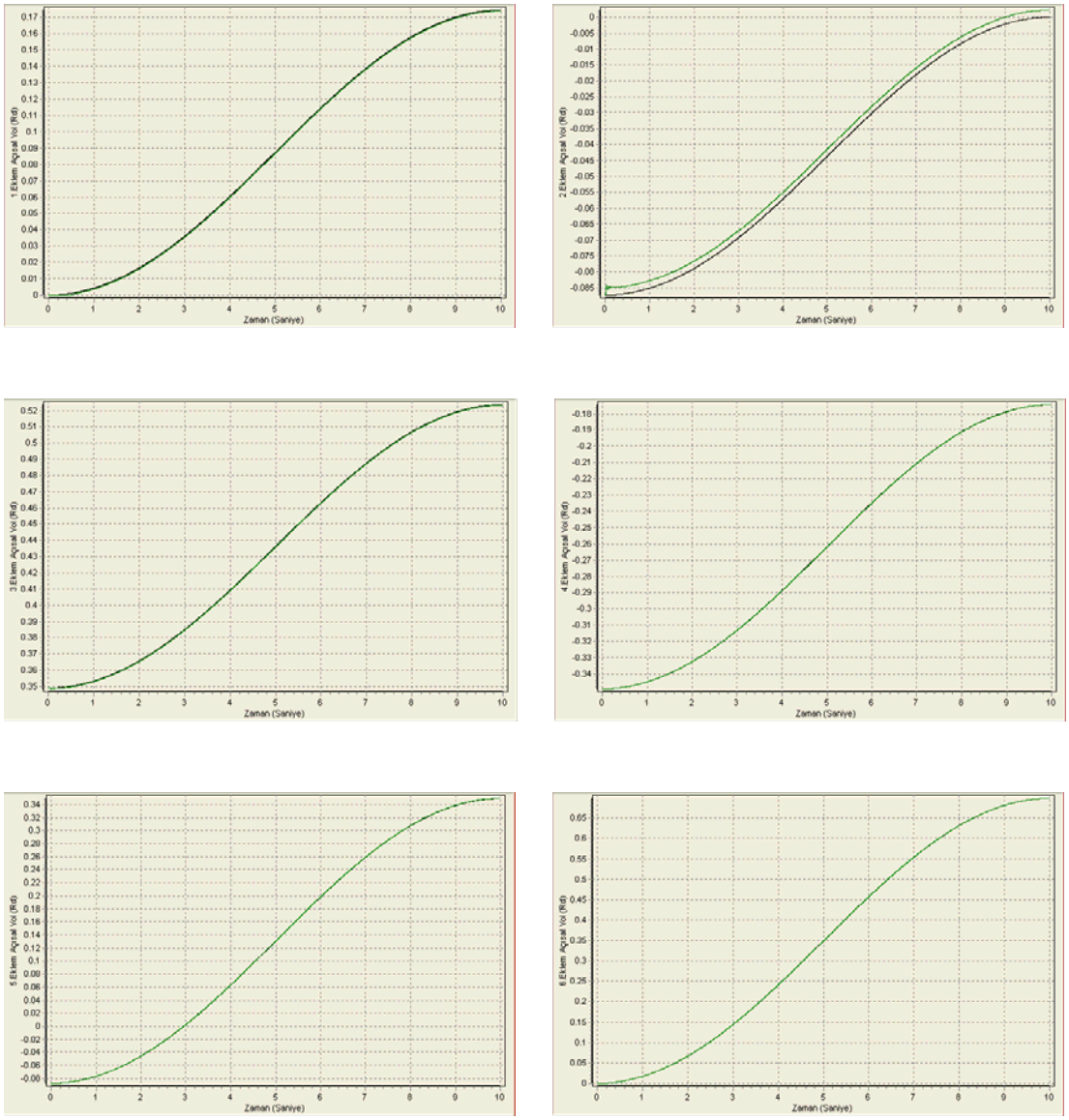



Şekil 4.118. Eklemlerin açısal hız hataları (GPC MIMO, Örnek 1, Durum 4)

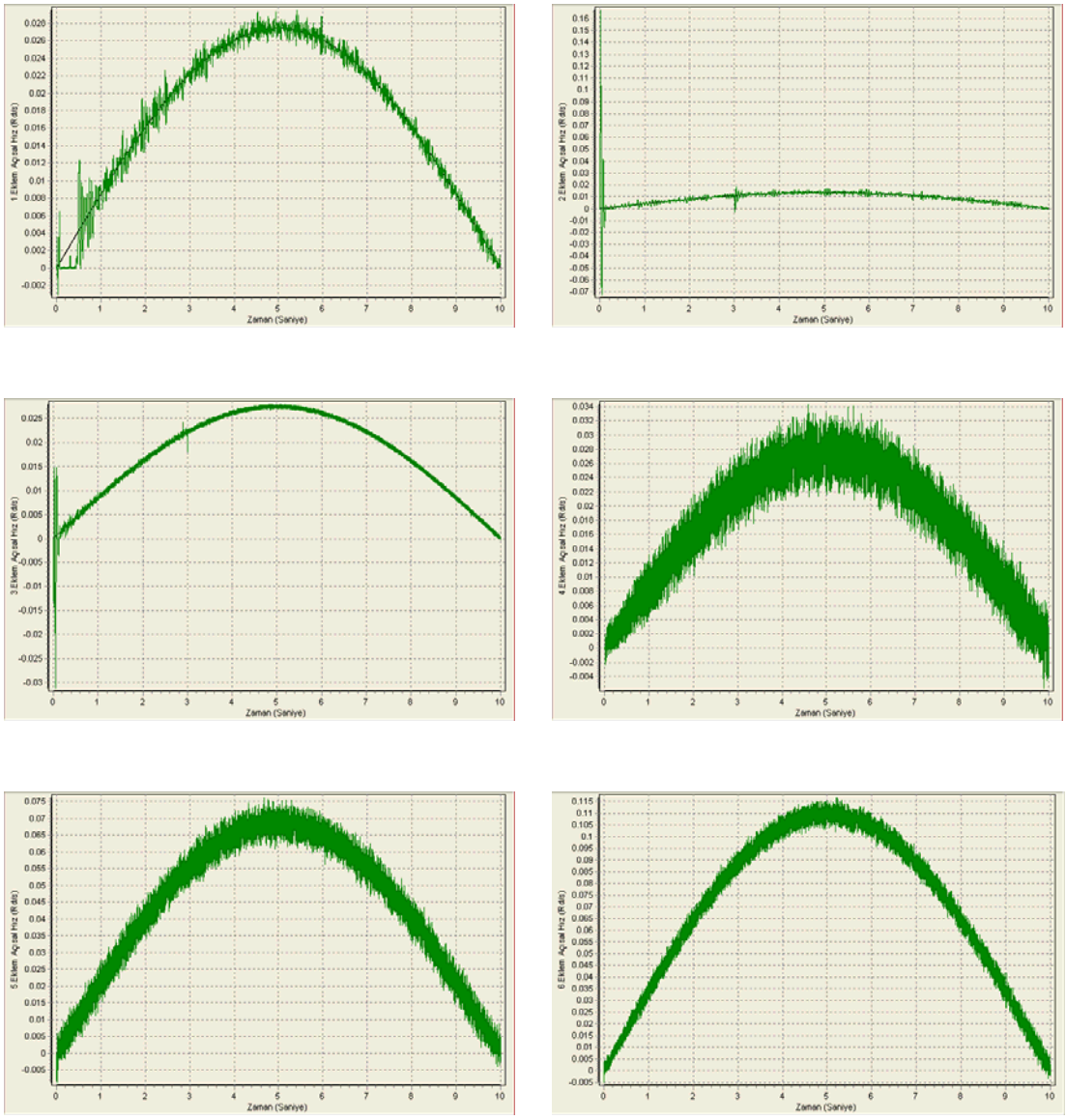
GPC-MIMO algoritması ile 4. Durum ‘da yörünge hataları oldukça büyüktür. Açısal hız eğrilerine bakıldığında eklemler hareketin sonunda referans eğrilerini yakalayabilmişlerdir.



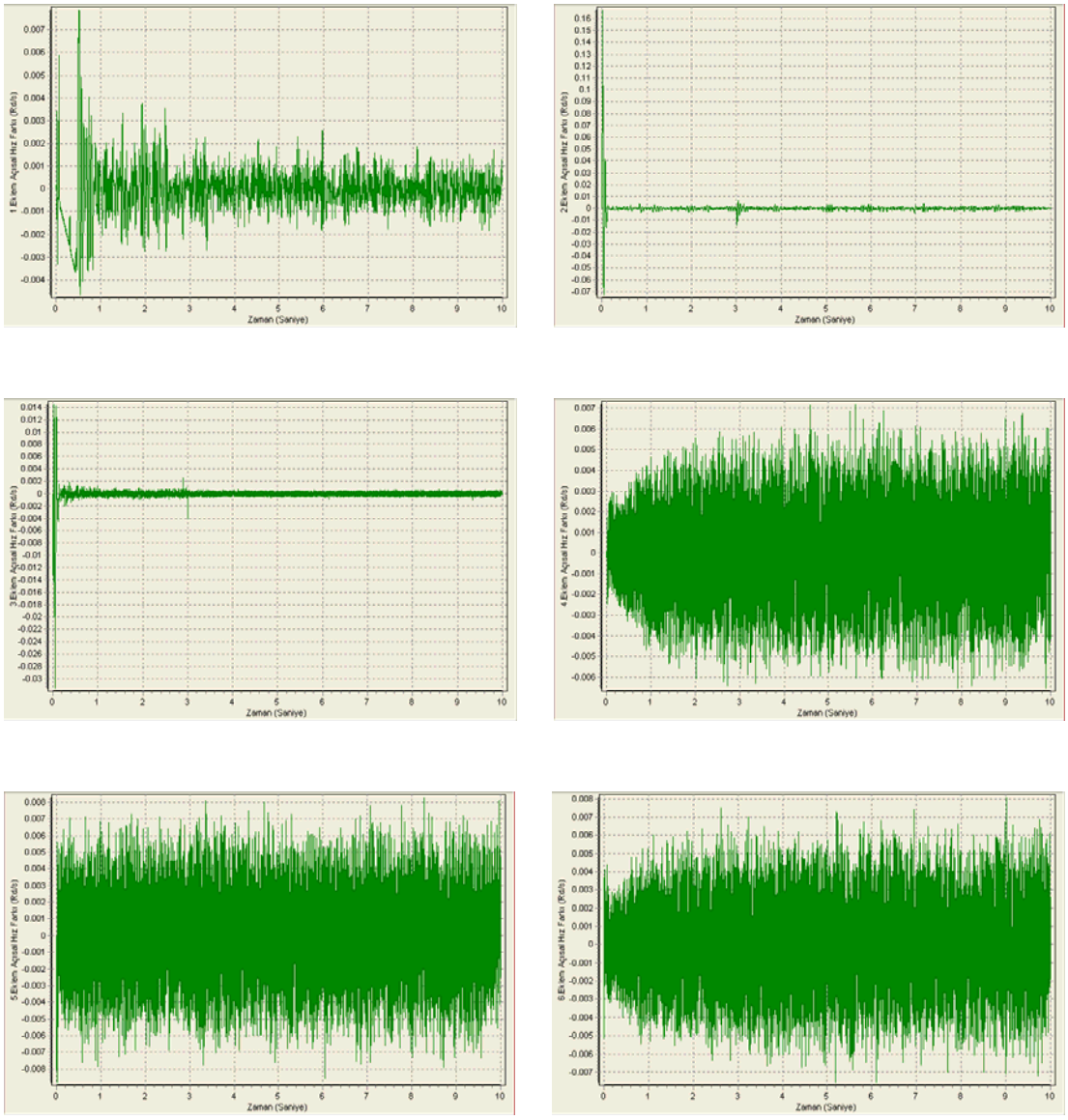

Şekil 4.119. Eklemlere uygulanan tork grafikleri (SGA-GPC SISO, Örnek 1, Durum 4)




Şekil 4.120. Eklemlerin takip ettiği açısal yol grafikleri (SGA-GPC SISO, Örnek 1, Durum 4)

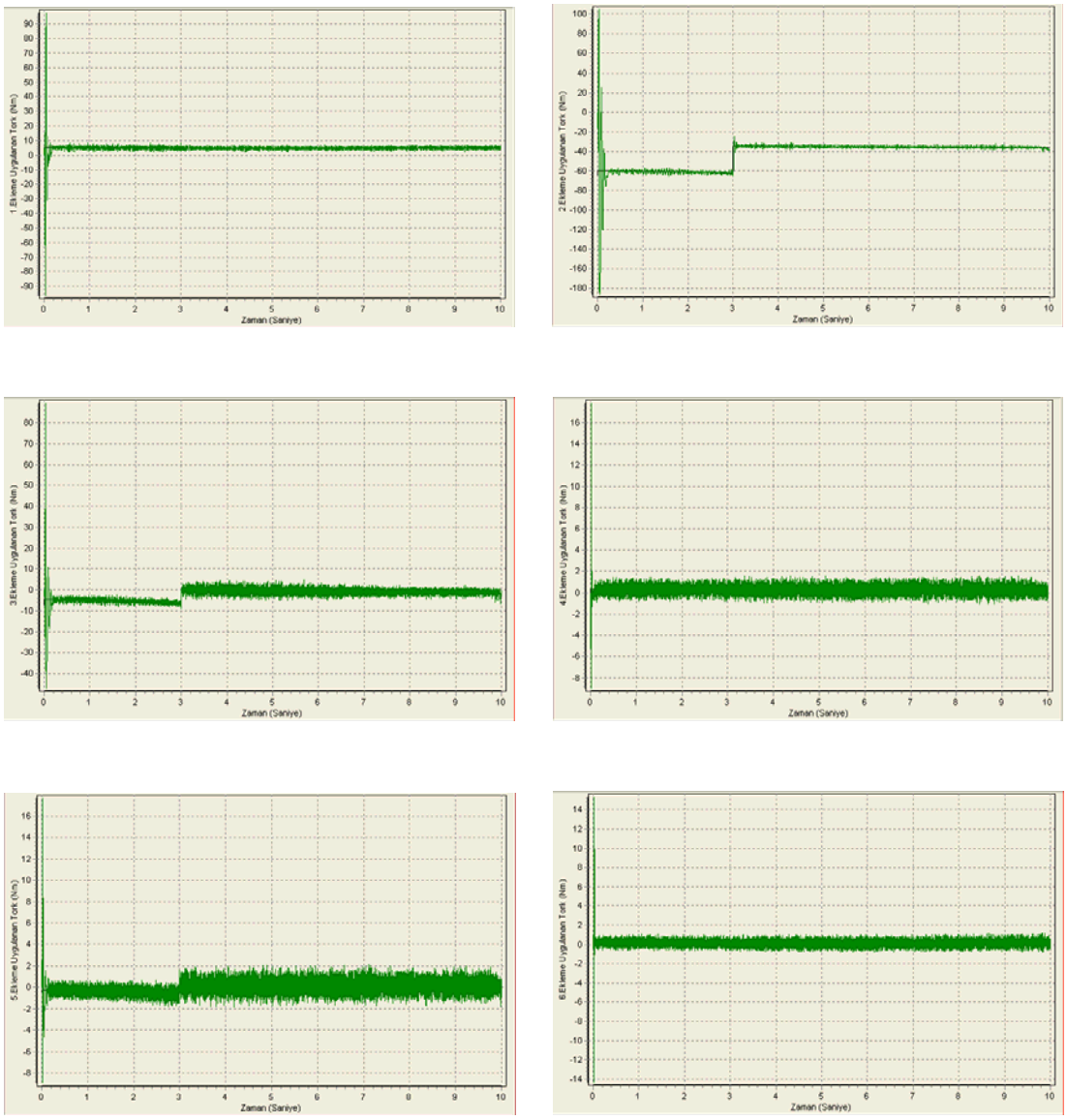



Şekil 4.121. Eklemlerin açısal hız grafikleri (SGA-GPC SISO, Örnek 1, Durum 4)

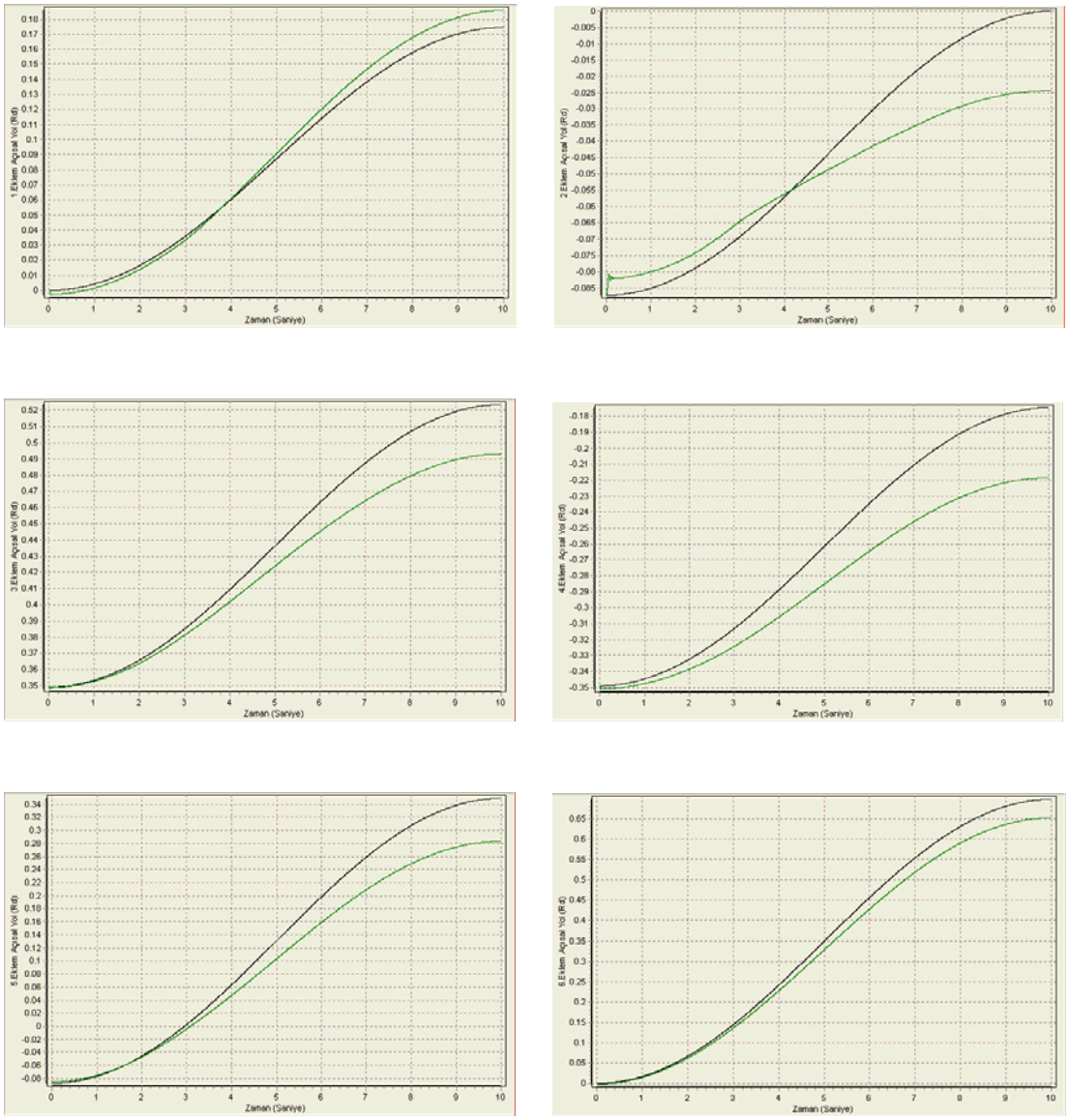



Şekil 4.122. Eklemlerin açısal hız hataları (SGA-GPC SISO, Örnek 1, Durum 4)

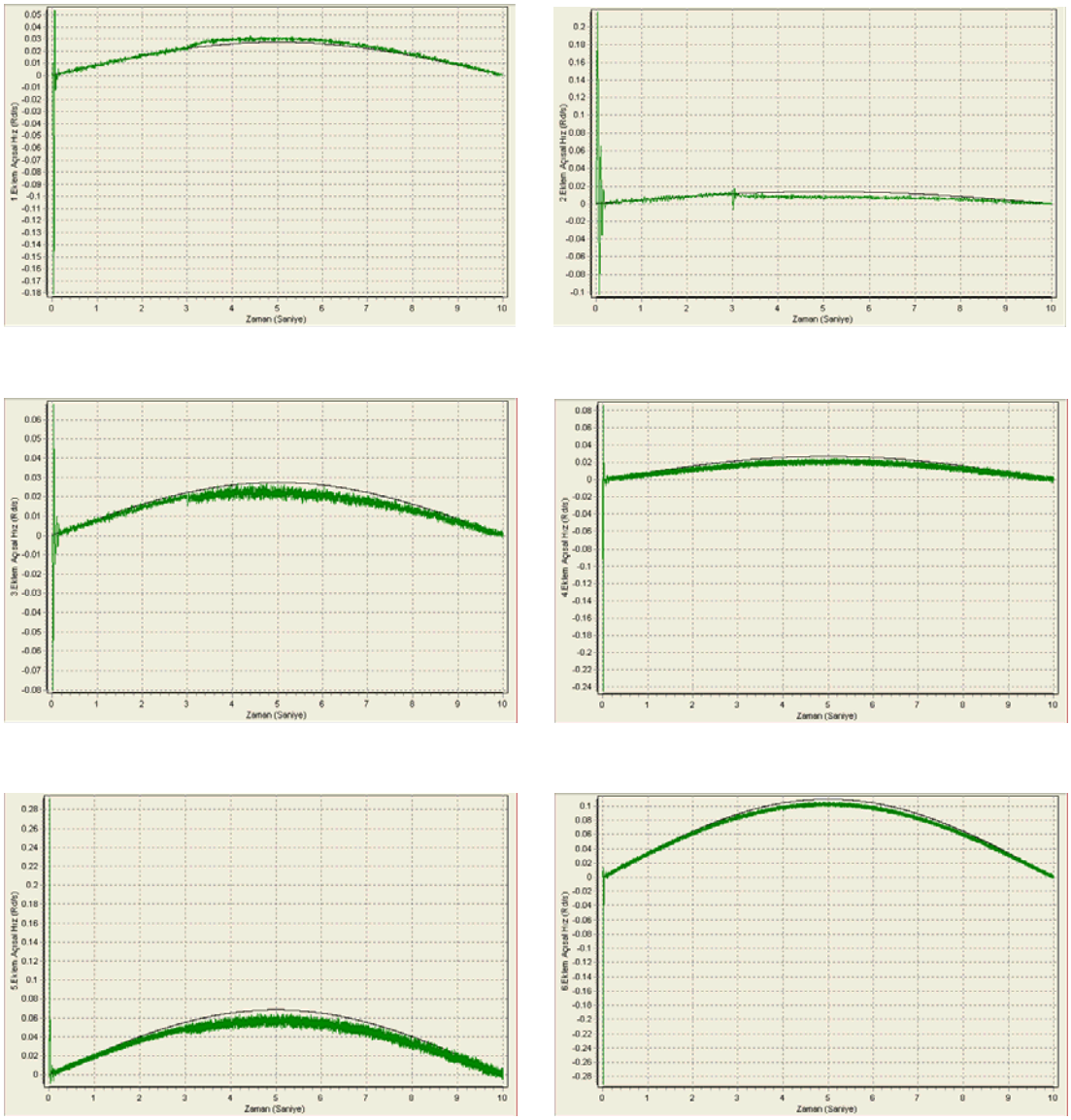
4. Durum ‘da SGA-GPC SISO algoritması ile eklemler referans yörüngelerini yakın takip etmişlerdir. Ancak, açısal hız ve tork eğrilerinde görüldüğü gibi ilave bozucular karşısında GA adaptif özelliğinin zayıf oluşu nedeniyle zorlanmıştır ve daha büyük genlikte titreşimler oluşmuştur.



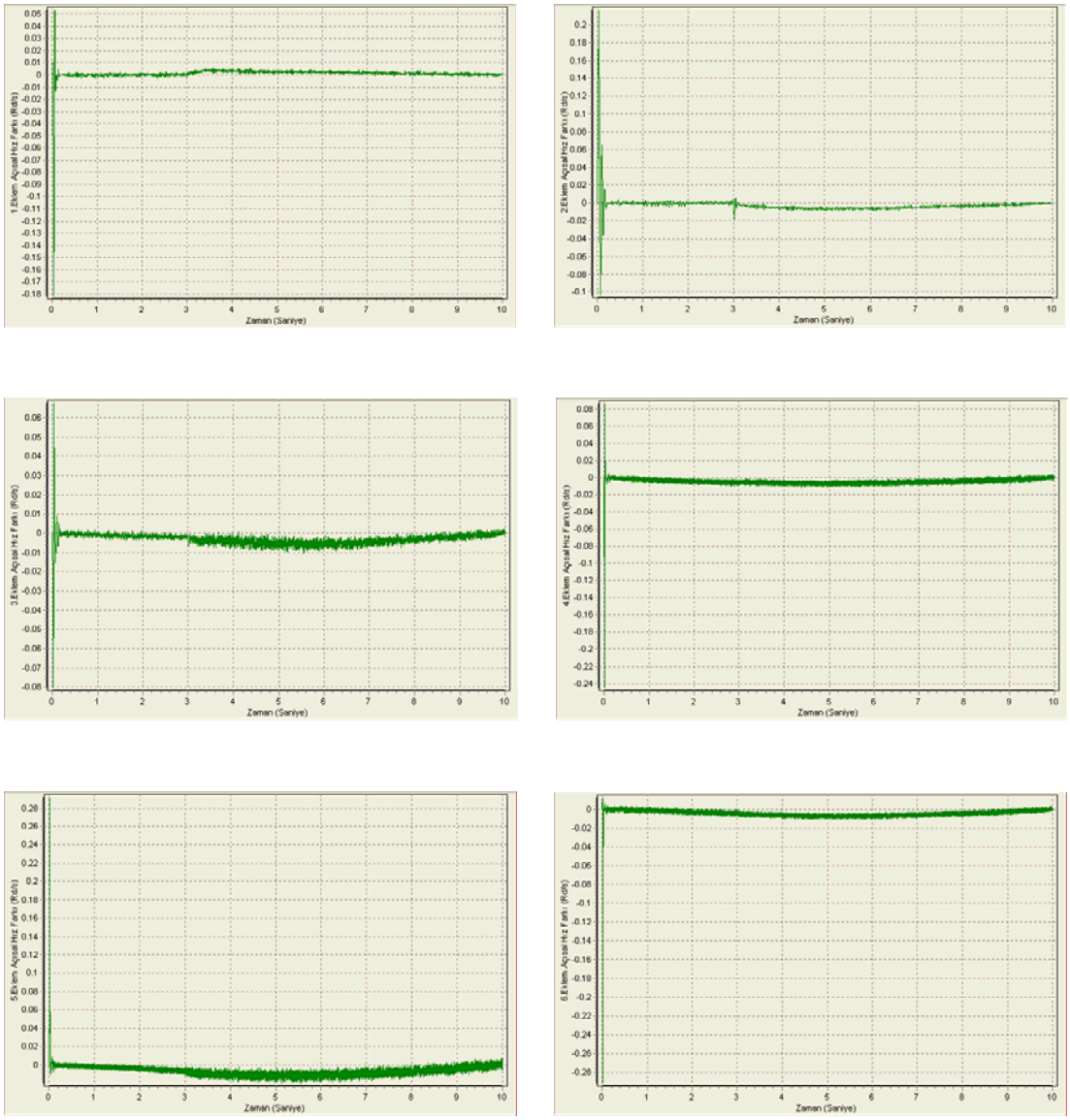

Şekil 4.123. Eklemlere uygulanan tork grafikleri (SGA-GPC MIMO, Örnek 1, Durum 4)




Şekil 4.124. Eklemlerin takip ettiği açısal yol grafikleri (SGA-GPC MIMO, Örnek 1, Durum 4)

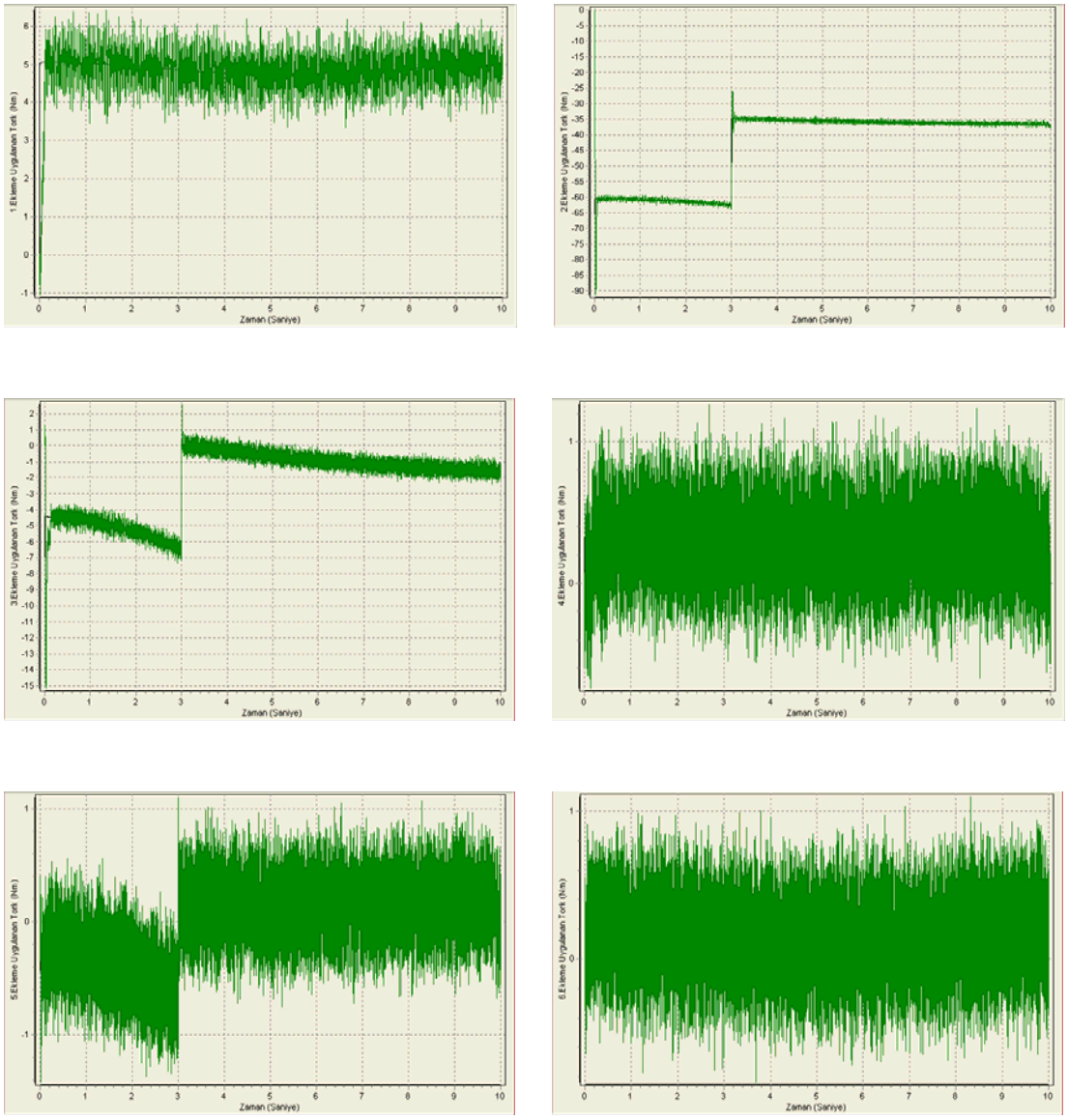



Şekil 4.125. Eklemlerin açısal hız grafikleri (SGA-GPC MIMO, Örnek 1, Durum 4)

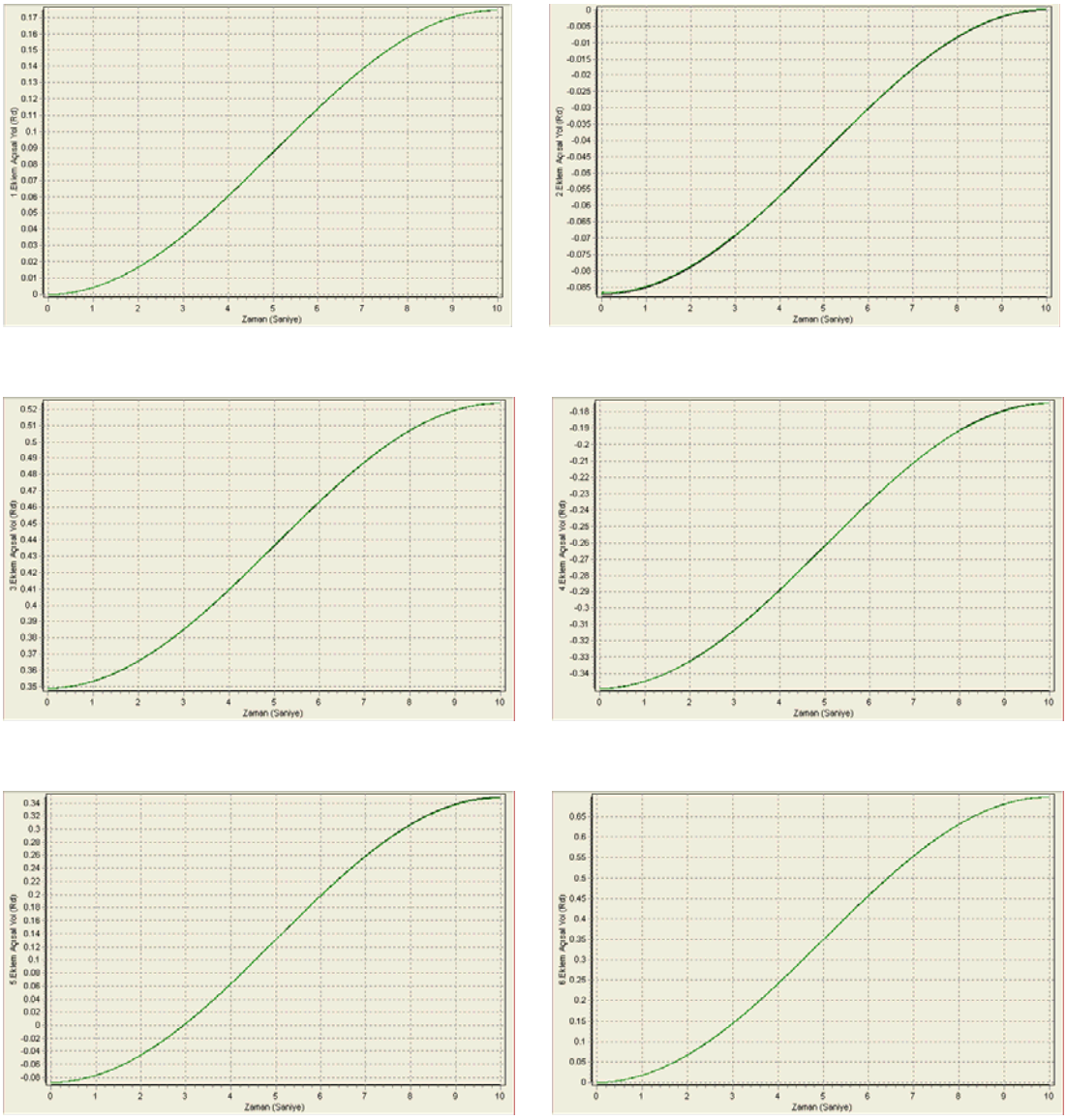



Şekil 4.126. Eklemlerin açısal hız hataları (SGA-GPC MIMO, Örnek 1, Durum 4)

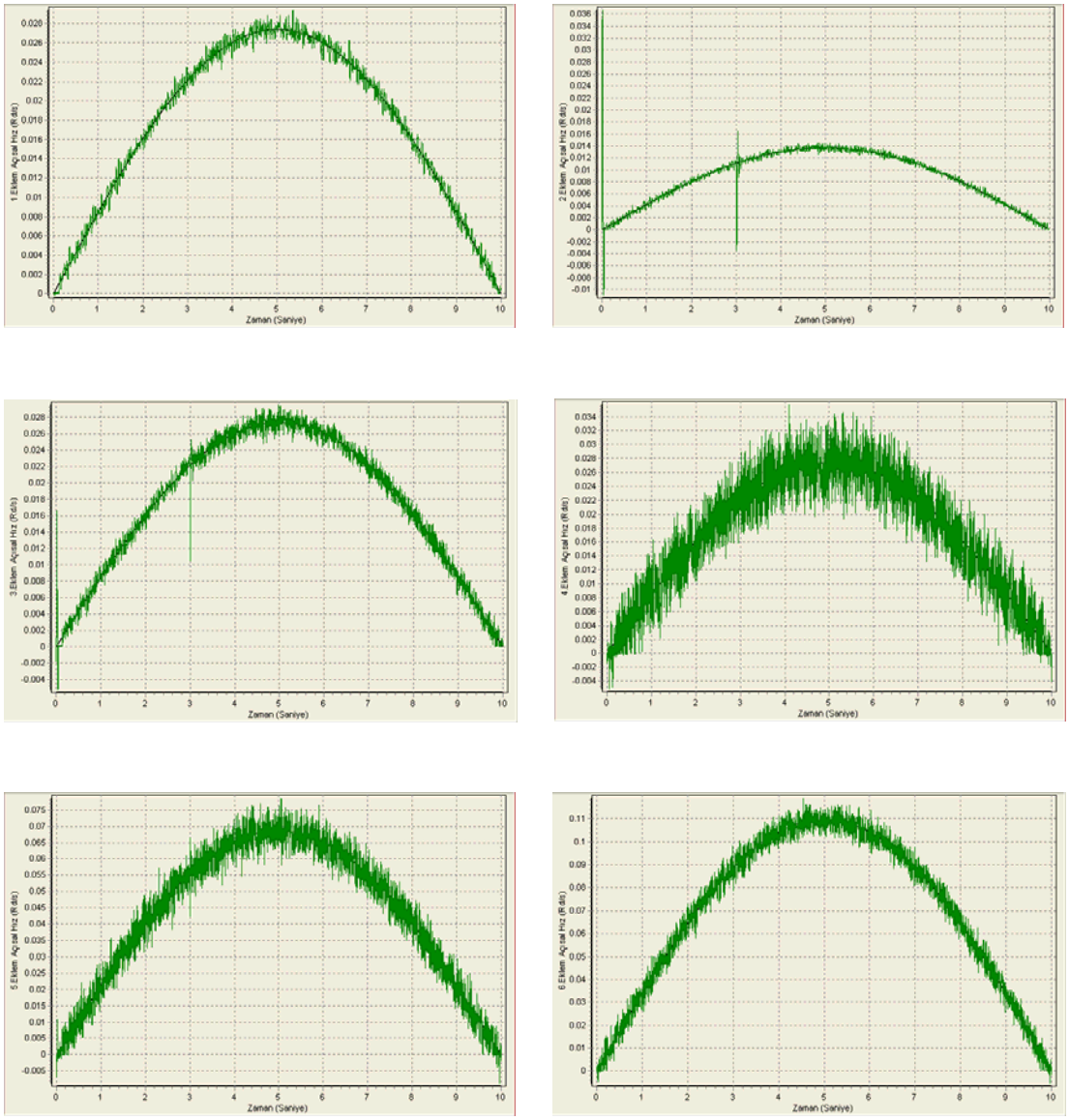
4. Durum ‘da SGA-GPC MIMO algoritması ile referans yörüngelerden sapmalar oluşmuştur. Açısal hız grafiklerinde ise referans yörüngeler hareketin sonunda yakalanmıştır.



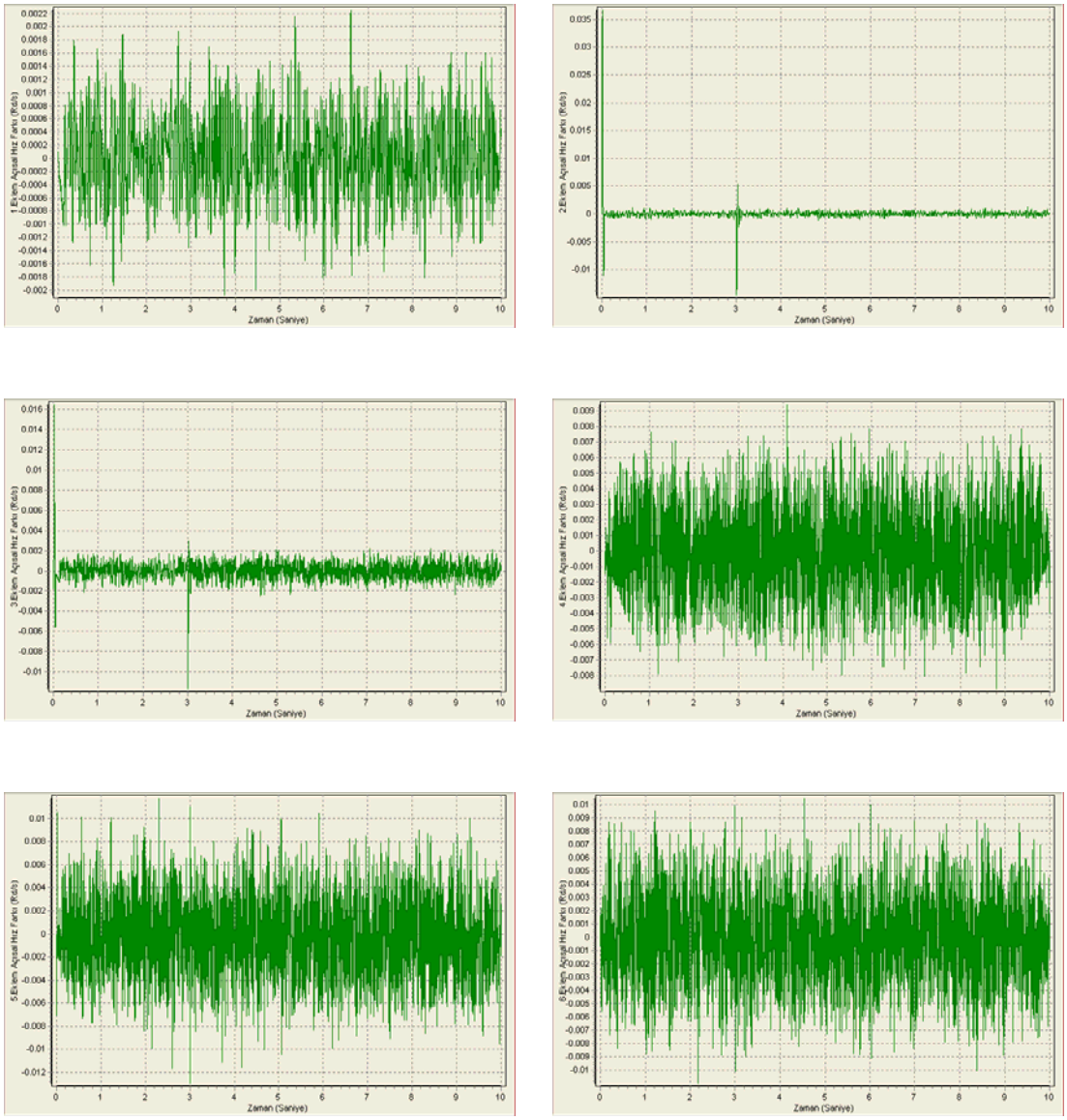

Şekil 4.127. Eklemlere uygulanan tork grafikleri (NGPC SISO, Örnek 1, Durum 4)




Şekil 4.128. Eklemlerin takip ettiği açısal yol grafikleri (NGPC SISO, Örnek 1, Durum 4)

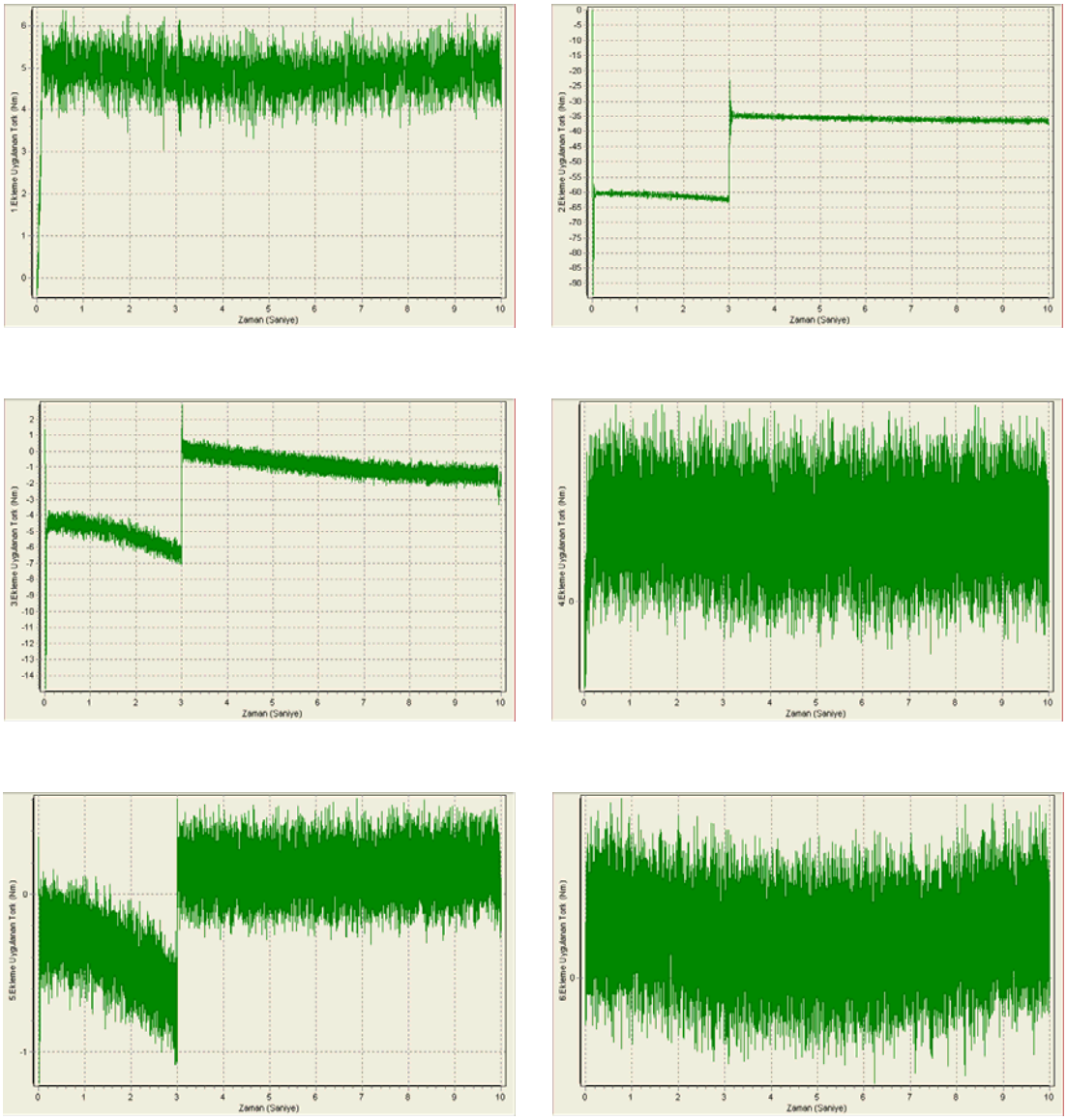



Şekil 4.129. Eklemlerin açısal hız grafikleri (NGPC SISO, Örnek 1, Durum 4)

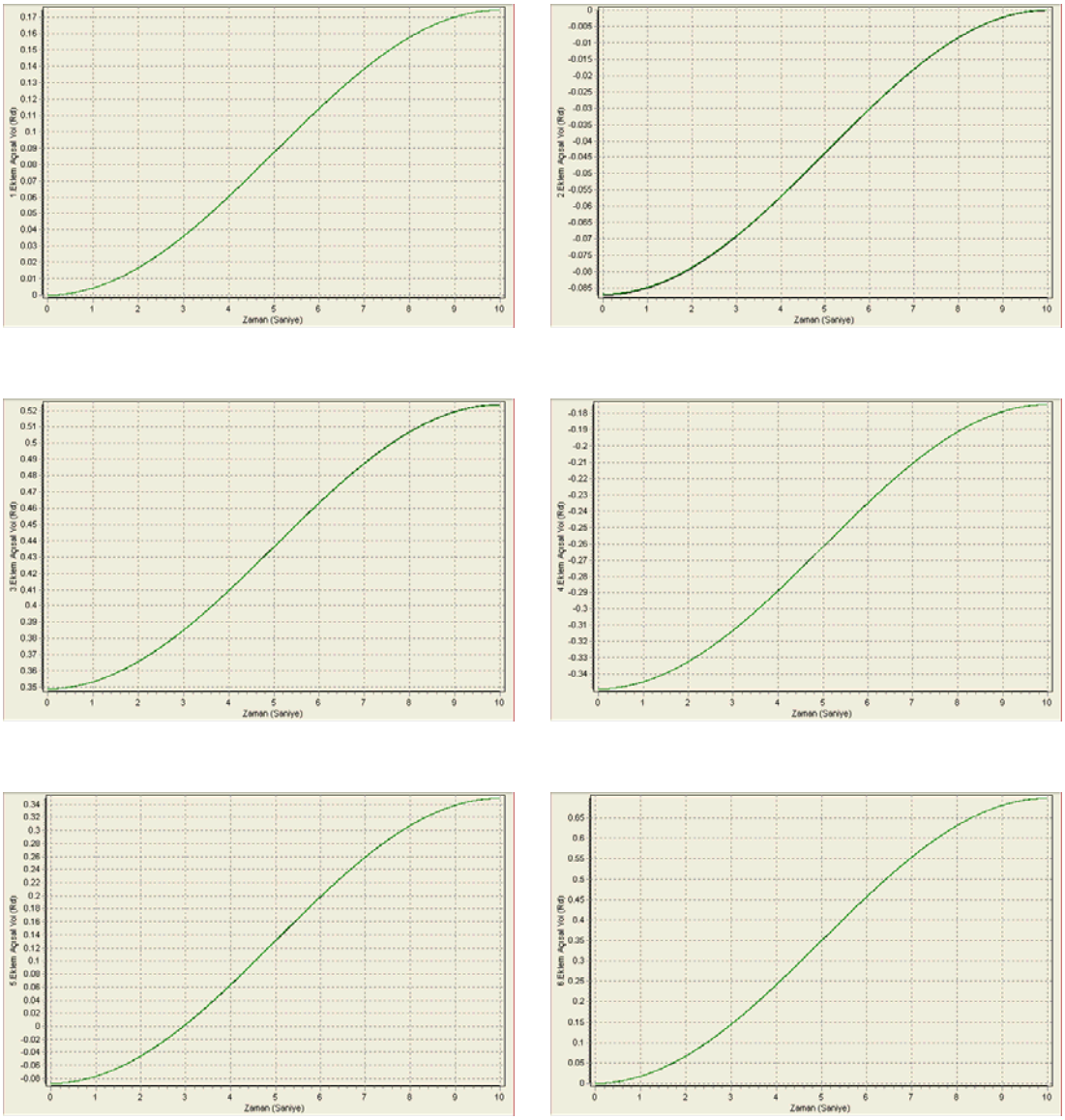



Şekil 4.130. Eklemlerin açısal hız hataları (NGPC SISO, Örnek 1, Durum 4)

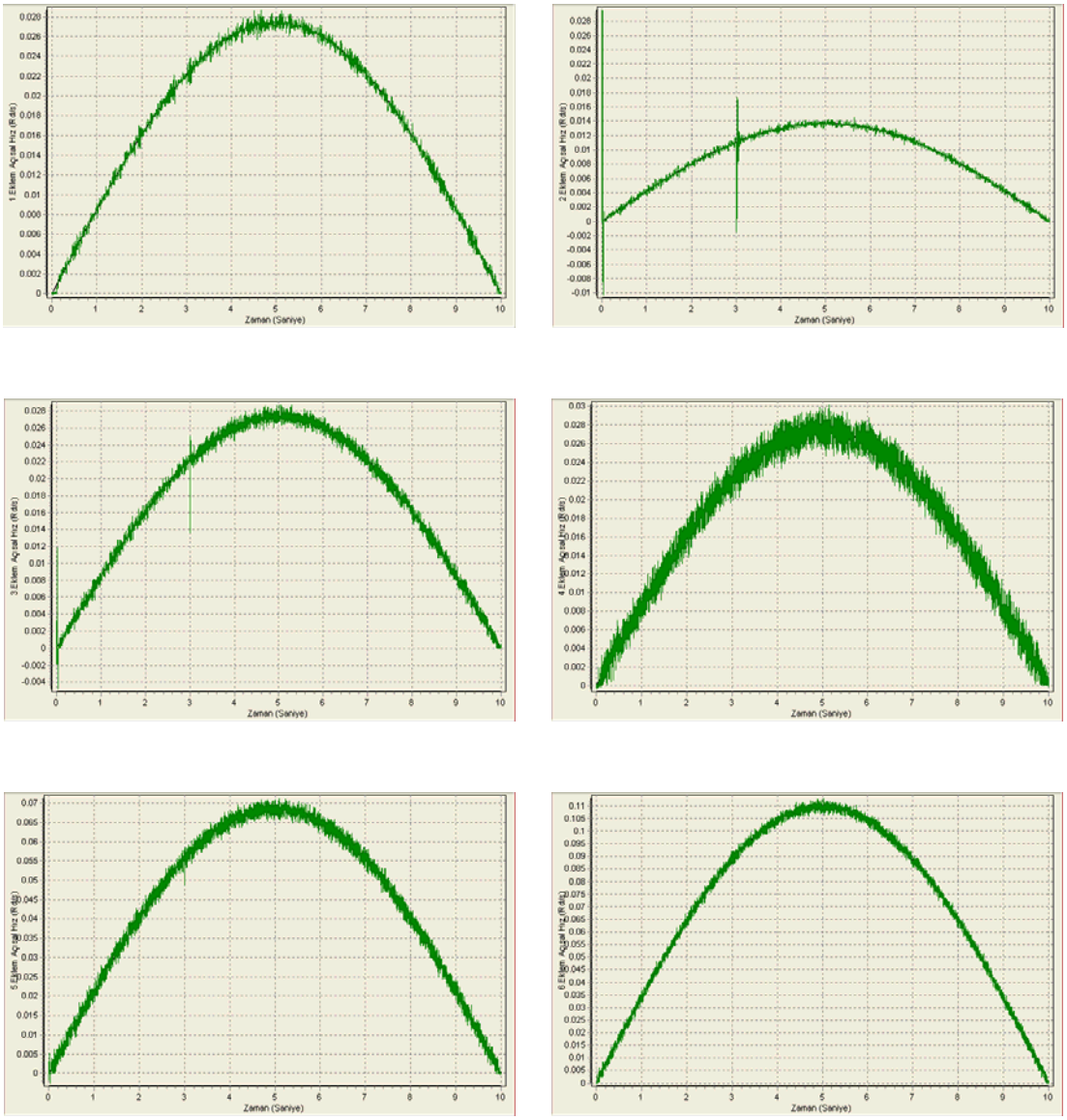
NGPC-SISO algoritması 4. Durum ‘da oldukça başarılıdır. Özellikle açısal yol grafikleri incelendiğinde eklemlerin referans yörüngelerini çok yakın takip ettiği görülmektedir. İlave bozuculardan kaynaklanan titreşimler tork ve açısal hız grafiklerine yansımıştır.



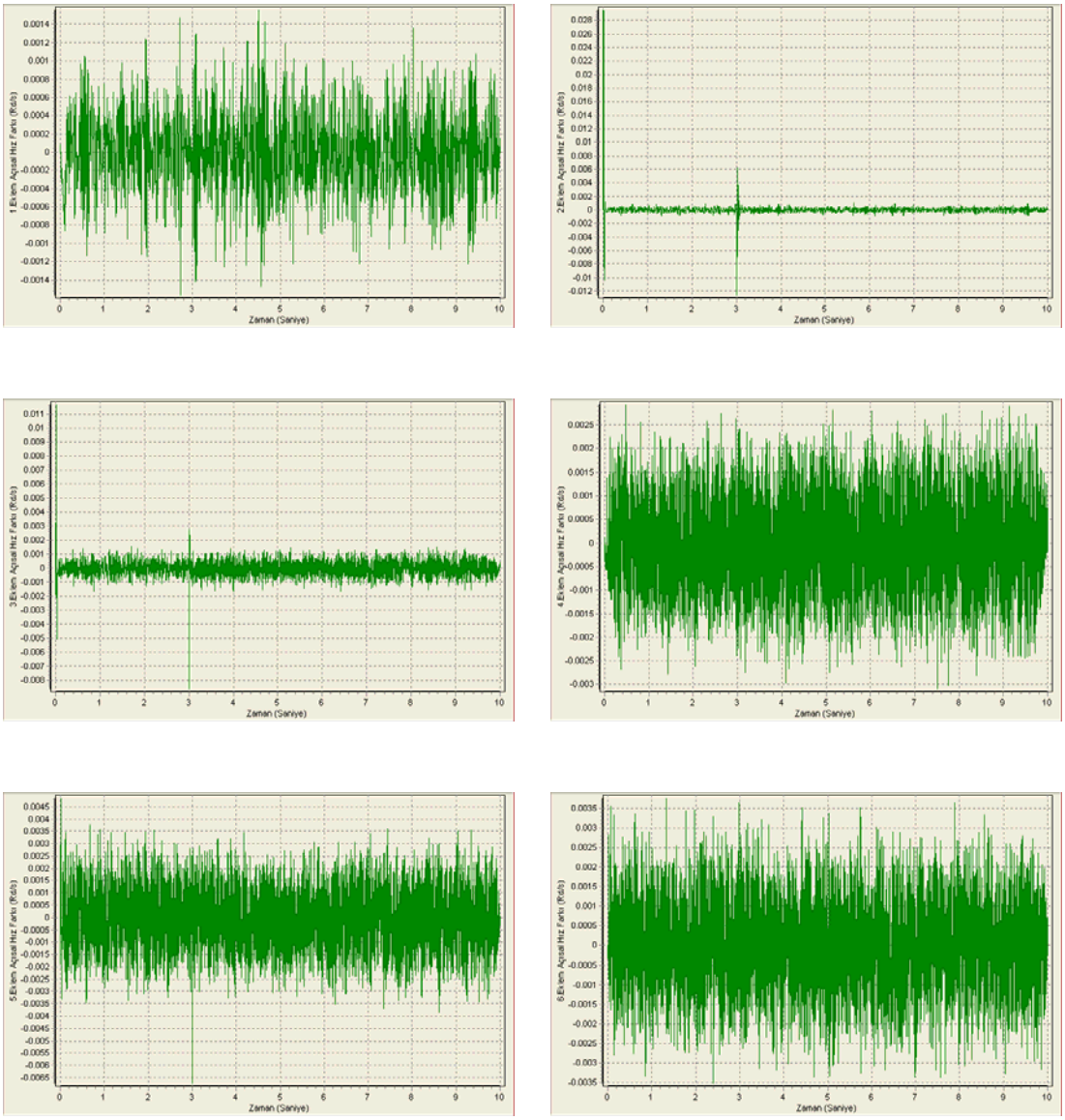

Şekil 4.131. Eklemlere uygulanan tork grafikleri (ENGPC SISO, Örnek 1, Durum 4)




Şekil 4.132. Eklemlerin takip ettiği açısal yol grafikleri (ENGPC SISO, Örnek 1, Durum 4)




Şekil 4.133. Eklemlerin açısal hız grafikleri (ENGPC SISO, Örnek 1, Durum 4)




Şekil 4.134. Eklemlerin açısal hız hataları (ENGPC SISO, Örnek 1, Durum 4)

ENGPC-SISO algoritması 4. Durum ‘da da oldukça başarılıdır. Özellikle açısal yol grafikleri incelendiğinde eklemlerin referans yörüngelerini çok yakın takip ettiği görülmektedir. İlave bozuculardan kaynaklanan titreşimler tork ve açısal hız grafiklerine yansımıştır.

4. Durum ‘a ait grafikler incelendiğinde ilave edilen rasgele değerlikteki bozucuların eklemlerin kontrollerini zorlaştırdığı açıkça görülmektedir. Özellikle MIMO yapılarda eklemlerin referans yörüngelerinden ayrıldıkları görülmüş ve bu durumun

Tablo 4.3 ‘deki veriler ışığında uç nokta koordinat hatalarına sebebiyet verdiği tespit edilmiştir.

SISO algoritmaların başarılı olduğu bu durumda eklemler ilave bozuculara rağmen referans yörüngelerini takip etmeyi başarabilmişlerdir. Özellikle ENGPC SISO ve NGPC SISO algoritmaları ile açısal hız ve tork eğrilerinde görülen bozucu tork kaynaklı titreşimler dışında, eklemler referans yörüngelerini yakın takip etmişlerdir. Fazladan bozucu ilavesi kontrol hassasiyetini çok az etkilemiştir. Sonuç olarak ENGPC SISO algoritması kullanılan algoritmalar içinde en iyisi ve başarılısıdır.

# BÖLÜM 5. TARTIŞMA VE ÖNERİLER

Model Tabanlı Öngörülü Kontrol (Model Based Predictive Control-MBPC) sınıfına ait olan Genelleştirilmiş Öngörülü Kontrol (Generalized Predictive Control - GPC), Yapay Sinir Ağlı Genelleştirilmiş Öngörülü Kontrol (Neural Generalized Predictive Control - NGPC) ile bu algoritmaları geliştirmek için önerilen Basit Genetik Algoritma uyarlamalı Genelleştirilmiş Öngörülü Kontrol (SGA-GPC) ve Yinelenen Elman Ağ uyarlamalı NGPC (ENGPC) kontrol algoritmalarının altı eklemli endüstriyel bir robotik manipülatöre eklem esaslı yörünge kontrolü için uygulanmalarıyla yapılan çalışmalar ve sonuçları bu bölümde tartışılacaktır.

Robotik manipülatörlerin dinamik kontrolü ile ilgili olarak literatürde yer alan çoğu çalışmada lineer olmayan denklemlerin karmaşıklığı vurgulanmış ve değişik metotlar geliştirilmiştir [59-60]. Önerilen bu metotlarda karmaşık hesaplamalar mevcuttur. Bu yüzden doğru yörünge kontrolü düşük hızlarda gerçekleştirilmektedir. Öte yandan öngörülü kontrol teknikleri, manipülatörlerin performansında önemli gelişmeler sağlamıştır ve manipülatörlere eğitilebilir özellikler kazandırmaktadır [41-47]. Özellikle tekrarlı görevlerde, büyük performans göstermektedir. Yüksek çalışma hızları ve taşınan yükteki değişmeler karşısında ortaya çıkan istenmeyen etkileri dengeleme özelliğine sahiptirler. Öngörülü kontrol teknikleri arasında en çok kullanılanı; Model Tabanlı Öngörülü Kontrol (Model Based Predictive Control-MBPC) yöntemidir. Hedef; bilinmeyen sistem çıkışını, verilen referans modelin çıkışına asimptotik olarak yakınsatmaktır. Bu tezde tasarlanan kontrolörler bu model temeline dayanır.

Yapılan çalışmalar iki aşamadan oluşmaktadır. İlk aşamada robotik manipülatörün dinamik davranışı oluşturulmuştur. Robot kolu eklemlerinin hareketlerini dinamik olarak modellemek için kullanılan Lagrange-Euler (L-E) [68, 69], Recursive-Lagrange (R-L) [70] ve Newton-Euler (N-E) [71] gibi pek çok yöntem mevcuttur. Bu

tezde kullanılan altı eklemli robot kolunun dinamik modelinin çıkartılmasında, kullanımı diğerlerine göre daha kolay ve sistematik olan L-E yöntemi tercih edilmiştir. Dinamik modellemeye sürtünme, yük taşıma ve taşınan yükün taşıma esnasında düşmesi durumları da ilave edilmiştir. Elde edilen dinamik model, 4. mertebeden Runge-Kutta bütünleştirme yöntemi kullanılarak robot kolu simülatörüne dönüştürülmüştür.

Robot kolunun her bir ekleminin takip etmesi istenilen referans konum ve referans hız yörüngeleri sinüzoidal yörünge esaslarına göre belirlenmiştir. Her bir eklemin açısal konum ve açısal hız yörüngeleri kendi koordinat çerçevesi temel alınarak diğer eklemlerden bağımsız olarak çıkartılmıştır. Bu sayede uç elemanın pozisyonu ve hızı, düz kinematik kullanılarak rahatlıkla hesaplanabilmektedir.

İkinci aşamada, robot manipülatörün eklem kontrolü için kontrol algoritmaları tasarlanmıştır. Bu yapı sistemin dinamik modeli ile tamamen bağımsızdır. Algoritmalar eklemlere uygulanacak tork değerlerini üretirler ve bu değerler dinamik sistemine iletilir. Dinamik modelden eklemlere ait geri besleme konum ve açısal hız bilgileri alınarak bir sonraki kontrol adımında kullanılmak üzere tekrardan kontrolörlere giriş bilgisi olarak alınır. Bu işlem hareket tamamlanana kadar tekrarlanır.

Bu tezde GPC, SGA-GPC, NGPC ve ENGPC olmak üzere toplam dört adet kontrol algoritması hem tek giriş-tek çıkış (SISO) hem de çok giriş-çok çıkış (MIMO) olarak tasarlanmıştır. SISO yapılarda eklemler birbirinden bağımsız olarak düşünülmüş ve her biri için yapı olarak benzer çalışma olarak birbirlerinden bağımsız altı adet kontrolör tasarlanmıştır. MIMO yapılarda ise altı girişli-altı çıkışlı bir sistem olmak üzere tek bir kontrolör tasarlanmıştır.

Referans model eklemlere ait girilen açı değerlerine göre eklemlerin hareket süresince takip edeceği referans konum ve referans açısal hız yörüngelerini üretir. Kontrol algoritmaları her kontrol adımında bu referans yörüngeleri takip edecek uygun eklem tork değerlerini hesaplarlar. Dolayısıyla kullanılan kontrol algoritmaları dinamik sistemden elde edilen geri besleme eklem açısal hızlarını kendilerine girdi

olarak almakta ve referans yörüngeler ışığında eklemlere uygulanacak torkları çıktı olarak üretmektedirler.

Altı eklemli endüstriyel bir robotik manipülatörün eklem esaslı yörünge kontrolünün yapıldığı bu çalışmada, tasarlanan kontrol algoritmalarının performansı eklemlere ait tork, açısal yol, açısal hız, açısal hız hataları grafikleri ile eklemlere ait açı hataları, açısal hız hatalarının kareleri ve uç nokta konum hataları üzerinden hem grafiksel hem de nümerik sonuçlarla karşılaştırılmıştır.

Simülasyon sonuçlarına göre kontrol algoritmaları NGPC MIMO ve ENGPC MIMO dışında genel olarak başarılıdır. Özellikle SISO yapılarda tasarlanan algoritmalar referans yörüngelerini çok yakın takip etmişlerdir ve çok küçük açı hataları ile hareketlerini tamamlamışlardır. MIMO yapılarda ise yörüngelerden sapmalar oluşmuştur. MIMO tasarımın karmaşıklığı nedeniyle bu beklenen bir sonuçtur. Yine de SGA-MIMO algoritması GPC MIMO ‘ya kıyasla konum hatalarını azaltmış ve hareketin sonlarına doğru eklemlerin referans yörüngelerini yakalayabilmelerini sağlamıştır. Sonuç olarak, Genetik Algoritma MIMO tasarımlar için iyileşme sağlamıştır.

Robot kol kontrolü için önemli olan robot kolu uç elemanının her hangi bir başlangıç pozisyonundan istenen bir bitiş pozisyonuna sarsıntısız bir şekilde en az hata ile gidebilmesidir. Her bir eklemin açısal hız hatalarının kareleri toplamı ne kadar az olursa robot kolundaki sarsıntı o derece az olur. Yine, eklemlerin açısal konum hataları ne kadar az ise robot kolunun uç elemanın hedef noktaya uzaklık hatası o derece azdır. Kullanılan algoritmalar içerisinde ENGPC SISO algoritması bu şartları diğer algoritmalara kısayla çok daha fazla sağlamaktadır. Bunu sırasıyla NGPC SISO, SGA-GPC SISO, GPC SISO, SGA-GPC MIMO ve GPC MIMO algoritmaları takip etmektedir. NGPC MIMO ve ENGPC MIMO algoritmaları ise eklemler arasındaki aşırı etkileşimlerden dolayı kontrol esnasında kilitlendiğinden bu algoritmalarla sonuç alınamamıştır.

GPC algoritması doğrusal öngörülü sistem modeli ve maliyet fonksiyonu olmak üzere iki ana kısımdan oluşur. Maliyet fonksiyonu minimize edilerek tork değişim

vektörü hesaplanır. Tork değişim vektörü içinde doğrusal öngörülü sistem modelinin parametreleri mevcuttur. Bu parametrelerin her kontrol adımı sonunda tekrardan güncellenmesi gerekmektedir. Bu iş için bir parametre kestirim yöntemine ihtiyaç vardır. Ardışık en küçük kareler (Recursive Least Squares - RLS) [74], ardışık karekökler (Recursive Square Roots - RSQR) [75] ve ardışık U-D faktörizasyon [76] gibi pek çok parametre kestirim yöntemi mevcuttur. Bu tezde, ardışık en küçük kareler yöntemi (RLS) tercih edilmiş ve başarılı sonuçlar alınmıştır. Tercih sebebi en fazla kullanılan yöntem olmasıdır. Diğerleri de kullanılabilir ve kendi aralarında kıyaslamalar yapılabilir.

Yapısındaki üç katmanlı yapay sinir ağ nedeniyle öngörülü sistem modeli doğrusal olmayan NGPC algoritması da aynı GPC algoritmasındaki gibi iki önemli kısımdan oluşur. Öngörme işlemi yapay sinir ağı üzerinden yapılır. Maliyet fonksiyonu minimizasyonu için Newton-Raphson güncelleme metodu kullanılmıştır. Hesapsal yükü az ve kontrol hassasiyeti en fazla olması nedeniyle bu metot seçilmiştir. Bu işlem için kullanılabilecek Non-gradient [78], Simplex [79], Successive Quadratic Programming [90, 91] ve Newton-Raphson [63-65] gibi pek çok güncelleme metodu mevcuttur. Bu metotlar da seçilebilir ve kendi aralarında kıyaslanabilir.

Basit genetik algoritma uyarlamalı GPC (SGA-GPC) algoritması yapı olarak GPC algoritmasına çok benzerdir. Öngörü modeli doğrusaldır, maliyet fonksiyonu optimizasyonu ise rastlantısal bir algoritma olan genetik algoritma ile yapılmıştır. Genetik algoritmalarının robot kolu uygulamalarında yörünge planlama tekniği olarak kullanıldığı literatürde yer almaktadır [86, 87]. Genetik algoritmanın öngörülü kontrol maliyet fonksiyonu optimizasyonu olarak kullanılması bu çalışmaya özgünlük kazandırmıştır.

Basit genetik algoritma işlemlerinde uygunluk değeri ile seçme, tek noktalı çaprazlama ve tek nokta mutasyon operatörler seçilmiştir. Bunların yanı sıra turnuva, rulet tekeri ile seçim, iki noktalı, üniform, sıralı çaprazlama gibi genetik operatörlerde mevcuttur [85]. Bu operatörlerde robotik uygulamalar için kendi aralarında karşılaştırılabilirler.

Dinamik sistemlerin büyük çoğunluğu doğrusal olmayan özellikler göstermektedir. Bu durum, GPC algoritması için yapısındaki öngörülü sistem modelinin doğrusal olması nedeniyle bir dezavantaj olarak görülebilir. Bununla beraber, GPC algoritması uzun menzilli öngörülü bir algoritmadır. Hem SISO hem de MIMO uygulamaları ile yapılan çalışmalardan başarılı sonuçlar elde edilmiştir. Yapısında yapay sinir ağı bulunduran NGPC ise doğrusal olmayan dinamik sistemler için doğrusal olmayan bir öngörü modeli sunmaktadır. NGPC, yapay sinir ağı ile öngörülü kontrolün avantajını bütünleştirmiştir.

Öte yandan NGPC algoritmasında var olan yapay sinir ağı için dinamik bir ağ olan yinelenen elman ağı önerilmiştir. Literatürde şimdiye kadar yer alan NGPC uygulamalarında ileri beslemeli ağlar gibi statik ağlar kullanılmış olup, NGPC için dinamik ağ kullanımı çalışmaya özgünlük kazandırmıştır. Ayrıca önerilen dinamik ağ uyarlamalı NGPC algoritması statik ağa sahip NGPC ‘ye göre daha iyi bir performans göstermiştir.

SISO mantığına göre tasarlanan algoritmalar daha basit olduklarından geliştirilmeleri daha kolaydır. Hesapsal yükleri daha az olduklarından gerçek zamanlı kontrol uygulamaları için daha uygundurlar. Ayrıca, her bir eklem ayrı bir kontrol algoritması ile kontrol edildiğinden paralel programlama için uygundurlar.

Her bir eklemin diğer eklemlerle dinamik olarak etkileşimli olması nedeniyle algoritmaların MIMO mantığına göre tasarlanması daha kapsamlı bir yaklaşım gibi görülebilir. Fakat bu etkileşimler eklem hızlarına yansıdığından algoritmaların SISO olarak tasarlanmasında bir sakınca yoktur. Yapılan simülasyon çalışmaları SISO tasarımın robot kolu kontrolünde daha iyi sonuçlar verdiğini göstermiştir. Öte yandan paralel hesaplamanın günümüzde yaygın hale gelmesi ile yapı olarak MIMO, çalışma prensibi olarak ise paralel bilgi işlemeyi kullanan tek kontrolör içinde SISO algoritmalardan oluşan tasarımlar da yapılabilir. Özellikle genetik algoritma kullanımında uzun işlem süresi gerçek-zaman uygulamaları için problem teşkil edeceği düşüncesiyle işlem operatörleri sınırlandırılmış olduğundan, paralel çalışan genetik algoritmaların bu sınırlandırmayı ortadan kaldıracağı ve işlem süresini azaltacağı hedeflenmektedir.

Kontrol algoritmalarının adaptif özelliklerinin sınanması için kontrolün belirli bir adımında taşınan yükün düştüğü varsayılmıştır. Yük düşmesi ile oluşan yeni dinamikler ve istenmeyen durumlar karşısında algoritmaların tepkileri Bölüm 4.2 ‘de sunulan eklemlere ait tork, açısal hız ve hız hataları grafikleri üzerinden değerlendirilmiştir. Grafiklere göre hareketsiz durumdaki robot kolunun eylemsizliği nedeniyle hareket başlangıcında ve sistem dinamiğinin değişmesiyle yük düşme anında yörüngelerde ani değişimler olmakta ve farklı genliklerde titreşimler oluşmuştur. Titreşimlerin genliği, sayıları ve sistem oturma süreleri incelendiğinde; NGPC SISO ve ENGPC SISO algoritmaları ile en az genliğe ve sayıya sahip titreşimler oluşmuştur ve sistem oturma zamanları çok daha kısa gerçekleşmiştir. Bunun anlamı yapay sinir ağına sahip algoritmalar diğer algoritmalara oranla değişken yük karşısında daha dengeleyici ve adaptif bir davranış sergilemişlerdir. Önerilen dinamik ağa sahip ENGPC algoritması, hızlı ağırlık güncelleme özelliği ile hareket başlangıcında ve yük değişimleri gibi belirsiz durumlar için öngörü hataları küçültmüş ve uç nokta koordinat hatalarında statik ağa sahip NGPC ‘ye göre daha başarılı olmuştur. Ayrıca değişen sistem dinamikleri karşısında daha hızlı adaptasyon sağlamıştır. ENGPC değişen yük durumunda statik ağlı NGPC ‘ye göre sistem oturma zamanını %50 oranında kısaltmıştır.

Kontrolün güçleştirildiği rasgele bozucuların eklendiği durumda eklemlere ait tork ve hız grafiklerinde titreşimler oluştuğu ve bu durum doğal olarak eklemler üzerinde titreşimlere sebebiyet verdiği gözlemlenmiştir. Ayrıca açı hatalarının oluştuğu ve bununda konum hatalarına neden olduğu bu durumda, kontrol güçlüğüne rağmen kontrol algoritmaları genel olarak başarılı bulunmuştur.

Altı eklemli endüstriyel bir robotik manipülatörünün eklem esaslı yörünge kontrolü için doğrusal öngörü, doğrusal olmayan öngörü ve rastlantısal metotlar kullanan kontrol algoritmalarının tasarlandığı bu tezde, farklı kontrol durumları için algoritmaların performansları incelenmiştir. Algoritmaların istenilen referans yörüngeleri takibi izlenmiş, uç nokta konum hataları, adaptif özellikleri ve değişken yük altındaki dengeleme özellikleri gibi istenmeyen durum karşısındaki davranışları karşılaştırılmıştır. Robotik manipülatörün eklem yörünge kontrolü için geliştirilen ve

öngörülü kontrol algoritmalarının iyileştirildiği bu çalışmalar referans yörüngenin önceden belirlendiği diğer kontrol uygulamaları içinde geliştirilebilir.

# KAYNAKLAR


[1] MARTINS F. N., CELESTE W. C., CARELLI R., An adaptive dynamic controller for autonomous mobile robot trajectory tracking, Control Engineering Practice, 16, pp. 1354-1363, 2008.

[2] KAWASAKI M., KAWAMURA S., TSUKAHARA M., MORITA S., KOMIYA M., NATSUGA M., Near-infrared spectroscopic sensing system for on-line milk quality assessment in a milking robot, Computers and Electronics in Agriculture, 63, pp. 22-27, 2008.

[3] LARSSON S., KJELLANDER J. A. P., Path planning for laser scanning with an industrial robot, Robotics and Autonomous Systems, 56, pp. 615-624, 2008.

[4] CARELLI R., FREIRE E. O., Corridor navigation and wall-following stable control for sonar-based mobile robots, Robotics and Autonomous Systems, 45, pp. 235-247, 2003.

[5] DELCOMYN F., NELSON M. E., Architectures for a biomimetic hexapod robot, Robotics and Autonomous Systems, 30, pp. 5-15, 2000.

[6] GOTO S., USUI T., KYURA N., NAKAMURA M., Forcefree control with independent compensation for industrial articulated robot arms, Control Engineering Practice, 15, pp. 627-638, 2007.

[7] FARID M., LUKASIEWICZ S. A., Dynamic modeling of spatial manipulators with flexible links and joints, Computers and Structures, 75, pp. 419-437, 2000.

[8] SOMOLINOS J. A., SANCHEZ V. F., Design, dynamic modelling and experimental validation of a new three-degree-of-freedom flexible arm, Mechatronics, 12, pp. 919-948, 2002.

[9] VIVAS A., POIGNET P., Predictive functional control of a paralel robot, Control Engineering Practice, 13, pp. 863-874, 2005.

[10] WHITNEY, D. E., Resolved Motion Rate Control of Manipulators and Human Prostheses, IEEE Trans. Man-Machine Syst., MMS-10, pp. 47-53, 1969.

[11] GOLDENBERG, A. A., APKARIAN, J. A., SMITH, H. W., An Approach to Adaptive Control of Robot Manipulators Using the Computed Torque Technique, Journal of Dynamic Systems, Measurement, and Control, 111, pp. 1-8, 1989.

[12] KAHN, M. E., ROTH, B., The Near-Minimum Time Control of Open-loop Articulated Kinematics Chains, Journal of Dynamic Systems, Measurement, Control, 93, pp. 164-172, 1971.

[13] SARIDIS, G. S., LEE, C. S. G., An Approximation Theory of Optimal Control for Trainable Manipulators, IEEE Trans. Syst. Man. Cybern., SMC-9, pp. 152-159, 1979.

[14] WANG, S. Y., TSUCHIYA, T., HASHIMOTO, Y., Robot Manipulator Path Control Based On Online Fuzzy Trajectory Planning, Japanese Journal of Fuzzy Theory and Systems, 5, 1993.

[15] BEZINE, H., DERBEL, N., ALIMI, A. M., Fuzzy control of robot manipulators: some issues on design and rule base size reduction, Engineering Applications of Artificial Intelligence, 15, 5, pp. 401-416, 2002.

[16] CHIN, I., MITAL, D. P., Application of Neural Network in Robotic Control, IEEE International Workshop on Emerging Technologies and Factory Automation, USA, 1994.

[17] OZAKI, T., SUZUKI, T., FURUHASHI, T., OKUMA, S., UCHIKAWA, Y., Trajectory control of robotic manipulators using neural networks, IEEE Transaction. Industrial Electronics, 38, 3, pp. 195-202, 1991.

[18] ER, M. J., LOW, C. B., NAH, K. H. Nah, LIM, M. H., NG S. Y., Real-time implementation of a dynamic fuzzy neural networks controller for a SCARA, Microprocessors and Microsystems, 26, 9-10, pp. 449-461, 2002.

[19] DONGBING, G., HUOSHENG, H., Neural Predictive Control for a Car-like Mobile Robot, Robotics and Autonomous Systems, 39, pp. 73-86, 2002.

[20] BAPTISTA, L. F., SOUSA, J. M., COSTA, J. M. G., Fuzzy predictive algorithms applied to real-time force control, Control Engineering Practice, 9, 4, pp. 411-423, 2001.

[21] DUBOWSKY, S., DES FORGES, D. T., The Application of Model Referenced Adaptive Control to Robotic Manipulators, ASME Journal of Dynamic Systems, Measurement, and Control, 101, 3, pp. 193-200, 1979.

[22] DUBOWSKY, S., DES FORGES, D. T., Robotic Manipulator Control Systems with Invariant Dynamic Characteristics, Proc. Fifth World Congress on Theory of Machines and Mechanisms, Montreal, Canada, pp. 1515-1518, 1979.

[23] PETERKA, V., ASTROM, K. J., Control of Multivariable Systems with Unknown but Constant Parameters, Proc. 3rd IFAC Symp. on Identification and Process Parameter Estimation, The Hague, Netherlands, pp. 535-544, 1973.

[24] LAM, K. P., Implicit and Explicit Self-Tuning Controllers, D. Phil. Thesis, Oxford University and OUEL report 1134180.

[25] CLARKE, D. W., KANJILAL, P. P., MOHTADI, C., A Generalized LQG Approach to Self-Tuning Control: Part I- Aspects of Design, International Journal of Control, 41, 6, pp. 1509-1523, 1987.

[26] LEE, C. S. G., CHUNG, M. J., An Adaptive Control Strategy for Computer-Based Manipulators, Proc. 21st IEEE Conference on Decision and Control, Orlando, Florida, USA, pp. 95-100, 1982.

[27] LEE, C. S. G., CHUNG, M. J., An Adaptive Perturbation Control with Feedforward Compensation for Robot Manipulators, Simulation, 44, pp. 127-136, 1985.

[28] CLARKE, D. W., MOHTADI, C., TUFFS, P. C., Generalized Predictive Control - Part 1: The Basic Algorithm, Automatica, 23, pp. 137-148, 1987.

[29] CLARKE, D. W., MOHTADI, C., TUFFS, P. C., Generalized Predictive Control - Part 2: The Basic Algorithm, Automatica, 23, pp. 149-163, 1987.

[30] CLARKE, D. W., MOHTADI, C., TUFFS, P. C., Generalized Predictive Control: A New Robust Self-Tuning Algorithm in Landau, I. D., and L. Dugard (Eds). Commande Adaptive - Aspects Pratiques et Theoriques, Masson, Paris, pp. 209-228, 1987.

[31] CLARKE, D. W., ZHANG, L., Long-Range Predictive Control Using Weighting-Sequence Models, IEE Proceedings Part. D, 134, 3, pp. 187-195, 1987.

[32] CLARKE, D. W., Generalized Predictive Control: A Robust Predictive Control: A Robust Self-Tuning Algorithm, ACC 1987.

[33] KAYNAK, O., Robot Kollarında Öngörülü Denetim, 4. Ulusal Otomatik Kontrol Sempozyumu, İstanbul, sf. 31-42, 1987.

[34] KAYNAK, O., Robot Kollarının Denetiminde Uyarlamalı ve Öngörülü Yöntemler, 5. Ulusal Otomatik Kontrol Sempozyomu, İstanbul, sf. 1-17, 1989.

[35] KAYNAK, O., ALPTEKİN, A. R., Bir artık robotun öngörülü denetimi, Elektrik Mühendisliği 3. Ulusal Kongresi, İstanbul, sf. 666-669, 1989.

[36] KAZAN, R., Üç Eklemli Bir Robot Kolunu Eklem Esaslı Öngörülü Kontrolü, Doktora tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, 1992.

[37] ORTEGA, J. G., CAMACHO, E. F., Neural Network GPC for Mobile Robot Path Tracking, Robotics and Computer-Integrated manufacturing, 11, pp. 271-278, 1994.

[38] SONG, B., KOIVO, A., Neural Network Model Based Control of a Flexible Link Manipulator, Proc. of the IEEE International Conference on Robotics & Automation, Leuven, Belgium, pp. 812-817, May 1998.

[39] ÖZ, C., Üç Eklemli Bir Robot Kolunun Yapay Sinir Ağları ile Eklem Esaslı Yörünge Kontrolü, Doktora tezi, Sakarya Ü. Fen Bilimleri Enstitüsü, 1998.

[40] TEMURTAS, H., Üç Eklemli Bir Robot Kolunun Nöro Genelleştirilmiş Öngörülü Kontrol İle Eklem Esaslı Yörünge Kontrolü, Doktora Tezi, Sakarya Ü. Fen Bilimleri Enstitüsü, 2004.

[41] CLARKE, D. W., Advances in Model-Based Predictive Control, in Advances in Model-Based Predictive Control, ed. by D. W. Clarke, Oxford University Press, 1994.

[42] PICHE, S., RODSARI, B., JOHNSON, D., GERULES, M., Nonlinear model predictive control using neural networks, IEEE Control Systems Magazine, 20, pp. 53-62, 2000.

[43] SORENSEN, P. H., NORGAARD, M., RAVN, O., POULSEN, N. K., Implementation of neural network based non-linear predictive control, Neurocomputing, 28, pp. 37-51, 1999.

[44] HUANG, Y. L., LOU, H. H., GONG, J. P., EDGAR, T. F., Fuzzy model predictive control, IEEE Transactions on Fuzzy Systems, 8, 6, pp. 665-677, 2000.

[45] JUANG, J., PHAN, M., Deadbeat Predictive Controllers, NASA Technical Memorandum 112862, Langley Research Center, Hampton, Virginia, May 1997.

[46] MARTINEZ, M., SENENT, J. S., BLASCO, X., Generalized Predictive Control using Genetic Algorithms, Artificial Intelligence, 11, pp. 355-367, 1998.

[47] RAMIREZ, D., Non-linear MBPC for Mobile Robot Navigation Using Genetic Algorithms, Proceedings of the 1999 IEEE International Conference on Robotics & Automation, Detroit, Michigan, USA, pp. 2452-2457, 1999.

[48] CASSEMIRO E.R., ROSARIO J.M., DUMUR D., Robot Axis Dynamics Control Using A Virtual Robotics Environment, 10 th IEEE International Conference on Emerging Technologies and Factory Automation, 1, pp. 305-311, 2005.

[49] ARMSTRONG B., KHATIP O., and BURDICK J., The explicit dynamic model and inertial parameters of the PUMA 560 arm, IEEE Conference on Robotics and Automation, San Francisco, USA, pp. 510-518, 1986.

[50] KHATIP O., JAMISOLA R., LIM T. M., The Operational Space Formulation Implementation to Aircraft Canopy Polishing Using A Mobil Manipulator, IEEE International Conference on Robotics&Automation, Washington D.C., USA, pp. 400-405, 2002.

[51] GOLDBERG D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, USA, 1989.

[52] FERRAGUD X.B., IRANZO M. A. M., ESPANOL J. S., SANCHIS J., Generalized Predictive Control Using Genetic Algorithms (GAGPC). An Application to Control of a Non-linear Process with Model Uncertainty, Lecture Notes In Computer Science, 1415, pp. 428 – 437, 1998.

[53] FILALI S., WERTZ V., Using genetic algorithms to optimize the design parameters of generalized predictive controllers, International Journal of System Science, 32, 4, pp. 503-512, 2001.

[54] MONTEIRO D.C. , MADRID M.K., Planning of Robot Trajectories with Genetic Algorithms, Proceedings of the RoMoCo'99 Robot Motion and Control, Poland, pp. 223-228, 1999.

[55] HU Y., YONG S. X., A Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot, 2004 IEEE International Conference on Robotic&Automation, New Orleans,USA, pp. 4350-4355, 2004.

[56] SOLOWAY D., Neural Generalized Predictive Control for Real-Time Control, Masters Thesis, Old Dominion University, 1996.

[57] ÖZ, C., KAZAN, R., FERİKOĞLU, A., YUMUŞAK, N., An Inverse Kinematics Solution for Robotic Manipulators with Artificial Neural Networks, 8th International Machine Design and Production Conference, Ankara, Turkey, pp. 402-403, 1998.

[58] ABDELHAMEED M. M., TOLBAH F. F. , A recurrent neural network based sequential controller for manufacturing automated systems, Mechatronics, 12, pp. 617-633, 2002.

[59] ELMAN J. L., Finding Structure in Time, Cognitive Science, 14, pp. 179-211, 1990.

[60] KREMER S. C. , On the computational power of Elman-style recurrent networks, IEEE Transactions Neural Networks, 6, 4, pp. 1000-1004, 1995.

[61] SANAYE-PASAND M., MALIK O. P., Implementation and laboratory test results of an Elman network-based transmission line directional relay, IEEE Transactions on Power Delivery, 14, 4, pp. 782-788, 1999.

[62] HUSKEN M., STAGGE P., Recurrent Neural Networks for Time Series classification, Neurocomputing, 50, pp. 233-235, 2003.

[63] SOLOWAY D., HALEY P. J, Neural Generalized Predictive Control: A Newton - Raphson Implementation, Proceedings of the IEEE CCA/ISIC/CACSD, IEEE Paper No. ISIAC-TA5.2, Sept. 15-18, 1996.

[64] SOLOWAY D., HALEY P. J., Neural Generalized Predictive Control: A Newton-Raphson Implementation, NASA Technical Memorandum 110244, Langley Research Center, Hampton, Virginia, February 1997.

[65] HALEY P. J., SOLOWAY D., GOLD B., Real-Time Adaptive Control Using Neural Generalized Predictive Control, 1999 American Control Conference, San Diego, California, 6, pp. 4278-4282, 1999.

[66] http://www.makinateknik.org/robotik/robot_nedir.php (06.05.2005)

[67] DENAVIT J., HARTENBERG R. S., A Kinematics Notation for Lower-Pair Mechanisms, J. Applied Mechanics, 22, pp. 215-221, 1955.

[68] LEE C. S. G., Robot Arm Kinematics, Dynamics, and Control, Computer, 15, pp. 62-80, 1982.

[69] PAUL R. P. C., Robot Manipulators: Mathematics, Programming and Control, MIT Press, USA, 1981.

[70] HOLLERBACH J. M., A Recursive Lagrange Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity, IEEE Trans. on Systems, Man, and Cybernetics, SMC-10, 11, pp. 730-736, 1980.

[71] ORIN D. E., GHEE M. C., VUKOBRATOVIC R. B., HARTOCH G., Kinematics and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods, Mathematical Biosciences, 43, pp. 107-130, 1979.

[72] LEE C. S. G., LEE B. H., NIGAM R., An Efficient Formulation of Robot Arm Dynamics for Control Analysis and Manipulator Design, Robotics and Integrated Manufacturing, University of Michigan, 1982.

[73] www.ar2.com (22.11.2007)

[74] http://www.rpautomation.com/Robotspecs/560spec.htm (13.07.2004)

[75] RODRIGO J., MARCELO H. A., OETOMO D., KHATIP O., LIM T. M., LIM S. Y., The Operational Space Formulation Implementation to Aircraft Canopy Polishing Using a Mobile Manipulator, International Conference on Robotics&Automation, Whashington D C., pp. 400-405, 2002.

[76] TARN T. J., BEJCZY A. K., MARTH G. T., RAMADORAI A. K., Performance Comparison of Four Manipulator Servo Schemes, IEEE Control Systems, ISDN:0272-1708/93/S03.00, pp. 22-29, 1993.

[77] PLACKETT R. L., Some Theorems in Least Squares, Biometrica, 37, pp. 149-157, 1950.

[78] PETERKA V., A Square Root Filter for Real Time Multivariable Regression, Cybernetica, 11, pp. 53-67, 1975.

[79] BIERMAN, G. J., Factorization Methods for Discrete Sequential Estimation, Academic Press, New York, 1977.

[80] HOLLAND J., Adaptation in Neural and Artificial Systems Ann Arbor, Michigan:University of Michigan Pres, USA, 1975.

[81] ENGİN O.,Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu, İTÜ Fen Bilimleri Enstitüsü, Doktora Tezi, 2001.

[82] JANG J.S.R., Neuro-Fuzzy and Soft Computing: AComputational Approach to Learning and Machine Intelligence, Chapter 7: Derivate-Free Optimization, Prentice-Hall, USA, pp. 173-196, 1997.

[83] YENİAY Ö., An Overwiev of Genetic Algorithms, Anadolu Üniversitesi Bilim ve Teknoloji Dergisi, 2, 1, sf. 37-49, 2001.

[84] YEO M. F., AGYEL E. O., Optimising Engineering Problems Using Genetic Algorithms, Engineering Computations, 15, 2, pp. 268-280, 1996.

[85] FILALI A., ENGİN O., Genetik Algoritmalarla Akış Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu, İTÜ Dergisi, sf. 1-6, 2002.

[86] BRAYSY O., Local Search and Variable Neighborhood Search Algorithms for The Vehicle Routing Problem With Time Windows, PhD Thesis, 2001.

[87] FUNG R.Y.K., TANG J., WANG D., Extension of A Hybrid Genetic Algorithm for Nonlinear Programming Problems with Equality and Inequality Constraints, Computers & Operations Research, 29, 3, pp. 261-274, 2001.

[88] ALTIPARMAK F., DENGİZ B., SMİTH A. E., An Evolutionary Approach for Reliability Optimization in Fixed Topology Computer Network, Transactions On Operational Research, 12, 1-2, pp. 57-75, 2000.

[89] SINREICH D., SAMAKH E., A Genetic Approach to the Pickup/Delivery Station Location Problem in Segmented Flow Based Material Handling Systems, Journal of Manufacturing Systems, 18, 2, pp. 81-99, 1999.

[90] NARENDRA K. S., PARTHASARATHY K., Identification and Control of Dynamical Systems using Neural Networks, IEEE Transactions on Neural Networks, 1, 1, pp. 4-27, 1990.

[91] MONTAGUE G. A., WILLIS M. T. T., MORRIS A. J., Artificial Neural Network Based Control, International Conference on Control, 1, pp. 266-271, 1991.

[92] TAKAHASHI Y., Adaptive Predictive Control of Nonlinear Time-Varying System using Neural Network, IEEE International Conference on Neural Networks, 3, pp. 1464-1468, 1993.

[93] PSICHOGIOS D. C., UNGAR L.H., Nonlinear Internal Model Control and Model Predictive Control using Neural Networks, 5th IEEE International Symposium on Intelligent Control, Philadelphia, PA, USA, pp. 1082-1087, 1990.

[94] JEONG J. S., SUNWON P., Neural Model-Predictive Control for Nonlinear Chemical Processes, Journal of Chemical Engineering of Japan, 26, 4, pp. 347-354, 1993.

# EKLER

## Ek A. Puma 560 Robot Kolu Açık Dinamik Modeli

Tez çalışmasında kullanılan altı eklemli Puma 560 model robot kolunun dinamik modeli için Brian Amstrong ve ekibinin yapmış olduğu çalışmadan faydalanılmıştır [49]. Çalışmada robot koluna ait atalet, coriolis, merkezkaç ve yerçekimi parametreleri belirlenerek tork değerini hesaplayan açık bir dinamik model oluşturulmuştur. Bu bölümde açık dinamik model parametreleri anlatılacaktır.

Robot kolu dinamik modelinde eklemlere ait tork ifadesi aşağıdaki formül ile temsil edilir:

$$D(q(t))\ddot{q} + B(q(t))[\dot{q}\dot{q}] + C(q(t))[\dot{q}^2] + G(q(t)) = \tau(t)$$

Burada;

$D(q(t))$ nxn boyutlu atalet matrisi

$B(q(t))$ nx(n-1)/2 boyutlu Coriolis tork matrisi

$C(q(t))$ nxn boyutlu merkezkaç tork matrisi

$\ddot{q}$ n boyutlu ivme vektörü

$G(q(t))$ n boyut yer çekim tork vektörü

$\tau(t)$ eklem tork vektörüdür.

Eklemlere ait tork vektörü;

$$\tau(t) = [\tau_1(t), \tau_2(t), \tau_3(t), \tau_4(t), \tau_5(t), \tau_6(t)]^T$$

İvme vektörü;

$$\ddot{q}(t) = [\ddot{q}_1(t), \ddot{q}_2(t), \ddot{q}_3(t), \ddot{q}_4(t), \ddot{q}_5(t), \ddot{q}_6(t)]^T$$

Açısal hız kareleri vektörü;

$$\dot{q}^2(t) = [\dot{q}_1(t), \dot{q}_2(t), \dot{q}_3(t), \dot{q}_4(t), \dot{q}_5(t), \dot{q}_6(t)]^T$$

Açısal hız kombinasyon vektörü;

$$\dot{q}\dot{q}(t) = \begin{bmatrix} \dot{q}_1\dot{q}_2, \dot{q}_1\dot{q}_3, \dot{q}_1\dot{q}_4, \dot{q}_1\dot{q}_5, \dot{q}_1\dot{q}_6, \dot{q}_2\dot{q}_3, \dot{q}_2\dot{q}_4, \dot{q}_2\dot{q}_5, \dot{q}_2\dot{q}_6, \dot{q}_3\dot{q}_4, \dot{q}_3\dot{q}_5, \dot{q}_3\dot{q}_6, \dot{q}_4\dot{q}_5 \\ , \dot{q}_4\dot{q}_6, \dot{q}_5\dot{q}_6 \end{bmatrix}^T$$

**Ek A.1. Atalet matrisi**

Altı eklemli robot kolu için $D(q(t))$ matrisi 6x6 boyutunda bir matristir.

$$D = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix}_{6x6}$$

Matris elemanları, robot kolunun atalet değerlerini temsil eder. Bu hesaplamalar atalet sabitleri, motor ataletleri ve açısal bilgiler kullanılarak yapılmıştır. Hesaplamalarda kolaylık olması için bazı kısaltmalar yapılmıştır. Örneğin $\sin(q_2)$ yerine S2, $\cos(q_{2+}q_3)$ yerine C23, $\cos(q_2)$* $\cos(q_2)$ için CC2, $\cos(q_4)$*$\sin(q_4)$ için CS4 kullanılmıştır. Atalet matrisi elemanları aşağıdaki gibi tanımlanır:

$$a_{11} = I_{m1} + I_1 + I_3 * CC2 + I_7 * SS23 + I_{10} * SC23 + I_{11} * SC2 + I_{20} * (SS5 * (SS23 * (1 + CC4) - 1) - 2 * SC23 * C4 * SC5) + I_{21} * SS23 * CC4 + 2 * \{I_5 * C2 * S23 + I_{12} * C2 * C23 + I_{15} * (SS23 * C5 + SC23 * C4 * S5) + I_{16} * C2 * (S23 * C5 + C23 * C4 * S5) + I_{18} * S4 * S5 + I_{22} * (SC23 * C5 + CC23 * C4 * S5)\}; \approx 2.57 + 1.38 * CC2 + 0.30 * SS23 + 7.44 \times 10^{-1} * C2 * S23$$

$$a_{12} = I_4 * S2 + I_8 * C23 + I_9 * C2 + I_{13} * S23 - I_{15} * C23 * S4 * S5 + I_{16} * S2 * S4 * S5 + I_{18} * (S23 * C4 * S5 - C23 * C5) + I_{19} * S23 * SC4 + I_{20} * S4 * (S23 * C4 * CC5 + C23 * SC5) + I_{22} * S23 * S4 * S5; \approx 6.90 \times 10^{-1} * S2 - 1.34 \times 10^{-1} * C23 + 2.38 \times 10^{-2} * C2$$

$$a_{13} = I_8 * C23 + I_{13} * S23 - I_{15} * C23 * S4 * S5 + I_{19} * S23 * SC4 + I_{18} * (S23 * C4 * S5 - C23 * C5) + I_{22} * S23 * S4 * S5 + I_{20} * S4 * (S23 * C4 * CC5 + C23 * SC5); \approx -1.34 \times 10^{-1} * C23 + -3.97 \times 10^{-3} * S23$$

$$a_{14} = I_{14} * C23 + I_{15} * S23 * C4 * S5 + I_{16} * C2 * C4 * S5 + I_{18} * C23 * S4 * S5 - I_{20} * (S23 * C4 * SC5 + C23 * SS5) + I_{22} * C23 * C4 * S5; \approx 0$$

$$a_{15} = I_{15} * S23 * S4 * C5 + I_{16} * C2 * S4 * C5 + I_{17} * S23 * S4 + I_{18} * (S23 * S5 - C23 * C4 * C5) + I_{22} * C23 * S4 * C5; \approx 0$$

$$a_{16} = I_{23} * (C23 * C5 - S23 * C4 * S5); \approx 0$$

$$a_{22} = I_{m2} + I_2 + I_6 + I_{20} * SS4 * SS5 + I_{21} * SS4 + 2 * \{I_5 * S3 + I_{12} * C3 + I_{15} * C5$$
$$+ I_{16} * (S3 * C5 + C3 * C4 * S5) + I_{22} * C4 * S5\}; \approx 6.79 + 7.44 \times 10^{-1} * S3$$
$$a_{23} = I_5 * S3 + I_6 + I_{12} * C3 + I_{16} * (S3 * C5 + C3 * C4 * S5) + I_{20} * SS4 * SS5$$
$$+ I_{21} * SS4 + 2 * \{I_{15} * C5 + I_{22} * C4 * S5; \approx .333 + 3.72 \times 10^{-1} * S3 - 1.10 \times 10^{-2} * C3$$

$$a_{24} = -I_{15} * S4 * S5 - I_{16} * S3 * S4 * S5 + I_{20} * S4 * SC5 \approx 0$$

$$a_{25} = I_{15} * C4 * C5 + I_{16} * (C3 * S5 + S3 * C4 * C5) + I_{17} * C4 + I_{22} * S5; \approx 0$$

$$a_{26} = I_{23} * S4 * S5; \approx 0$$

$$a_{33} = I_{m3} + I_6 + I_{20} * SS4 * SS5 + I_{21} * SS4 + 2 * \{I_{15} * C5 + I_{22} * C4 * S5); \approx 1.16$$

$$a_{34} = -I_{15} * S4 * S5 + I_{20} * S4 * SC5; \approx -1.25 \times 10^{-3} * S4 * S5$$

$$a_{35} = I_{15} * C4 * C5 + I_{17} * C4 + I_{22} * S5; \approx 1.25 \times 10^{-3} * C4 * C5$$

$$a_{36} = I_{23} * S4 * S5; \approx 0$$

$$a_{44} = I_{m4} + I_{14} - I_{20} * SS5; \approx 0.20$$

$$a_{45} = 0$$

$$a_{46} = I_{23} * C5; \approx 0$$

$$a_{55} = I_{m5} + I_{17}; \approx 0.18$$

$$a_{56} = 0$$

$$a_{66} = I_{m6} + I_{23} \approx 0.19$$

Hesaplanan bu değerler kg-m$^2$ birimine sahiptir.

Yukarıda parametre değerlerinde geçen $I_{mi}$ ifadeleri motor ataletlerini, $I_{nn}$ ise atalet sabitlerini temsil eder ve değerleri aşağıdaki tablolarda verilmiştir.

Tablo A.1. Puma 560 eklem motor ataletleri (kg-m$^2$)

| 1.motor ($I_{m1}$) | 2.motor ($I_{m2}$) | 3.motor ($I_{m3}$) | 4.motor ($I_{m4}$) | 5.motor ($I_{m5}$) | 6.motor ($I_{m6}$) |
|---|---|---|---|---|---|
| 1.14 | 4.71 | 0.83 | 0.2 | 0.179 | 0.193 |

Atalet sabitleri:

$$I_1 = I_{zz1} + m_1 * r_{y1}^{\ 2} + m_2 * d_2^{\ 2} + (m_4 + m_5 + m_6) * a_3^{\ 2} + m_2 * r_{z2}^{\ 2} + (m_3 + m_4 + m_5 + m_6) * (d_2 + d_3)^2 + I_{zz2} + I_{yy3} + 2 * m_2 * d_2 * r_{z2} + m_2 * r_{y2}^{\ 2} + m_3 * r_{z3}^{\ 2} + 2 * m_3 * (d_2 + d_3) * r_{z3} + I_{zz4} + I_{yy5} + I_{zz6};$$

$$I_2 = I_{zz2} + m_2 * (r_{x2}^{\ 2} + r_{y2}^{\ 2}) + (m_3 + m_4 + m_5 + m_6) * a_2^{\ 2};$$

$$I_3 = -I_{xx2} + I_{yy2} + (m_3 + m_4 + m_5 + m_6) * a_2^{\ 2} * m_2 * r_{x2}^{\ 2} - m_2 * r_{y2}^{\ 2};$$

$$I_4 = m_2 * r_{x2} * (d_2 + r_{z2}) + m_3 * a_2 * r_{z3} + (m_3 + m_4 + m_5 + m_6) * a_2 * (d_2 + d_3);$$

$$I_5 = -m_3 * a_2 * r_{y3} + (m_4 + m_5 + m_6) * a_2 * d_4 + m_4 * a_2 * r_{z4};$$

$$I_6 = I_{zz3} + m_3 * r_{y3}^{\ 2} + m_4 * a_3^{\ 2} + m_4 * (d_4 + r_{z4})^2 + I_{yy4} + m_5 * a_3^{\ 2} + m_5 * d_4^{\ 2} + I_{zz5} + m_6 * a_3^{\ 2} + m_6 * d_4^{\ 2} + m_6 * r_{z6}^{\ 2} + I_{xx6};$$

$$I_7 = m_3 * r_{y3}^{\ 2} + I_{xx3} - I_{yy3} + m_4 * r_{z4}^{\ 2} + 2 * m_4 * d_4 * r_{z4} + (m_4 + m_5 + m_6) * (d_4^{\ 2} - a_3^{\ 2}) + I_{yy4} - I_{zz4} + I_{zz5} - I_{yy5} + m_6 * r_{z6}^{\ 2} - I_{zz6} + I_{xx6};$$

$$I_8 = -m_4 * (d_2 + d_3) * (d_4 + r_{z4}) - (m_5 + m_6) * (d_2 + d_3) * d_4 + m_3 * r_{y3} * r_{z3} + m_3 * (d_2 + d_3) * r_{y3};$$

$$I_9 = m_2 * r_{y2} * (d_2 + r_{z2});$$

$$I_{10} = 2 * m_4 * a_3 * r_{z4} + 2 * (m_4 + m_5 + m_6) * a_3 * d_4;$$

$$I_{11} = -2 * m_2 * r_{x2} * r_{y2};$$

$$I_{12} = (m_4 + m_5 + m_6) * a_2 * a_3;$$

$$I_{13} = (m_4 + m_5 + m_6) * a_3 * (d_2 + d_3);$$

$$I_{14} = I_{zz4} + I_{yy5} + I_{zz6};$$

$$I_{15} = m_6 * d_4 * r_{z6};$$

$$I_{16} = m_6 * a_2 * r_{z6};$$

$$I_{17} = I_{zz5} + I_{xx6} + m_6 * r_{z6}^{\ 2};$$

$$I_{18} = m_6 * (d_2 + d_3) * r_{z6};$$

$$I_{19} = I_{yy4} - I_{xx4} + I_{zz5} - I_{yy5} + m_6 * r_{z6}^{\ 2} + I_{xx6} - I_{zz6};$$

$$I_{20} = I_{yy5} - I_{xx5} - m_6 * r_{z6}^{\ 2} + I_{zz6} - I_{xx6};$$

$$I_{21} = I_{xx4} - I_{yy4} + I_{xx5} - I_{zz5};$$

$$I_{22} = m_6 * a_3 * r_{z6};$$

$$I_{23} = I_{zz6};$$

Atalet sabitleri formüllerinde geçen $I_{xxi}, I_{yyi}, I_{zzi}$ ifadeleri uzuvlara ait atalet dyadic sabitlerini, $r_{xi}, r_{yi}, r_{zi}$ ifadeleri uzuvların ağırlık merkezlerini, $m_i$ ise uzuvların kütlelerini temsil eder. Bu parametre değerleri aşağıdaki tablolarda verilmiştir.

Tablo A.2. Uzuvlara ait dyadic sabitleri (kg-m$^2$)

| Uzuvlar | $I_{xx}$ | $I_{yy}$ | $I_{zz}$ |
|---|---|---|---|
| 1.Uzuv | - | - | 0.35 |
| 2.Uzuv | 0.130 | 0.524 | 0.539 |
| 3.Uzuv | 0.066 | 0.0125 | 0.086 |
| 4.Uzuv | $1.8*10^{-3}$ | $1.8*10^{-3}$ | $1.3*10^{-3}$ |
| 5.Uzuv | $0.3*10^{-3}$ | $0.3*10^{-3}$ | $0.4*10^{-3}$ |
| 6.Uzuv | $0.15*10^{-3}$ | $0.15*10^{-3}$ | $0.04*10^{-3}$ |

Tablo A.3. Uzuvların ağırlık merkezleri (metre)

| Uzuvlar | $r_x$ | $r_y$ | $r_z$ |
|---|---|---|---|
| 2.Uzuv | 0.068 | 0.006 | -0.016 |
| 3.Uzuv | 0 | -0.07 | 0.014 |
| 4.Uzuv | 0 | 0 | -0.019 |
| 5.Uzuv | 0 | 0 | 0 |
| 6.Uzuv | 0 | 0 | 0.032 |
| Bilek | 0 | 0 | -0.064 |

Tablo A.4. Uzuvların ağırlığı (kg)

| Uzuvlar | Ağırlığı |
|---|---|
| 2.Uzuv | 17.4 |
| 3.Uzuv | 4.8 |
| 4.Uzuv | 0.82 |
| 5.Uzuv | 0.34 |
| 6.Uzuv | 0.09 |

Tablolarda verilen değerler kullanılarak atalet sabitleri hesaplanmıştır:

$I_1 = 1.43 \quad \pm 0.05$
$I_2 = 1.75 \quad \pm 0.07$
$I_3 = 1.38 \quad \pm 0.05$
$I_4 = 6.90 \times 10^{-1} \pm 0.20 \times 10^{-1}$
$I_5 = 3.72 \times 10^{-1} \pm 0.31 \times 10^{-1}$
$I_6 = 3.33 \times 10^{-1} \pm 0.16 \times 10^{-1}$
$I_7 = 2.98 \times 10^{-1} \pm 0.29 \times 10^{-1}$
$I_8 = -1.34 \times 10^{-1} \pm 0.14 \times 10^{-1}$
$I_9 = 2.38 \times 10^{-2} \pm 1.20 \times 10^{-2}$
$I_{10} = -2.13 \times 10^{-2} \pm 0.22 \times 10^{-2}$
$I_{11} = -1.42 \times 10^{-2} \pm 0.70 \times 10^{-2}$
$I_{12} = -1.10 \times 10^{-2} \pm 0.11 \times 10^{-2}$
$I_{13} = -3.79 \times 10^{-3} \pm 0.90 \times 10^{-3}$
$I_{14} = 1.64 \times 10^{-3} \pm 0.07 \times 10^{-3}$
$I_{15} = 1.25 \times 10^{-3} \pm 0.30 \times 10^{-3}$
$I_{16} = 1.24 \times 10^{-3} \pm 0.30 \times 10^{-3}$
$I_{17} = 6.42 \times 10^{-4} \pm 3.00 \times 10^{-4}$
$I_{18} = 4.31 \times 10^{-4} \pm 1.30 \times 10^{-4}$
$I_{19} = 3.00 \times 10^{-4} \pm 14.0 \times 10^{-4}$
$I_{20} = -2.02 \times 10^{-4} \pm 8.00 \times 10^{-4}$
$I_{21} = -1.00 \times 10^{-4} \pm 6.00 \times 10^{-4}$
$I_{22} = -5.80 \times 10^{-5} \pm 1.50 \times 10^{-5}$
$I_{23} = 4.00 \times 10^{-5} \pm 2.00 \times 10^{-5}$

Hesaplanan bu sabitler, robot kolu atalet matrisi $D(q(t))$'nin elemanlarının hesaplanmasında kullanılmıştır.

## Ek A.2. Coriolis tork matrisi

Robot kolu genel tork ifadesinde geçen $B(q(t))$ matrisi coriolis tork matrisi olarak adlandırılmaktadır. Bu matris 6x15 boyutunda bir matris olup matris elemanları aşağıdaki gibi hesaplanır:

$$B=\begin{bmatrix} b_{112} & b_{113} & b_{114} & b_{115} & b_{116} & b_{117} & b_{118} & b_{119} & b_{120} & b_{121} & b_{122} & b_{123} & b_{124} & b_{125} & b_{126} \\ b_{212} & b_{213} & b_{214} & b_{215} & b_{216} & b_{217} & b_{218} & b_{219} & b_{220} & b_{221} & b_{222} & b_{223} & b_{224} & b_{225} & b_{226} \\ b_{312} & b_{313} & b_{314} & b_{315} & b_{316} & b_{317} & b_{318} & b_{319} & b_{320} & b_{321} & b_{322} & b_{323} & b_{324} & b_{325} & b_{326} \\ b_{412} & b_{413} & b_{414} & b_{415} & b_{416} & b_{417} & b_{418} & b_{419} & b_{420} & b_{421} & b_{422} & b_{423} & b_{424} & b_{425} & b_{426} \\ b_{512} & b_{513} & b_{514} & b_{515} & b_{516} & b_{517} & b_{518} & b_{519} & b_{520} & b_{521} & b_{522} & b_{523} & b_{524} & b_{525} & b_{526} \\ b_{612} & b_{613} & b_{614} & b_{615} & b_{616} & b_{617} & b_{618} & b_{619} & b_{620} & b_{621} & b_{622} & b_{623} & b_{624} & b_{625} & b_{626} \end{bmatrix}_{6x15}$$

$$b_{112}=2*\{-I_3*SC2+I_5*C223+I_7*SC23-I_{12}*S223+I_{15}*(2*SC23*C5 + (1-2*SS23)*C4*S5)+I_{16}*(C223*C5-S223*C4*S5)+I_{21}*SC23*CC4 + I_{20}*((1+CC4)*SC23*SS5-(1-2*SS23)*C4*SC5)+I_{22}*((1-2*SS23)*C5 - 2*SC23*C4*S5)\}+I_{10}*(1-2*SS23)+I_{11}*(1-2*SS2);\approx -2.76*SC2 + 7.44\times10^{-1}*C223+0.60*SC23-2.13\times10^{-2}*(1-2*SS23)$$

$$b_{113}=2*\{I_5*C2*C23+I_7*S23-I_{12}*C2*S23+I_{15}*(2*SC23*C5 + (1-2*SS23)*C4*S5)+I_{16}*C2*(C23*C5-S23*C4*S5)+I_{21}*SC23*CC4 + I_{20}*((1+CC4)*SC23*SS5-(1-2*SS23)*C4*SC5)+I_{22}*((1-2*SS23)*C5 - 2*SC23*C4*S5)\}+I_{10}*(1-2*SS23);\approx 7.44\times10^{-1}*C2*C23+0.60*SC23 + 2.20\times10^{-2}*C2*S23-2.13\times10^{-2}*(1-2*SS23)$$

$$b_{114}=2*\{-I_{15}*SC23*S4*S5-I_{16}*C2*C23*S4*S5+I_{18}*C4*S5-I_{20}* (SS23*SS5*SC4-SC23*S4*SC5)-I_{22}*CC23*S4*S5-I_{21}*SS23*SC4\}; \approx -2.50\times10^{-3}*SC23*S4*S5+8.60\times10^{-4}*C4*S5-2.48\times10^{-3}*C2*C23*S4*S5$$

$$b_{115}=2*\{I_{20}*(SC5*CC4*(1-CC23)-CC23)-SC23*C4*(1-2*SS5)) - I_{15}*(SS23*S5-SC23*C4*C5)-I_{16}*C2*(S23*S5-C23*C4*C5) + I_{18}*S4*C5+I_{22}*(CC23*C4*C5-SC23*S5)\}((1-2*SS23)*C5 - 2*SC23*C4*S5)\};\approx -2.50\times10^{-3}*(SS23*S5-SC23*C4*C5) - 2.48\times10^{-3}*C2*(S23*S5-C23*C4*C5)+8.60\times10^{-4}*S4*C5$$

$$b_{116}=0$$

$$b_{123} = 2*\{-I_8*S23 + I_{13}*C23 + I_{15}*S23*S4*S5 + I_{18}*(C23*C4*S5 + S23*C5) + I_{19}*C23*SC4 + I_{20}*S4*(C23*C4*CC5 - S23*SC5) + I_{22}*C23*S4*S5\}$$
$$;\approx 2.67\times10^{-1}*S23 - 7.58\times10^{-3}*C23$$

$$b_{124} = -I_{18}*2*S23*S4*S5 + I_{19}*S23*(1-(2*SS4)) + I_{20}*S23*(1-2*SS4*CC5) - I_{14}*S23;\approx 0$$

$$b_{125} = I_{17}*C23*S4 + I_{18}*2*(S23*C4*C5 + C23*S5) + I_{20}*S4*(C23*(1-2*SS5) - S23*C4*2*SC5);\approx 0$$

$$b_{126} = -I_{23}*(S23*C5 + C23*C4*S5\};\approx 0$$

$$b_{134} = b_{124}$$

$$b_{135} = b_{125}$$

$$b_{136} = b_{126}$$

$$b_{145} = 2*\{I_{15}*S23*C4*C5 + I_{16}*C2*C4*C5 + I_{18}*C23*S4*C5 + I_{22}*C23*C4*C5\} + I_{17}*S23*C4 - I_{2O}*(S23*C4*(1-2*SS5) + 2*C23*SC5);\approx 0$$

$$b_{146} = I_{23}*S23*S4*S5;\approx 0$$

$$b_{156} = -I_{23}*(C23*S5 + S23*C4*C5);\approx 0$$

$$b_{212} = 0$$

$$b_{213} = 0$$

$$b_{214} = I_{14}*S23 + I_{19}*S23*(1-(2*SS4)) + 2*\{-I_{15}*C23*C4*S5 + I_{16}*S2*C4*S5 + I_{20}*(S23*(CC5*CC4 - 0.5) + C23*C4*SC5) + I_{22}*S23*C4*S5\}$$
$$;\approx 1.64\times10^{-3}*S23 - 2.50\times10^{-3}*C23*C4S5 + 2.48\times10^{-3}*S2*C4*S5 + 0.30\times10^{-3}*S23*(1-(2*SS4))$$

$$b_{215} = 2*\{-I_{15}*C23*S4*C5 + I_{22}*S23*S4*C5 + I_{16}*S2*S4*C5\} - I_{17}*C23*S4 + I_{20}*(C23*S4*(1-2*SS5) - 2*S23*SC4*SC5)$$
$$;\approx -2.50\times10^{-3}*C23*S4*C5 + 2.48\times10^{-3}*S2*S4*C5 - 6.42\times10^{-4}*C23*S4$$

$$b_{216} = -b_{126}$$

$$b_{223} = 2*\{-I_{12}*S3 + I_5*C3 + I_{16}*(C3*C5 - S3*C4*S5)\};\approx 2.20\times10^{-2}*S3 + 7.44\times10^{-1}*C3$$

$$b_{224} = 2 * \{-I_{16} * C3 * S4 * S5 + I_{20} * SC4 * SS5 + I_{21} * SC4 - I_{22} * S4 * S5\}$$
$$\approx -2.48 \times 10^{-3} * C3 * S4 * S5$$

$$b_{225} = 2 * \{-I_{15} * S5 + I_{16} * (C3 * C4 * C5 - S3 * S5) + I_{20} * SS4 * SC5 + I_{22} * C4 * C5\}$$
$$\approx -2.50 \times 10^{-3} * S5 + 2.48 \times 10^{-3} * (C3 * C4 * C5 - S3 * S5)$$

$$b_{226} = 0$$

$$b_{234} = b_{224}$$

$$b_{235} = b_{225}$$

$$b_{236} = 0$$

$$b_{245} = 2 * \{-I_{15} * S4 * C5 - I_{16} * S3 * S4 * C5\} - I_{17} * S4 + I_{20} * S4 * (1 - 2 * SS5) \approx 0$$

$$b_{246} = I_{23} * C4 * S5 \approx 0$$

$$b_{256} = I_{23} * S4 * C5; \approx 0$$

$$b_{312} = 0$$

$$b_{313} = 0$$

$$b_{314} = 2 * \{-I_{15} * C23 * C4 * S5 + I_{22} * S23 * C4 * S5 + I_{20} * \{S23 * (CC5 * CC4 - 0.5)$$
$$+ C23 * C4 * SC5)\} + I_{14} * S23 + I_{19} * S23 * (1 - (2 * SS4)); \approx -2.50 \times 10^{-3} * C23 * C4$$
$$* S5 + 1.64 \times 10^{-3} * S23 + 0.30 \times 10^{-3} * S23 * (1 - 2 * SS4).$$

$$b_{315} = 2 * (-I_{15} * C23 * S4 * C5 + I_{22} * S23 * S4 * C5) - I_{17} * C23 * S4 + I_{20} * S4$$
$$* (C23 * (1 - 2 * SS5) - 2 * S23 * C4 * SC5); \approx -2.5 \times 10^{-3} * C23 * S4 * C5$$
$$- 6.42 \times 10^{-4} * C23 * S4$$

$$b_{316} = -b_{136}$$

$$b_{323} = 0$$

$$b_{324} = 2 * \{I_{20} * SC4 * SS5 + I_{21} * SC4 - I_{22} * S4 * S5\}; \approx 0$$

$$b_{325} = 2 * \{-I_{15} * S5 + I_{20} * SS4 * SC5 + I_{22} * C4 * C5\}; \approx -2.50 \times 10^{-3} * S5$$

$$b_{326} = 0$$

$$b_{334} = b_{324}$$

$$b_{335} = b_{325}$$

$$b_{336} \neq 0$$

$$b_{345} = -I_{15} * 2 * S4 * C5 - I_{17} * S4 + I_{20} * S4 * (1 - 2 * SS5); \approx -2.50 \times 10^{-3} * S4 * C5$$

$$b_{346} = b_{246}$$

$$b_{356} = b_{256}$$

$$b_{412} = -b_{214}$$

$$b_{413} = -b_{314}$$

$$b_{414} = 0$$

$$b_{415} = -I_{20} * (S23 * C4 * (1 - 2 * SS5) + 2 * C23 * SC5) - I_{17} * S23 * C4$$
$$; \approx -6.42 \times 10^{-4} * S23 * C4$$

$$b_{416} = -b_{146}$$

$$b_{423} = -b_{324}$$

$$b_{424} = 0$$

$$b_{425} = I_{17} * S4 + I_{20} * S4 * (1 - 2 * SS5); \approx -6.42 \times 10^{-4} * S4$$

$$b_{426} = -b_{246}$$

$$b_{434} = 0$$

$$b_{435} = b_{425}$$

$$b_{436} = -b_{346}$$

$$b_{445} = -I_{20} * 2 * SC5; \approx 0$$

$$b_{446} = 0$$

$$b_{456} = -I_{23} * S5; \approx 0$$

$$b_{512} = -b_{215}$$
$$b_{513} = -b_{315}$$

$$b_{514} = -b_{415}$$

$$b_{515} = 0$$

$$b_{516} = -b_{156}$$

$$b_{523} = -b_{325}$$

$$b_{524} = -b_{425}$$

$$b_{525} = 0$$

$$b_{526} = -b_{256}$$

$$b_{534} = b_{524}$$

$$b_{535} = 0$$

$$b_{536} = -b_{356}$$

$$b_{545} = 0$$

$$b_{546} = -b_{456}$$

$$b_{556} = 0$$

$$b_{612} = b_{126}$$

$$b_{613} = b_{136}$$

$$b_{614} = b_{146}$$

$$b_{615} = b_{156}$$

$$b_{616} = 0$$

$$b_{623} = 0$$

$$b_{624} = b_{246}$$

$b_{625} = b_{256}$

$b_{626} = 0$

$b_{634} = b_{624}$

$b_{635} = b_{625}$

$b_{636} = 0$

$b_{645} = b_{456}$

$b_{646} = 0$

$b_{656} = 0$

Hesaplanan bu değerler kg-m$^2$ birimine sahiptir.

**Ek A.3. Merkezkaç kuvvet matrisi**

$C(q(t))$ matrisi merkezkaç kuvvetlerini temsil eden 6x6 boyutunda bir matristir.

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ c_{31} & c_{32} & c_{33} & c_{34} & c_{35} & c_{36} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} & c_{46} \\ c_{51} & c_{52} & c_{53} & c_{54} & c_{55} & c_{56} \\ c_{61} & c_{62} & c_{63} & c_{64} & c_{65} & c_{66} \end{bmatrix}_{6x6}$$

$c_{11} = 0$

$$c_{12} = +I_4 * C2 - I_8 * S23 - I_9 * S2 + I_{13} * C23 + I_{15} * S23 * S4 * S5 + I_{16} * C2 * S4 * S5 + I_{18} * (C23 * C4 * S5 + S23 * C5) + I_{19} * C23 * SC4 + I_{20} * S4 * (C23 * C4 * cC5 - S23 * SC5) + I_{22} * C23 * S4 * S5 \approx 6.90 \times 10^{-1} * C2 + 1.34 * 10^{-1} * S23 - 2.38 \times 10^{-2} * S2$$

$c_{13} = 0.5 * b_{123}$

$$c_{14} = -I_{15} * S23 * S4 * S5 - I_{16} * C2 * S4 * S5 + I_{18} * C23 * C4 * S5 + I_{20} * S23 * S4 * SC5 - I_{22} * C23 * S4 * S5 \approx 0$$

$$c_{15} = -I_{15} * S23 * S4 * S5 - I_{16} * C2 * S4 * S5 + I_{18} * (S23 * C5 + C23 * C4 * S5) - I_{22} * C23 * S4 * S5 \approx 0$$

$$c_{16} = 0$$

$$c_{21} = -0.5 * b_{112}$$

$$c_{22} = 0$$

$$c_{23} = 0.5 * b_{223}$$

$$c_{24} = -I_{15} * C4 * S5 - I_{16} * S3 * C4 * S5 + I_{20} * C4 * SC5 \approx 0$$

$$c_{25} = -I_{15} * C4 * S5 - I_{16} * (C3 * C5 - S3 * C4 * S5) + I_{22} * C5 \approx 0$$

$$c_{26} = 0$$

$$c_{31} = -0.5 * b_{113}$$

$$c_{32} = -c_{23}$$

$$c_{33} = 0$$

$$c_{34} = -I_{15} * C4 * S5 - I_{20} * C4 * SC5 \approx -1.25 \times 10^{-3} * C4 * S5$$

$$c_{35} = -I_{15} * C4 * S5 + I_{22} * C5 \approx c_{34}$$

$$c_{36} = 0$$

$$c_{41} = -0.5 * b114$$

$$c_{42} = -0.5 * b_{224}$$

$$c_{43} = 0.5 * b_{423}$$

$$c_{44} = 0$$

$$c_{45} = 0$$

$$c_{46} = 0$$

$$c_{51} = -0.5 * b_{115}$$

$$c_{52} = -0.5 * b_{225}$$

$$c_{53} = 0.5 * b_{525}$$

$c_{54} = -0.5 * b_{445}$

$c_{55} = 0$

$c_{56} = 0$

$c_{61} = 0$

$c_{62} = 0$

$c_{63} = 0$

$c_{64} = 0$

$c_{65} = 0$

$c_{66} = 0$

Hesaplanan bu değerler kg-m$^2$ birimine sahiptir.

**Ek A.4. Yerçekim tork vektörü**

Yer çekim tork vektörü $G(q(t))$, uzuvların yerçekiminden oluşan çekim kuvvetine karşı enerjilerini temsil eden 6 boyutlu bir vektördür.

$G(q(t)) = [g_1, g_2, g_3, g_4, g_5, g_6]^T$

$g_1 = 0$

$g_2 = g1 * C2 + g2 * S23 + g3 * S2 + g4 * C23 + g5 * (S23 * C5 + C23 * C4 * S5) \approx -37.2 * C2 - 8.4 * S23 + 1.02 * S2)$

$g_3 = g2 * S23 + g4 * C23 + g5 * (S23 * C5 + C23 * C4 * S5) \approx -8.4 * S23 + 0.25 * C23$

$g_4 = -g5 * S23 * S4 * S5 \approx 2.8 \times 10^{-2} * S23 * S4 * S5$

$g_5 = g5 * (C23 * S5 + S23 * C4 * C5) \approx -2.8 \times 10^{-2} * (C23 * S5 + S23 * C4 * C5)$

$g_6 = 0$

Hesaplanan bu değerler kg-m$^2$ birimine sahiptir. Vektör elemanlarında yer alan $g_i$ ifadeleri, eklemlere ait yerçekimi sabitleridir ve aşağıdaki değerler sahiptirler:

$g_1 = -37.2 \pm 00.5$

$g_2 = -8.44 \pm 0.20$

$g_3 = 1.02 \pm 0.50$

$g_4 = 2.49 \times 10^{-1} \pm 0.25 \times 10^{-1}$

$g_5 = -2.82 \times 10^{-2} \pm 0.56 \times 10^{-2}$

**Ek A.5. Homojen dönüşüm matrisi**

Bir robot kolun kinematik ve dinamik denklemlerini sistematik ve genelleştirilmiş bir şekilde çıkarabilmek için, her bir $(i)$ uzvuna cisim koordinat çerçevesi ( $\vec{x}_i, \vec{y}_i, \vec{z}_i$ ) yerleştirilir. Komşu koordinat çerçeveleri arasında Denavit ve Hartenberg [47] tarafından geliştirilen dört parametre yardımı ile ilişki kurulmaktadır (Şekil 2.3 ve tablo 1). Bu parametreler aşağıdaki şekilde tanımlanmaktadır.

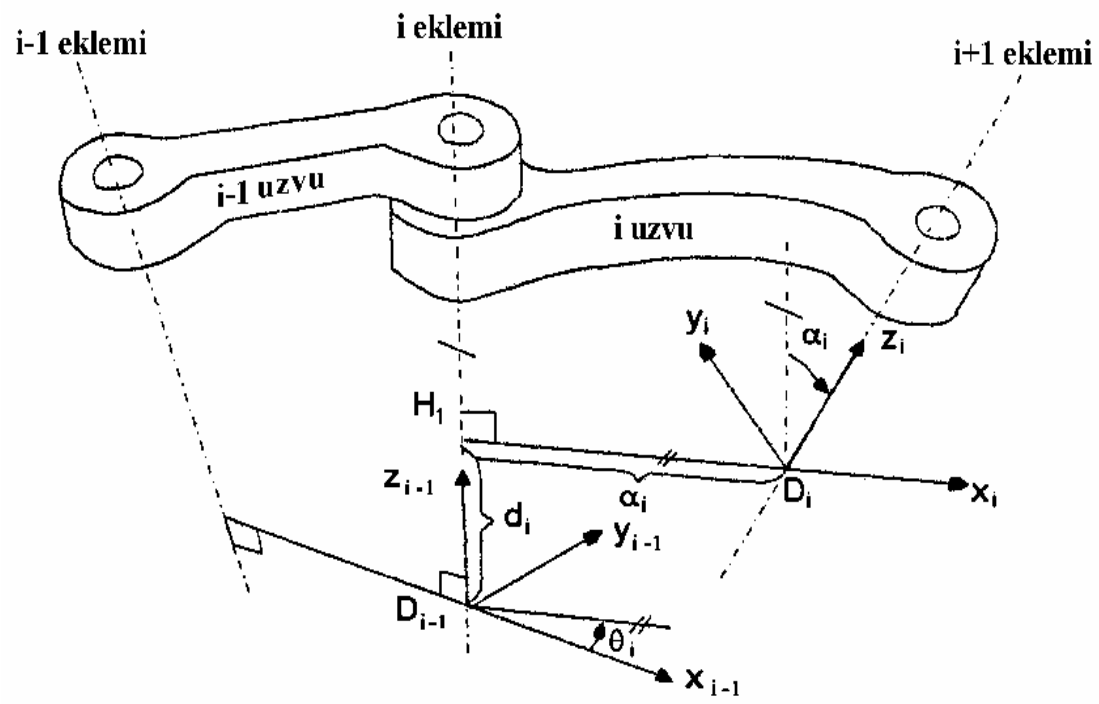
$\theta_i$ : $\vec{z}_{i-1}$ ekseni etrafında $\vec{x}_{i-1}$ ekseninden $\vec{x}_i$ eksenine kadar olan eklem açısı(sağ el kuralı kullanılarak),

$d_i$ : $\vec{z}_{i-1}$ ekseni boyunca $(i-1)$.koordinat çerçevesinin orijininden $\vec{x}_i$ ekseni ile $\vec{z}_{i-1}$ ekseninin kesişme yerine kadar olan mesafe,

$a_i$ : $\vec{x}_i$ ekseni boyunca $\vec{x}_i$ ekseni ile $\vec{z}_{i-1}$ ekseninin kesişme noktasından $(i)$.koordinat çerçevesinin orijinine kadar olan mesafe ($\vec{z}_{i-1}$ ve $\vec{z}_i$ eksenleri arasındaki en kısa mesafe),

$\alpha_i$ : $\vec{x}_i$ ekseni etrafında $\vec{z}_{i-1}$ ekseninden $\vec{z}_i$ eksenine kadar olan açı (sağ el kuralı kullanılarak)

Dönel eklem için $d_i$, $a_i$, $\alpha_i$ parametreleri sabit, $\theta_i$ parametresi değişkendir ve $(i)$ uzvu $(i-1)$ uzvuna göre döndüğünde değişmektedir.




Şekil A.1. Denavit-Hartenberg notasyonu

Tablo A.5. Puma 560'a ait Denavit-Hartenberg parametreleri

| $i$ | $\alpha_{i-1}$ (derece) | $\theta_i$ | $a_{i-1}$ (metre) | $d_i$ (metre) |
|---|---|---|---|---|
| 1 | 0 | $\theta_1$ | 0 | 0 |
| 2 | -90 | $\theta_2$ | 0 | 0.2435 |
| 3 | - | $\theta_3$ | 0.4318 | -0.0934 |
| 4 | 90 | $\theta_4$ | -0.0203 | 0.4331 |
| 5 | -90 | $\theta_5$ | 0 | 0 |
| 6 | 90 | $\theta_6$ | 0 | 0 |

Her bir koordinat çerçevesi aşağıdaki üç temel kurala göre yerleştirilmektedir.

1. $\vec{z}_{i-1}$ ekseni $(i)$.eklemin hareket ekseni boyunca yerleştirilir,
2. $\vec{x}_i$ ekseni $\vec{z}_{i-1}$ eksenine diktir (ucu dışarıya doğru olacak şekilde),
3. $\vec{y}_i$ ekseni sağ el kuralına göre koordinat sistemini tamamlayacak şekildedir.

Yukarıdaki kurallara uyularak ( $\vec{x}_0, \vec{y}_0, \vec{z}_0$ ) referans çerçevesi, $\vec{z}_0$ ekseni ilk eklemin hareket ekseni boyunca olacak şekilde temel uzuv üzerinde herhangi bir yere yerleştirilebilir. Son koordinat çerçevesi ($n$.çerçeve), $\vec{x}_n$ ekseni $\vec{z}_{n-1}$ eksenine dik olacak şekilde el üzerinde herhangi bir yere yerleştirilebilir.

$(i)$ uzvunun koordinat sisteminde tanımlanan herhangi bir vektörü, $(i-1)$ uzvunun koordinat sistemine taşımak için $A_{i-1}^{i}$ homojen dönüşüm matrisi kullanılmaktadır. Denavit-Hartenberg parametrelerini kullanarak ( $\vec{x}_i, \vec{y}_i, \vec{z}_i$ ) koordinat sistemde tanımlamayı gerçekleştiren homojen dönüşüm matrisi aşağıdaki şekilde verilmektedir.

$$A_{i-1}^{i} = \begin{bmatrix} Cos\theta_i & -Cos\alpha_i\ Sin\theta_i & Sin\alpha_i\ Sin\theta_i & a_i\ Cos\theta_i \\ Sin\theta_i & Cos\alpha_i\ Cos\theta_i & -Sin\alpha_i\ Cos\theta_i & a_i\ Sin\theta_i \\ 0 & Sin\alpha_i & Cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Puma 560 robot kolu $A_{i-1}^{i}$ homojen dönüşüm matrisi aşağıdaki gibi hesaplanmaktadır:

$${}^{0}A_1 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad {}^{1}A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^{2}A_3 = \begin{bmatrix} C_3 & 0 & S_3 & a_3C_3 \\ S_3 & 0 & -C_3 & a_3S_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{3}A_4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^{4}A_5 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{5}A_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T={}^{0}A_{1}\,{}^{1}A_{2}\,{}^{2}A_{3}\,{}^{3}A_{4}\,{}^{4}A_{5}\,{}^{5}A_{6}=\begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n_x = C_1\left[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6\right] - S_1(S_4C_5C_6 + C_4S_6)$$

$$n_y = S_1\left[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6\right] + C_1(S_4C_5C_6 + C_4S_6)$$

$$n_z = -S_{23}\left[C_4C_5C_6 - S_4S_6\right] - C_{23}S_5C_6$$

$$s_x = C_1\left[-C_{23}(C_4C_5S_6 + S_4C_6) + S_{23}S_5S_6\right] - S_1(-S_4C_5S_6 + C_4C_6)$$

$$s_y = S_1\left[-C_{23}(C_4C_5S_6 + S_4C_6) + S_{23}S_5S_6\right] + C_1(-S_4C_5S_6 + C_4C_6)$$

$$s_z = S_{23}(C_4C_5S_6 + S_4C_6) + C_{23}S_5S_6$$

$$a_x = C_1(C_{23}C_4S_5 + S_{23}C_5) - S_1S_4S_5$$

$$a_y = S_1(C_{23}C_4S_5 + S_{23}C_5) + C_1S_4S_5$$

$$a_z = -S_{23}C_4S_5 + C_{23}C_5$$

$$p_x = C_1\left[d_6(C_{23}C_4S_5 + S_{23}C_5) + S_{23}d_4 + a_3C_{23} + a_2C_2\right] - S_1(d_6S_4S_5 + d_2)$$

$$p_y = S_1\left[d_6(C_{23}C_4S_5 + S_{23}C_5) + S_{23}d_4 + a_3C_{23} + a_2C_2\right] + C_1(d_6S_4S_5 + d_2)$$

$$p_z = d_6(C_{23}C_5 - S_{23}C_4S_5) + C_{23}d_4 - a_3S_{23} - a_2S_2$$

## Ek B. 4. Mertebeden Runge-Kutta Bütünleştirme Yöntemleri

$n$ eklemli bir robot kolunun dinamik davranışı, birbiriyle yüksek oranda etkileşimli $n$ tane ikinci dereceden doğrusal olmayan (non-linear) diferansiyel denklem ile ifade edilebilir. Bu tür diferansiyel denklemleri normal matematiksel yöntemler kullanarak çözmek imkansızdır. Bu yüzden, uygun bir nümerik analiz yöntemine gereksinim vardır. Runge-Kutta bütünleştirme (integration) yöntemi bu bakımdan en idealidir. Değişik mertebeden Runge-Kutta bütünleştirme yöntemleri olmasına rağmen yaygın olarak kullanılanı 4. mertebeden olanıdır. Bu bölümde, 4. mertebenden Runge-Kutta bütünleştirme yönteminin birinci ve ikinci dereceden diferansiyel denklemlere uygulanışları anlatılacaktır.

### Ek B.1. 4. Mertebeden runge-kutta bütünleştirme yönteminin birinci dereceden diferansiyel denkleme uygulanışı

Birinci dereceden diferansiyel denklemi $\dot{q}(t) = f(t, q(t))$, başlangıç şartı $q(t_0) = q_0$ olarak verilen $q(t)$ fonksiyonu $[t_0, t_N]$ aralığında 4. mertebeden Runge-Kutta bütünleştirme yöntemi ile üç adımda şu şekilde hesaplanır.

1. $[t_0, t_N]$ aralığını $N$ parçaya böl. (N adım sayısı)

   $h = (t_N - t_0)/N$ (h adım uzunluğu)

   $t_{n+1} = t_0 + (n+1)\cdot h, \qquad n = 0, 1, \cdots, N-1$

   $n = 0$ ile başla.

2. $k_1 = h \cdot f(t_n, q_n)$

   $k_2 = h \cdot f(t_n + \frac{h}{2}, q_n + \frac{k_1}{2})$

   $k_3 = h \cdot f(t_n + \frac{h}{2}, q_n + \frac{k_2}{2})$

   $k_4 = h \cdot f(t_n + h, q_n + k_3)$

   $t_{n+1} = t_n + h$

$$q_{n+1} = q_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

$$n = n + 1$$

3. $n = N$ ise bitir değilse Adım 2'ye git.

**Ek B.2. 4. Mertebeden runge-kutta bütünleştirme yönteminin birinci dereceden birbiriyle etkileşimli altı diferansiyel denkleme uygulanışı**

Birinci dereceden diferansiyel denklemleri

$$\dot{q}_1(t) = f_1(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t)),$$
$$\dot{q}_2(t) = f_2(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t)),$$
$$\dot{q}_3(t) = f_3(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t)),$$
$$\dot{q}_4(t) = f_4(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t)),$$
$$\dot{q}_5(t) = f_5(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t)),$$
$$\dot{q}_6(t) = f_6(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t))$$

olan ve başlangıç şartları

$$q_1(t_0) = q_{10}, \quad q_2(t_0) = q_{20}, \quad q_3(t_0) = q_{30}, \; q_4(t_0) = q_{40}, \; q_5(t_0) = q_{50}, \; q_6(t_0) = q_{60}$$

olarak verilen birbiriyle etkileşimli $q_1(t)$, $q_2(t)$, $q_3(t)$, $q_4(t)$, $q_5(t)$, $q_6(t)$ fonksiyonları $[t_0, t_N]$ aralığında 4. mertebeden Runge-Kutta bütünleştirme yöntemi ile üç adımda şu şekilde hesaplanır.

1. $[t_0, t_N]$ aralığını $N$ parçaya böl. (N adım sayısı)

   $h = (t_N - t_0)/N$ (h adım uzunluğu)

   $t_{n+1} = t_0 + (n+1) \cdot h,$ $\qquad n = 0, 1, \cdots, N-1$

   $n = 0$ ile başla.

2. $k_{11} = h \cdot f_1(t_n, q_{1n}, q_{2n}, q_{3n}, q_{4n}, q_{5n}, q_{6n})$

$$k_{12} = h \cdot f_2(t_n, q_{1n}, q_{2n}, q_{3n}, q_{4n}, q_{5n}, q_{6n})$$

$$k_{13} = h \cdot f_3(t_n, q_{1n}, q_{2n}, q_{3n}, q_{4n}, q_{5n}, q_{6n})$$

$$k_{14} = h \cdot f_4(t_n, q_{1n}, q_{2n}, q_{3n}, q_{4n}, q_{5n}, q_{6n})$$

$$k_{15} = h \cdot f_5(t_n, q_{1n}, q_{2n}, q_{3n}, q_{4n}, q_{5n}, q_{6n})$$

$$k_{16} = h \cdot f_6(t_n, q_{1n}, q_{2n}, q_{3n}, q_{4n}, q_{5n}, q_{6n})$$

$$k_{21} = h \cdot f_1(t_n + \frac{h}{2}, q_{1n} + \frac{k_{11}}{2}, q_{2n} + \frac{k_{12}}{2}, q_{3n} + \frac{k_{13}}{2}, q_{4n} + \frac{k_{14}}{2}, q_{5n} + \frac{k_{15}}{2}, q_{6n} + \frac{k_{16}}{2})$$

$$k_{22} = h \cdot f_2(t_n + \frac{h}{2}, q_{1n} + \frac{k_{11}}{2}, q_{2n} + \frac{k_{12}}{2}, q_{3n} + \frac{k_{13}}{2}, q_{4n} + \frac{k_{14}}{2}, q_{5n} + \frac{k_{15}}{2}, q_{6n} + \frac{k_{16}}{2})$$

$$k_{23} = h \cdot f_3(t_n + \frac{h}{2}, q_{1n} + \frac{k_{11}}{2}, q_{2n} + \frac{k_{12}}{2}, q_{3n} + \frac{k_{13}}{2}, q_{4n} + \frac{k_{14}}{2}, q_{5n} + \frac{k_{15}}{2}, q_{6n} + \frac{k_{16}}{2})$$

$$k_{24} = h \cdot f_4(t_n + \frac{h}{2}, q_{1n} + \frac{k_{11}}{2}, q_{2n} + \frac{k_{12}}{2}, q_{3n} + \frac{k_{13}}{2}, q_{4n} + \frac{k_{14}}{2}, q_{5n} + \frac{k_{15}}{2}, q_{6n} + \frac{k_{16}}{2})$$

$$k_{25} = h \cdot f_5(t_n + \frac{h}{2}, q_{1n} + \frac{k_{11}}{2}, q_{2n} + \frac{k_{12}}{2}, q_{3n} + \frac{k_{13}}{2}, q_{4n} + \frac{k_{14}}{2}, q_{5n} + \frac{k_{15}}{2}, q_{6n} + \frac{k_{16}}{2})$$

$$k_{26} = h \cdot f_6(t_n + \frac{h}{2}, q_{1n} + \frac{k_{11}}{2}, q_{2n} + \frac{k_{12}}{2}, q_{3n} + \frac{k_{13}}{2}, q_{4n} + \frac{k_{14}}{2}, q_{5n} + \frac{k_{15}}{2}, q_{6n} + \frac{k_{16}}{2})$$

$$k_{31} = h \cdot f_1(t_n + \frac{h}{2}, q_{1n} + \frac{k_{21}}{2}, q_{2n} + \frac{k_{22}}{2}, q_{3n} + \frac{k_{23}}{2}, q_{4n} + \frac{k_{24}}{2}, q_{5n} + \frac{k_{25}}{2}, q_{6n} + \frac{k_{26}}{2})$$

$$k_{32} = h \cdot f_2(t_n + \frac{h}{2}, q_{1n} + \frac{k_{21}}{2}, q_{2n} + \frac{k_{22}}{2}, q_{3n} + \frac{k_{23}}{2}, q_{4n} + \frac{k_{24}}{2}, q_{5n} + \frac{k_{25}}{2}, q_{6n} + \frac{k_{26}}{2})$$

$$k_{33} = h \cdot f_3(t_n + \frac{h}{2}, q_{1n} + \frac{k_{21}}{2}, q_{2n} + \frac{k_{22}}{2}, q_{3n} + \frac{k_{23}}{2}, q_{4n} + \frac{k_{24}}{2}, q_{5n} + \frac{k_{25}}{2}, q_{6n} + \frac{k_{26}}{2})$$

$$k_{34} = h \cdot f_4(t_n + \frac{h}{2}, q_{1n} + \frac{k_{21}}{2}, q_{2n} + \frac{k_{22}}{2}, q_{3n} + \frac{k_{23}}{2}, q_{4n} + \frac{k_{24}}{2}, q_{5n} + \frac{k_{25}}{2}, q_{6n} + \frac{k_{26}}{2})$$

$$k_{35} = h \cdot f_5(t_n + \frac{h}{2}, q_{1n} + \frac{k_{21}}{2}, q_{2n} + \frac{k_{22}}{2}, q_{3n} + \frac{k_{23}}{2}, q_{4n} + \frac{k_{24}}{2}, q_{5n} + \frac{k_{25}}{2}, q_{6n} + \frac{k_{26}}{2})$$

$$k_{36} = h \cdot f_6(t_n + \frac{h}{2}, q_{1n} + \frac{k_{21}}{2}, q_{2n} + \frac{k_{22}}{2}, q_{3n} + \frac{k_{23}}{2}, q_{4n} + \frac{k_{24}}{2}, q_{5n} + \frac{k_{25}}{2}, q_{6n} + \frac{k_{26}}{2})$$

$$k_{41} = h \cdot f_1(t_n + h, q_{1n} + k_{31}, q_{2n} + k_{32}, q_{3n} + k_{33}, q_{4n} + k_{34}, q_{5n} + k_{35}, q_{6n} + k_{36})$$

$$k_{42} = h \cdot f_2(t_n + h, q_{1n} + k_{31}, q_{2n} + k_{32}, q_{3n} + k_{33}, q_{4n} + k_{34}, q_{5n} + k_{35}, q_{6n} + k_{36})$$

$$k_{43} = h \cdot f_3(t_n + h, q_{1n} + k_{31}, q_{2n} + k_{32}, q_{3n} + k_{33}, q_{4n} + k_{34}, q_{5n} + k_{35}, q_{6n} + k_{36})$$

$$k_{44} = h \cdot f_4(t_n + h, q_{1n} + k_{31}, q_{2n} + k_{32}, q_{3n} + k_{33}, q_{4n} + k_{34}, q_{5n} + k_{35}, q_{6n} + k_{36})$$

$$k_{45} = h \cdot f_5(t_n + h, q_{1n} + k_{31}, q_{2n} + k_{32}, q_{3n} + k_{33}, q_{4n} + k_{34}, q_{5n} + k_{35}, q_{6n} + k_{36})$$

$$k_{46} = h \cdot f_6(t_n + h, q_{1n} + k_{31}, q_{2n} + k_{32}, q_{3n} + k_{33}, q_{4n} + k_{34}, q_{5n} + k_{35}, q_{6n} + k_{36})$$

$$t_{n+1} = t_n + h$$

$$q_{1(n+1)} = q_{1n} + \frac{k_{11} + 2k_{21} + 2k_{31} + k_{41}}{6}$$

$$q_{2(n+1)} = q_{2n} + \frac{k_{12} + 2k_{22} + 2k_{32} + k_{42}}{6}$$

$$q_{3(n+1)} = q_{3n} + \frac{k_{13} + 2k_{23} + 2k_{33} + k_{43}}{6}$$

$$q_{4(n+1)} = q_{4n} + \frac{k_{14} + 2k_{24} + 2k_{34} + k_{44}}{6}$$

$$q_{5(n+1)} = q_{5n} + \frac{k_{15} + 2k_{25} + 2k_{35} + k_{45}}{6}$$

$$q_{6(n+1)} = q_{6n} + \frac{k_{16} + 2k_{26} + 2k_{36} + k_{46}}{6}$$

$$n = n + 1$$

3. $n = N$ ise bitir değilse Adım 2'ye git.

**Ek B.3. 4. Mertebeden runge-kutta bütünleştirme yönteminin ikinci dereceden diferansiyel denkleme uygulanışı**

İkinci dereceden diferansiyel denklemi $\ddot{q}(t) = f(t, q(t), \dot{q}(t))$, başlangıç şartları $q(t_0) = q_0$ ve $\dot{q}(t_0) = \dot{q}_0$ olarak verilen $q(t)$ fonksiyonu ve türevi $\dot{q}(t)$ $[t_0, t_N]$ aralığında 4. mertebeden Runge-Kutta bütünleştirme yöntemi ile üç adımda şu şekilde hesaplanır.

1. $[t_0, t_N]$ aralığını $N$ parçaya böl. (N adım sayısı)

   $h = (t_N - t_0)/N$ (h adım uzunluğu)

   $t_{n+1} = t_0 + (n+1) \cdot h$, $\qquad n = 0, 1, \cdots, N-1$

   $n = 0$ ile başla.

2. $k_1 = \frac{h}{2} \cdot f(t_n, q_n, \dot{q}_n)$

$l = \frac{h}{2} \cdot (\dot{q}_n + \frac{k_1}{2})$

$k_2 = \frac{h}{2} \cdot f(t_n + \frac{h}{2}, q_n + l, \dot{q}_n + k_1)$

$k_3 = \frac{h}{2} \cdot f(t_n + \frac{h}{2}, q_n + l, \dot{q}_n + k_2)$

$m = h \cdot (\dot{q}_n + k_3)$

$k_4 = \frac{h}{2} \cdot f(t_n + h, q_n + m, \dot{q}_n + 2k_3)$

$t_{n+1} = t_n + h$

$q_{n+1} = q_n + h \cdot (\dot{q}_n + \frac{k_1 + k_2 + k_3}{3})$

$\dot{q}_{n+1} = \dot{q}_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{3}$

$n = n + 1$

3. $n = N$ ise bitir değilse Adım 2'ye git.

**Ek B.4. 4. Mertebeden runge-kutta bütünleştirme yönteminin ikinci dereceden birbiriyle etkileşimli altı diferansiyel denkleme uygulanışı**

İkinci dereceden diferansiyel denklemleri

$$\ddot{q}_1(t) = f_1(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t), \dot{q}_1(t), \dot{q}_2(t), \dot{q}_3(t), \dot{q}_4(t), \dot{q}_5(t), \dot{q}_6(t)),$$

$$\ddot{q}_2(t) = f_2(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t), \dot{q}_1(t), \dot{q}_2(t), \dot{q}_3(t), \dot{q}_4(t), \dot{q}_5(t), \dot{q}_6(t)),$$

$$\ddot{q}_3(t) = f_3(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t), \dot{q}_1(t), \dot{q}_2(t), \dot{q}_3(t), \dot{q}_4(t), \dot{q}_5(t), \dot{q}_6(t)),$$

$$\ddot{q}_4(t) = f_4(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t), \dot{q}_1(t), \dot{q}_2(t), \dot{q}_3(t), \dot{q}_4(t), \dot{q}_5(t), \dot{q}_6(t)),$$

$$\ddot{q}_5(t) = f_5(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t), \dot{q}_1(t), \dot{q}_2(t), \dot{q}_3(t), \dot{q}_4(t), \dot{q}_5(t), \dot{q}_6(t)),$$

$$\ddot{q}_6(t) = f_6(t, q_1(t), q_2(t), q_3(t), q_4(t), q_5(t), q_6(t), \dot{q}_1(t), \dot{q}_2(t), \dot{q}_3(t), \dot{q}_4(t), \dot{q}_5(t), \dot{q}_6(t))$$

olan ve başlangıç şartları

$$q_1(t_0)=q_{10},\ q_2(t_0)=q_{20},\ q_3(t_0)=q_{30},\ q_4(t_0)=q_{40},\ q_5(t_0)=q_{50},\ q_6(t_0)=q_{60}$$
$$\dot{q}_1(t_0)=\dot{q}_{10},\ \dot{q}_2(t_0)=\dot{q}_{20},\ \dot{q}_3(t_0)=\dot{q}_{30},\ \dot{q}_4(t_0)=\dot{q}_{40},\ \dot{q}_5(t_0)=\dot{q}_{50},\ \dot{q}_6(t_0)=\dot{q}_{60}$$

olarak verilen birbiriyle etkileşimli $q_1(t),\ q_2(t),\ q_3(t),\ q_4(t),\ q_5(t),\ q_6(t)$ fonksiyonları ve onların $\dot{q}_1(t),\ \dot{q}_2(t),\ \dot{q}_3(t),\ \dot{q}_4(t),\ \dot{q}_5(t),\ \dot{q}_6(t)$ türevleri $[t_0,t_N]$ aralığında dördüncü mertebeden Runge-Kutta bütünleştirme yöntemi ile üç adımda şu şekilde hesaplanır.

1. $[t_0,t_N]$ aralığını $N$ parçaya böl. (N adım sayısı)

   $h=(t_N-t_0)/N$ (h adım uzunluğu)

   $t_{n+1}=t_0+(n+1)\cdot h,\qquad n=0,1,\cdots,N-1$

   $n=0$ ile başla

2. $$k_{11}=\frac{h}{2}\cdot f_1(t_n,q_{1n},q_{2n},q_{3n},q_{4n},q_{5n},q_{6n},\dot{q}_{1n},\dot{q}_{2n},\dot{q}_{3n},\dot{q}_{4n},\dot{q}_{5n},\dot{q}_{6n})$$

   $$k_{12}=\frac{h}{2}\cdot f_2(t_n,q_{1n},q_{2n},q_{3n},q_{4n},q_{5n},q_{6n},\dot{q}_{1n},\dot{q}_{2n},\dot{q}_{3n},\dot{q}_{4n},\dot{q}_{5n},\dot{q}_{6n})$$

   $$k_{13}=\frac{h}{2}\cdot f_3(t_n,q_{1n},q_{2n},q_{3n},q_{4n},q_{5n},q_{6n},\dot{q}_{1n},\dot{q}_{2n},\dot{q}_{3n},\dot{q}_{4n},\dot{q}_{5n},\dot{q}_{6n})$$

   $$k_{14}=\frac{h}{2}\cdot f_4(t_n,q_{1n},q_{2n},q_{3n},q_{4n},q_{5n},q_{6n},\dot{q}_{1n},\dot{q}_{2n},\dot{q}_{3n},\dot{q}_{4n},\dot{q}_{5n},\dot{q}_{6n})$$

   $$k_{15}=\frac{h}{2}\cdot f_5(t_n,q_{1n},q_{2n},q_{3n},q_{4n},q_{5n},q_{6n},\dot{q}_{1n},\dot{q}_{2n},\dot{q}_{3n},\dot{q}_{4n},\dot{q}_{5n},\dot{q}_{6n})$$

   $$k_{16}=\frac{h}{2}\cdot f_6(t_n,q_{1n},q_{2n},q_{3n},q_{4n},q_{5n},q_{6n},\dot{q}_{1n},\dot{q}_{2n},\dot{q}_{3n},\dot{q}_{4n},\dot{q}_{5n},\dot{q}_{6n})$$

   $$l_1=\frac{h}{2}\cdot(\dot{q}_{1n}+\frac{k_{11}}{2}),\quad l_2=\frac{h}{2}\cdot(\dot{q}_{2n}+\frac{k_{12}}{2}),\quad l_3=\frac{h}{2}\cdot(\dot{q}_{3n}+\frac{k_{13}}{2})$$

   $$l_4=\frac{h}{2}\cdot(\dot{q}_{4n}+\frac{k_{14}}{2}),\quad l_5=\frac{h}{2}\cdot(\dot{q}_{5n}+\frac{k_{15}}{2}),\quad l_6=\frac{h}{2}\cdot(\dot{q}_{6n}+\frac{k_{16}}{2})$$

$$k_{21}=\frac{h}{2}\cdot f_1(t_n+\frac{h}{2},\, q_{1n}+l_1,\, q_{2n}+l_2,\, q_{3n}+l_3,\, q_{4n}+l_4,\, q_{5n}+l_5,\, q_{6n}+l_6,$$
$$\dot{q}_{1n}+k_{11},\, \dot{q}_{2n}+k_{12},\, \dot{q}_{3n}+k_{13},\, \dot{q}_{4n}+k_{14},\, \dot{q}_{5n}+k_{15},\, \dot{q}_{6n}+k_{16})$$

$$k_{22}=\frac{h}{2}\cdot f_2(t_n+\frac{h}{2},\, q_{1n}+l_1,\, q_{2n}+l_2,\, q_{3n}+l_3,\, q_{4n}+l_4,\, q_{5n}+l_5,\, q_{6n}+l_6,$$
$$\dot{q}_{1n}+k_{11},\, \dot{q}_{2n}+k_{12},\, \dot{q}_{3n}+k_{13},\, \dot{q}_{4n}+k_{14},\, \dot{q}_{5n}+k_{15},\, \dot{q}_{6n}+k_{16})$$

$$k_{23}=\frac{h}{2}\cdot f_3(t_n+\frac{h}{2},\, q_{1n}+l_1,\, q_{2n}+l_2,\, q_{3n}+l_3,\, q_{4n}+l_4,\, q_{5n}+l_5,\, q_{6n}+l_6,$$
$$\dot{q}_{1n}+k_{11},\, \dot{q}_{2n}+k_{12},\, \dot{q}_{3n}+k_{13},\, \dot{q}_{4n}+k_{14},\, \dot{q}_{5n}+k_{15},\, \dot{q}_{6n}+k_{16})$$

$$k_{24}=\frac{h}{2}\cdot f_4(t_n+\frac{h}{2},\, q_{1n}+l_1,\, q_{2n}+l_2,\, q_{3n}+l_3,\, q_{4n}+l_4,\, q_{5n}+l_5,\, q_{6n}+l_6,$$
$$\dot{q}_{1n}+k_{11},\, \dot{q}_{2n}+k_{12},\, \dot{q}_{3n}+k_{13},\, \dot{q}_{4n}+k_{14},\, \dot{q}_{5n}+k_{15},\, \dot{q}_{6n}+k_{16})$$

$$k_{25}=\frac{h}{2}\cdot f_5(t_n+\frac{h}{2},\, q_{1n}+l_1,\, q_{2n}+l_2,\, q_{3n}+l_3,\, q_{4n}+l_4,\, q_{5n}+l_5,\, q_{6n}+l_6,$$
$$\dot{q}_{1n}+k_{11},\, \dot{q}_{2n}+k_{12},\, \dot{q}_{3n}+k_{13},\, \dot{q}_{4n}+k_{14},\, \dot{q}_{5n}+k_{15},\, \dot{q}_{6n}+k_{16})$$

$$k_{26}=\frac{h}{2}\cdot f_6(t_n+\frac{h}{2},\, q_{1n}+l_1,\, q_{2n}+l_2,\, q_{3n}+l_3,\, q_{4n}+l_4,\, q_{5n}+l_5,\, q_{6n}+l_6,$$
$$\dot{q}_{1n}+k_{11},\, \dot{q}_{2n}+k_{12},\, \dot{q}_{3n}+k_{13},\, \dot{q}_{4n}+k_{14},\, \dot{q}_{5n}+k_{15},\, \dot{q}_{6n}+k_{16})$$

$$k_{31}=\frac{h}{2}\cdot f_1(t_n+\frac{h}{2},\, q_{1n}+l_1,\, q_{2n}+l_2,\, q_{3n}+l_3,\, q_{4n}+l_4,\, q_{5n}+l_5,\, q_{6n}+l_6,$$
$$\dot{q}_{1n}+k_{21},\, \dot{q}_{2n}+k_{22},\, \dot{q}_{3n}+k_{23},\, \dot{q}_{4n}+k_{24},\, \dot{q}_{5n}+k_{25},\, \dot{q}+k_{26})$$

$$k_{32}=\frac{h}{2}\cdot f_2(t_n+\frac{h}{2},\, q_{1n}+l_1,\, q_{2n}+l_2,\, q_{3n}+l_3,\, q_{4n}+l_4,\, q_{5n}+l_5,\, q_{6n}+l_6,$$
$$\dot{q}_{1n}+k_{21},\, \dot{q}_{2n}+k_{22},\, \dot{q}_{3n}+k_{23},\, \dot{q}_{4n}+k_{24},\, \dot{q}_{5n}+k_{25},\, \dot{q}+k_{26})$$

$$k_{33}=\frac{h}{2}\cdot f_3(t_n+\frac{h}{2},\, q_{1n}+l_1,\, q_{2n}+l_2,\, q_{3n}+l_3,\, q_{4n}+l_4,\, q_{5n}+l_5,\, q_{6n}+l_6,$$
$$\dot{q}_{1n}+k_{21},\, \dot{q}_{2n}+k_{22},\, \dot{q}_{3n}+k_{23},\, \dot{q}_{4n}+k_{24},\, \dot{q}_{5n}+k_{25},\, \dot{q}+k_{26})$$

$$k_{34}=\frac{h}{2}\cdot f_4(t_n+\frac{h}{2},\, q_{1n}+l_1,\, q_{2n}+l_2,\, q_{3n}+l_3,\, q_{4n}+l_4,\, q_{5n}+l_5,\, q_{6n}+l_6,$$
$$\dot{q}_{1n}+k_{21},\, \dot{q}_{2n}+k_{22},\, \dot{q}_{3n}+k_{23},\, \dot{q}_{4n}+k_{24},\, \dot{q}_{5n}+k_{25},\, \dot{q}+k_{26})$$

$$k_{35}=\frac{h}{2}\cdot f_5(t_n+\frac{h}{2},\, q_{1n}+l_1,\, q_{2n}+l_2,\, q_{3n}+l_3,\, q_{4n}+l_4,\, q_{5n}+l_5,\, q_{6n}+l_6,$$
$$\dot{q}_{1n}+k_{21},\, \dot{q}_{2n}+k_{22},\, \dot{q}_{3n}+k_{23},\, \dot{q}_{4n}+k_{24},\, \dot{q}_{5n}+k_{25},\, \dot{q}+k_{26})$$

$$k_{36} = \frac{h}{2} \cdot f_6(t_n + \frac{h}{2}, q_{1n} + l_1, q_{2n} + l_2, q_{3n} + l_3, q_{4n} + l_4, q_{5n} + l_5, q_{6n} + l_6,$$
$$\dot{q}_{1n} + k_{21}, \dot{q}_{2n} + k_{22}, \dot{q}_{3n} + k_{23}, \dot{q}_{4n} + k_{24}, \dot{q}_{5n} + k_{25}, \dot{q} + k_{26})$$

$$m_1 = h \cdot (\dot{q}_{1n} + k_{31}), \quad m_2 = h \cdot (\dot{q}_{2n} + k_{32}), \quad m_3 = h \cdot (\dot{q}_{3n} + k_{33})$$
$$m_4 = h \cdot (\dot{q}_{4n} + k_{34}), \quad m_5 = h \cdot (\dot{q}_{5n} + k_{35}), \quad m_6 = h \cdot (\dot{q}_{6n} + k_{36})$$

$$k_{41} = \frac{h}{2} \cdot f_1(t_n + h, q_{1n} + m_1, q_{2n} + m_2, q_{3n} + m_3, q_{4n} + m_4, q_{5n} + m_5, q_{6n} + m_6,$$
$$\dot{q}_{1n} + 2k_{31}, \dot{q}_{2n} + 2k_{32}, \dot{q}_{3n} + 2k_{33}, \dot{q}_{4n} + 2k_{34}, \dot{q}_{5n} + 2k_{35}, \dot{q}_{6n} + 2k_{36})$$

$$k_{42} = \frac{h}{2} \cdot f_2(t_n + h, q_{1n} + m_1, q_{2n} + m_2, q_{3n} + m_3, q_{4n} + m_4, q_{5n} + m_5, q_{6n} + m_6,$$
$$\dot{q}_{1n} + 2k_{31}, \dot{q}_{2n} + 2k_{32}, \dot{q}_{3n} + 2k_{33}, \dot{q}_{4n} + 2k_{34}, \dot{q}_{5n} + 2k_{35}, \dot{q}_{6n} + 2k_{36})$$

$$k_{43} = \frac{h}{2} \cdot f_3(t_n + h, q_{1n} + m_1, q_{2n} + m_2, q_{3n} + m_3, q_{4n} + m_4, q_{5n} + m_5, q_{6n} + m_6,$$
$$\dot{q}_{1n} + 2k_{31}, \dot{q}_{2n} + 2k_{32}, \dot{q}_{3n} + 2k_{33}, \dot{q}_{4n} + 2k_{34}, \dot{q}_{5n} + 2k_{35}, \dot{q}_{6n} + 2k_{36})$$

$$k_{44} = \frac{h}{2} \cdot f_4(t_n + h, q_{1n} + m_1, q_{2n} + m_2, q_{3n} + m_3, q_{4n} + m_4, q_{5n} + m_5, q_{6n} + m_6,$$
$$\dot{q}_{1n} + 2k_{31}, \dot{q}_{2n} + 2k_{32}, \dot{q}_{3n} + 2k_{33}, \dot{q}_{4n} + 2k_{34}, \dot{q}_{5n} + 2k_{35}, \dot{q}_{6n} + 2k_{36})$$

$$k_{45} = \frac{h}{2} \cdot f_5(t_n + h, q_{1n} + m_1, q_{2n} + m_2, q_{3n} + m_3, q_{4n} + m_4, q_{5n} + m_5, q_{6n} + m_6,$$
$$\dot{q}_{1n} + 2k_{31}, \dot{q}_{2n} + 2k_{32}, \dot{q}_{3n} + 2k_{33}, \dot{q}_{4n} + 2k_{34}, \dot{q}_{5n} + 2k_{35}, \dot{q}_{6n} + 2k_{36})$$

$$k_{46} = \frac{h}{2} \cdot f_6(t_n + h, q_{1n} + m_1, q_{2n} + m_2, q_{3n} + m_3, q_{4n} + m_4, q_{5n} + m_5, q_{6n} + m_6,$$
$$\dot{q}_{1n} + 2k_{31}, \dot{q}_{2n} + 2k_{32}, \dot{q}_{3n} + 2k_{33}, \dot{q}_{4n} + 2k_{34}, \dot{q}_{5n} + 2k_{35}, \dot{q}_{6n} + 2k_{36})$$

$$t_{n+1} = t_n + h$$

$$q_{1(n+1)} = q_{1n} + h \cdot (\dot{q}_{1n} + \frac{k_{11} + k_{21} + k_{31}}{3})$$

$$q_{2(n+1)} = q_{2n} + h \cdot (\dot{q}_{2n} + \frac{k_{12} + k_{22} + k_{32}}{3})$$

$$q_{3(n+1)} = q_{3n} + h \cdot (\dot{q}_{3n} + \frac{k_{13} + k_{23} + k_{33}}{3})$$

$$q_{4(n+1)} = q_{4n} + h \cdot (\dot{q}_{4n} + \frac{k_{14} + k_{24} + k_{34}}{3})$$

$$q_{5(n+1)} = q_{5n} + h \cdot (\dot{q}_{5n} + \frac{k_{15} + k_{25} + k_{35}}{3})$$

$$q_{6(n+1)} = q_{6n} + h \cdot (\dot{q}_{6n} + \frac{k_{16} + k_{26} + k_{36}}{3})$$

$$\dot{q}_{1(n+1)} = \dot{q}_{1n} + \frac{k_{11} + 2k_{21} + 2k_{31} + k_{41}}{3}$$

$$\dot{q}_{2(n+1)} = \dot{q}_{2n} + \frac{k_{12} + 2k_{22} + 2k_{32} + k_{42}}{3}$$

$$\dot{q}_{3(n+1)} = \dot{q}_{3n} + \frac{k_{13} + 2k_{23} + 2k_{33} + k_{43}}{3}$$

$$\dot{q}_{4(n+1)} = \dot{q}_{4n} + \frac{k_{14} + 2k_{24} + 2k_{34} + k_{44}}{3}$$

$$\dot{q}_{5(n+1)} = \dot{q}_{5n} + \frac{k_{15} + 2k_{25} + 2k_{35} + k_{45}}{3}$$

$$\dot{q}_{6(n+1)} = \dot{q}_{6n} + \frac{k_{16} + 2k_{26} + 2k_{36} + k_{46}}{3}$$

$$n = n + 1$$

3. $n = N$ ise bitir değilse Adım 2'ye git.

# ÖZGEÇMİŞ

10.01.1978 tarihinde Bilecik'te doğdu. İlkokulu, ortaokulu ve liseyi aynı ilde tamamladı. Liseden 1994 yılında mezun oldu. 2000 yılında Dumlupınar Üniversitesinden Elektrik-Elektronik Mühendisi olarak mezun oldu. 2003 yılında Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik anabilim dalında yüksek lisansını tamamladı. Aynı yıl Sakarya Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik anabilim dalında doktoraya başladı ve halen doktora çalışmaları devam etmektedir. 2000-2004 yılları arası Dumlupınar Üniversitesi, 2004'ten beri ise Sakarya Üniversitesinde Araştırma Görevlisi olarak çalışmaktadır. Evlidir.