**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL**

# DESIGN, ANALYSIS AND DEVELOPMENT OF OPTIMAL SATELLITE ATTITUDE CONTROL SYSTEM

**M.Sc. THESIS**

**Emre SAYIN**

**Department of Aeronautical and Astronautical Engineering**

**Aeronautical and Astronautical Engineering Programme**

**JUNE 2023**

**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL**

# DESIGN, ANALYSIS AND DEVELOPMENT OF OPTIMAL SATELLITE ATTITUDE CONTROL SYSTEM

**M.Sc. THESIS**

**Emre SAYIN**
**(511201121)**

**Department of Aeronautical and Astronautical Engineering**

**Aeronautical and Astronautical Engineering Programme**

**Thesis Advisor: Assist. Prof. Dr. Ismail BAYEZIT**

**JUNE 2023**

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

# OPTİMAL UYDU YÖNELİM KONTROL SİSTEMİ TASARIM, ANALİZ VE GELİŞTİRİLMESİ

**YüKSEK LİSANS TEZİ**

**Emre SAYIN**
**(511201121)**

**Uçak ve Uzay Mühendisliği Anabilim Dalı**

**Uçak ve Uzay Mühendisliği Programı**

**Tez Danışmanı: Dr. Öğr. Üyesi İsmail BAYEZİT**

**HAZİRAN 2023**

Emre SAYIN, a M.Sc. student of ITU Graduate School student ID 511201121, successfully defended the thesis entitled "DESIGN, ANALYSIS AND DEVELOPMENT OF OPTIMAL SATELLITE ATTITUDE CONTROL SYSTEM", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

| | | |
|---|---|---|
| **Thesis Advisor :** | **Assist. Prof. Dr. Ismail BAYEZIT**<br>Istanbul Technical University | ..... |
| **Jury Members :** | **Prof. Dr. Ömer Kemal KINACI**<br>Istanbul Technical University | ..... |
| | **Assist. Prof. Dr. Serhat YILMAZ**<br>National Defence University | ..... |

**Date of Submission : 31 May 2023**
**Date of Defense : 14 June 2023**

***To my family,***

## FOREWORD

I would like to express my heartfelt gratitude to my thesis advisor, Assist. Prof. Dr. İsmail BAYEZİT, for his unwavering support throughout my master's education and the writing process of this thesis.

I also extend my sincere thanks to my beloved spouse, Yağmur Su, for her tremendous support and patience during this journey. I am also grateful to my parents for their support in my education, which brought me to this significant milestone in my life.

June 2023

Emre SAYIN
(Astronautical Engineer)

**TABLE OF CONTENTS**

**Page**

**FOREWORD** ..... **ix**
**TABLE OF CONTENTS** ..... **xi**
**ABBREVIATIONS** ..... **xiii**
**SYMBOLS** ..... **xv**
**LIST OF TABLES** ..... **xvii**
**LIST OF FIGURES** ..... **xix**
**SUMMARY** ..... **xxi**
**ÖZET** ..... **xxiii**
**1. INTRODUCTION** ..... **1**
1.1 Purpose of Thesis ..... 2
1.2 Literature Review ..... 2
1.3 Outline of Thesis ..... 3
**2. CONCEPTUAL BACKGROUND** ..... **5**
2.1 Reference Frames ..... 5
2.1.1 Earth-centered Earth-fixed (ECEF) reference frame ..... 5
2.1.2 Earth-centered inertial (ECI) reference frame ..... 5
2.1.3 Local-vertical local-horizontal reference frame (LVLH) ..... 6
2.1.4 Body-fixed reference frame ..... 6
2.2 Euler Angles ..... 6
2.3 Quaternions ..... 8
2.4 Conversion Between Euler Angles and Quaternions ..... 10
2.5 Skew-Symmetic Matrix ..... 10
**3. SATELLITE EQUATIONS OF MOTION** ..... **13**
3.1 Satellite Dynamics ..... 14
3.2 Satellite Kinematics ..... 15
3.3 Disturbance Modeling ..... 17
3.3.1 Gravity gradient ..... 17
3.3.2 Aerodynamic drag ..... 17
3.4 Nonlinear State Space Model ..... 18
3.4.1 Model validation ..... 18
3.4.2 Dimension reduction of the nonlinear state space model ..... 20
3.5 Linearizaton of Equations of Motion ..... 21
3.5.1 Analytical linearization ..... 23
3.5.2 Numerical linearization ..... 26
3.6 Comparison of Nonlinear and Linear Models ..... 27
**4. CONTROLLER DESIGN** ..... **31**
4.1 Linear Quadratic Regulator ..... 31
4.1.1 Controller tuning ..... 35

4.2 Model Predictive Controller ..... 38
4.3 Comparison of Controllers ..... 42
**5. CONCLUSIONS AND RECOMMENDATIONS ..... 45**
**REFERENCES ..... 47**
**APPENDICES ..... 51**
APPENDIX A.1: MATLAB Function block: "SCmodel_7state" ..... 53
APPENDIX A.2: MATLAB Function block: "quat2Euler" ..... 55
APPENDIX A.3: MATLAB Code: Controllability Check Matrix ..... 57
**CURRICULUM VITAE ..... 59**

## ABBREVIATIONS

| | | |
|---|---|---|
| **ADCS** | : | Attitude Determination and Control System |
| **DCM** | : | Direction Cosine Matrix |
| **ECEF** | : | Earth-Centered Earth-Fixed |
| **ECI** | : | Earth-Centered Inertial |
| **HIL** | : | Hardware-in-the-loop |
| **LEO** | : | Low Earth Orbit |
| **LQR** | : | Linear Quadratic Regulator |
| **LVLH** | : | Local-Vertical Local-Horizontal |
| **MPC** | : | Model Predictive Controller |
| **PIL** | : | Processor-in-the-loop |
| **PWM** | : | Pulse Width Modulation |
| **RMSE** | : | Root Mean Square Error |

## SYMBOLS

| | |
|---|---|
| $\mathbf{A}$ | : State matrix |
| $\boldsymbol{A_1}$ | : Rotation matrix around x-axis |
| $\boldsymbol{A_2}$ | : Rotation matrix around y-axis |
| $\boldsymbol{A_3}$ | : Rotation matrix around z-axis |
| $\boldsymbol{A_{DCM}}$ | : Direction Cosine Matrix |
| $\mathbf{A_{ref}}$ | : Reference area of the satellite that faces with incoming air |
| $\mathbf{B}$ | : Input matrix |
| $\mathbf{C}$ | : Output matrix |
| $\mathbf{c_D}$ | : Drag coefficient |
| $\mathbf{D}$ | : Feed-forward matrix |
| $\mathbf{F_d}$ | : Drag force |
| $\vec{\boldsymbol{g}}$ | : Gibbs vector |
| $\mathbf{G_x}, \mathbf{G_y}, \mathbf{G_z}$ | : Gravity gradient torque elements |
| $\mathbf{h_I}$ | : Angular momentum represented in inertial frame |
| $\mathbf{h_s}$ | : Satellite angular momentum |
| $\mathbf{I}$ | : Identity matrix |
| $\mathbf{I_s}$ | : Satellite moment of inertia matrix |
| $\mathbf{I_{xx}}, \mathbf{I_{yy}}, \mathbf{I_{zz}}$ | : Diagonal elements of the satellite moment of inertia matrix |
| $\mathbf{J}$ | : Cost function |
| $\mathbf{M}$ | : Controllability matrix |
| $\boldsymbol{p_1}$,$\boldsymbol{p_2}$ | : Points on the surface of sphere |
| $\boldsymbol{q}_0, \boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3$ | : Quaternion elements |
| $\boldsymbol{q}$ | : Quaternion |
| $\bar{\mathbf{q}}$ | : Quaternion adjoint |
| $\boldsymbol{q}^{-1}$ | : Quaternion inverse |
| $\|\|\mathbf{q}\|\|$ | : Quaternion norm |
| $\mathbf{T_d}$ | : External moment |
| $\mathbf{v}$ | : Linear velocity vector |
| $\vec{\boldsymbol{v}}$ | : Vector on the surface of sphere |
| $\boldsymbol{\alpha}$ | : Angle of rotation on sphere |
| $\boldsymbol{\phi}$ | : Roll |
| $\boldsymbol{\theta}$ | : Pitch |
| $\boldsymbol{\psi}$ | : Yaw |
| $\rho$ | : Air density |
| $\tau_\mathbf{c}$ | : Control torque |
| $\tau_\mathbf{d}$ | : Disturbance torque |
| $\omega_\mathbf{1}, \omega_\mathbf{2}, \omega_\mathbf{3}$ | : Angular velocity components |
| $\omega$ | : Angular velocity |
| $\omega_\times$ | : Skew-symmetric cross product matrix of angular velocity |

## LIST OF TABLES

**Page**

**Table 4.1** **:** MPC simulation parameters..... **41**

## LIST OF FIGURES

**Page**


**Figure 2.1** : Visualization of ECI, LVLH and body-fixed frames. ..... **6**
**Figure 2.2** : Euler angle coordinate transformation sequence. ..... **7**
**Figure 2.3** : Visual representation of quaternions. ..... **9**
**Figure 3.1** : Open loop block diagram. ..... **13**
**Figure 3.2** : Nonlinear state space model constructed in Simulink. ..... **19**
**Figure 3.3** : Variation of angles for a given test input. ..... **19**
**Figure 3.4** : Variation of angular velocity for a given test input. ..... **20**
**Figure 3.5** : Plots of $f(x) = x$ and sine function. ..... **22**
**Figure 3.6** : Visualization of the central difference method. ..... **27**
**Figure 3.7** : Nonlinear-linear model comparison on Simulink. ..... **28**
**Figure 3.8** : Output of nonlinear and linear models for given input. ..... **29**
**Figure 4.1** : Closed loop control diagram. ..... **31**
**Figure 4.2** : LQR control diagram. ..... **33**
**Figure 4.3** : LQR simulation output with given references. ..... **34**
**Figure 4.4** : LQR simulation output for a custom reference signal. ..... **35**
**Figure 4.5** : LQR simulation output different Q values. ..... **36**
**Figure 4.6** : LQR simulation output different R values. ..... **36**
**Figure 4.7** : Effect of controller saturation on $q_1$ ..... **37**
**Figure 4.8** : LQR control torque with and without saturation. ..... **37**
**Figure 4.9** : MPC block diagram. ..... **41**
**Figure 4.10** : Step response of the MPC controller. ..... **41**
**Figure 4.11** : Control torques for MPC controller. ..... **42**
**Figure 4.12** : Disturbance rejection performance of LQR and MPC controllers. .... **42**

**DESIGN, ANALYSIS AND DEVELOPMENT OF OPTIMAL SATELLITE ATTITUDE CONTROL SYSTEM**

## SUMMARY

Satellite attitude control is an essential part of satellite missions, as it provides the orientation of the satellite from one point of view to another one. In this thesis, a comprehensive investigation on design, analysis and optimization of a satellite attitude controller are presented, by mostly focusing on enhancing controller performance and robustness. Two primary controllers are investigated as Linear Quadratic Regulator (LQR) and Model Predictive Controller (MPC) and their disturbance rejection capabilities are compared.

Initial chapter provides the fundamental concepts and definitions such as reference frames, Euler angles, quaternions and so on. Conversion between quaternions and Euler angles are also presented. Additionally, nonlinear mathematical model is constructed as dynamics and kinematics subsystems. Disturbance model acting on satellite is also presented as gravity gradient and aerodynamic torque. The total nonlinear model is presented as state space equations. Dimension reduction of the nonlinear model is executed by decreasing the number of quaternion elements. Then, the linearization of nonlinear model is performed. Both analytical and numerical linearization methods are explained. It is shown and validated that the linearized state space system model presents the nonlinear model for small angle deviations.

The controller design focuses on optimal control methods, that are used in many areas as well as satellite attitude control systems. Two primary controller approaches are presented as LQR and MPC. The LQR controller is introduced as a optimal controller algorithm. LQR is based on the minimization of a cost function that is constructed by states and inputs of the system. Weight parameters, Q and R, affecting the performance of the LQR controller are investigated and fine tuning of these parameters is performed. It is observed that the appropriate selection of cost function parameters significantly affects the controller performance, especially in terms of steady state error. LQR controller shows the capability of optimal control on stabilizing and attitude tracking performance of the satellites.

Additionally, Model Predictive Control approach is explored in detail. As distinct from classical control methods, in MPC, a prediction model is utilized; states and outputs are "predicted" by using this model. Construction of prediction equations in state space is explained in detail. MPC control also is an optimal control approach. However, the MPC updates the solution of the optimization problem in each time step and produces updated predictions to be applied on calculation of control input. The MPC has also constraint handling capability, meaning that one can limit the control signals, states and

outputs of the system as a part of optimization problem. In this case, the optimization problem becomes a constrained quadratic programming problem in which the problem can be solved numerically.

A comparative analysis has been executed between LQR and MPC controllers. Their disturbance rejection capabilities are analyzed by utilizing a simulation experiment. A constant disturbance was applied to the system after a specific simulation time. It is observed that the MPC controller outperforms the LQR controller by successfully compensating the error between disturbed signal and the reference where the LQR controller fails to do so. Disturbance rejection performances of two controllers are also compared with the calculation of root mean square error values between disturbed and reference signals. It is found that the root mean square error value for LQR controller is 0.01 whereas this value is 0.0029 for MPC controller, which is significantly lower than LQR controller.

Based on the outcomes of this thesis, further recommendations for future research are proposed. As an ongoing study, more advanced satellite attitude control methods can be utilized such as Linear Quadratic Integrator and Linear Quadratic Gaussian controllers. These advanced control methods offer improved performance since they also account the integral action on control, system uncertainties, noise and so on. Additionally, nonlinear control methods such as sliding mode control and adaptive control can provide robustness to the system as they account for nonlinearities and uncertainties in the system. Development of estimators such as Kalman Filter and Particle Filter can also be utilized to model the measurement noise and including this noise into the system model to enhance the control performance of the system under the presence of disturbance. Model-based design techniques such as processor-in-the-loop and hardware-in-the-loop simulations may assist to improve the system's control performance as they allow for the evaluation of control algorithms in real-time. Implementation of proposed control algorithms into the embedded hardware will also enable the testing of hardware and software constraints, as well as actual implementation on real satellites.

**OPTİMAL UYDU YÖNELİM KONTROL SİSTEMİ TASARIM, ANALİZ VE GELİŞTİRİLMESİ**

## ÖZET

Günümüzde Dünya yörüngesinde dolaşan on binlerce aktif, operasyonel uydu bulunmaktadır. Bu uydular keşif, seyrüsefer, iletişim gibi birçok amaçla kullanılmaktadır. Bir uydunun en kritik alt bileşenlerinden birisi de uydunun yönelim belirleme ve kontrol sistemidir. Bu alt sistem, uzayda uydunun yönünü tayin etmesine imkân kılmanın yanısıra, bir yönden diğer bir yöne dönmesini gerçekleştirir. Bu tezde bir uydu yönelim kontrolü sistemi tasarlanmış, analizi ve hesaplamaları yapılmıştır. Özellikle kontrolcüler seviyesinde, Doğrusal Karesel Regülatör (LQR) ve Model Öngörülü Kontrol (MPC) üzerine çalışılmıştır. İki kontrolcünün bozucu red performansları karşılaştırılmış ve analizleri yapılmıştır.

Başlangıç bölümünde matematiksel modeli kurmak için gerekli tanımlamalar ve kavramlar incelenmiştir. İlk olarak referans sistemleri tanıtılmıştır. Bu başlık altında, öncelikle "ataletsel" olma kavramı incelenmiş ve ilk olarak Yer Merkezli-Yer Sabit (ECEF) koordinat sistemi tanıtılmıştır. Bu koordinat sisteminin x-y ve z bileşenleri verilmiştir. Sonrasında sırasıyla Yer Merkezli Ataletsel (ECI), Yerel Düşey-Yerel Yatay (LVLH) ve Cisim-Sabit koordinat sistemleri tanıtılmış, x-y ve z eksenleri verilmiştir. Koordinat sistemlerinin anlaşılmasının, ilgili dönüşümleri yapmak için önemli olduğu belirtilmiştir. Bunun sonrasında, uydu açısal gösteriminin iki türü olan Euler açıları ve Kuaterniyonlar tanıtılmıştır. Kuaternion ve Euler açıları arasındaki dönüşüm formülleri verilmiştir. Bu bölümün sonunda, vektör çarpımını matris çarpımı biçiminde ifade edilebilecek hale dönüştüren anti-simetrik matrisler tanıtılmıştır.

Temel kavramlar ve tanımların açıklanması sonrası uydu hareket denklemleri tanıtılmıştır. Hareket denklemleri, dinamik ve kinematik alt başlıkları ile incelenmiştir. Uydu dinamik denklemleri, ataletsel koordinat sisteminde verilmiştir. Dinamik denklemi, girdi olarak uyduya etkiyen kuvvetleri alıp, çıktı olarak uydunun açısal hız vektörünü üretmeye yaramaktadır. Dinamik denklemler sonrasında kinematik denklemler verilmiştir. Kinematik denklemler hem Euler, hem de kuaterniyon formunda incelenmiştir. Kontrolcünün tasarlandığı kısımda kuaternion formunda verilen kinematik denklemi kullanılmıştır. İlk bölümde tanıtılan anti-simetrik matrisler kinematik ve dinamik denklemlerinde vektör çarpımını ifade etmek için kullanılmıştır. Kinematik ve dinamik denklemlerin yazımı sonrası uyduya etkiyen bozucular, yerçekimi gradyanı ve aerodinamik sürükleme kuvveti altında iki ana başlık olarak incelenmiştir. Yerçekim gradyanı, uydunun her yönden simetrik olmaması dolayısıyla, uyduya farklı büyüklüklerde etkiyen yerçekimi kuvveti sonucu oluşur. Bu durum, kütle merkezi etrafında bir bozucu tork oluşturur. Bu torkun denklemleri incelenmiştir.

İkinci olarak, Düşük Dünya Yörüngesi'nde hareket eden uydular atmosferin üst kısımlarında kaldıklarından, üzerlerine etki eden hava dolayısıyla sürtünmeye maruz kalırlar. Bu aerodinamik sürtünme, sürükleme kuvvet olarak adlandırılır ve bir diğer bozucu etkidir. Bu sürükleme kuvvetinin modellenmesi için ilgili denklemler verilmiştir. Dinamik, kinematik ve bozucu kuvvet denklemleri verilmesinin ardından, doğrusal olmayan durum uzayı denkemleri oluşturulmuş, uydunun açısal hızı ve kuaterniyon elemanları durum vektörü olarak seçilmiştir.

Oluşturulan doğrusal olmayan matematik modelinin doğrulaması için bir benzetim düzeneği kurulmuştur. Bu benzetim sisteminde, uydunun dinamik denklemlerinde girdi olarak, x ekseni etrafında 1 Nm büyüklüğünde bir tork uygulanmıştır. Bu torkun sonucunda kinematik denklemlerin çıktısına bakarak, uydunun beklenildiği gibi x ekseni etrafında döndüğü görülmüştür. Bu girdi torku, sabit olarak verilmeye devam ettiğinden uydunun açısal hızının gittikçe arttığı gözlemlenmiştir; bu da beklenilen bir sonuçtur ve model doğrulanmıştır. Aynı zamanda, uydunun doğrusal olmayan durum uzayı hareket denklemlerinin, durum eleman sayısının azaltılabildiği gösterilmiştir. Kuaterniyon elemanlarından skaler olan eleman kaldırıldığında, uydunun kontrol edilebilir olduğundan bahsedilmiştir. Bunun sonucunda doğrusal olmayan indirgenmiş durum uzayı denklemi elde edilmiştir.

Elde edilen doğrusal olmayan durum uzayı denklemleri sonrası, kontrol sistemi tasarlamak ve analizi amacıyla bu denklemlerin doğrusallaştırılmasına karar verilmiştir. Denklemlerin doğrusallaştırması için kullanılan yöntemlerden biri olan Taylor serisi açılımı anlatılmıştır. Taylor serisi açılımı ile elde edilen Jacobi matrisinin elemanları türev içermektedir. Jacobi matrisindeki bu türev elemanlarının analitik ve sayısal olarak hesaplamaları açıklanmıştır. Doğrusallaştırılmış model analitik olarak elde edilmiş, aynı zamanda türevlerin sayısal olarak elde edilmesi için bir yöntem olan Merkezi Farklar Yöntemi tanıtılmıştır. Son olarak doğrusal olmayan ve doğrusallaştırılmış modellerin karşılaştırılması yapılmıştır. Bunun testi için uyduya girdi torku olarak bir beyaz gürültü verilmiş, iki modelin de çıktısı incelenmiştir. Bu inceleme sonucu doğrusal olmayan ve doğrusallaştırılmış modellerin çıktılarının yaklaşık 40 derece dönme açısına kadar neredeyse eşdeğer olduğu gözlemlenmiş, böylelikle doğrusallaştırılmış modelin doğrulaması yapılmıştır. Bu bölümün sonunda uydu hareket denklemleri, 6 elemanlı bir durum vektörü içeren doğrusallaştırılmış durum uzayı sistemi şeklinde elde edilmiştir.

Elde edilen doğrusal durum uzayı hareket denklemleri kullanılarak kapalı çevrim bir kontrol sistemi tasarlanmıştır. Bu tasarımda iki kontrolcü, LQR ve MPC incelenmiştir. LQR kontrolcü temelde bir maliyet fonksiyonunun minimize edilmesi üzerine kurulmuş bir optimal kontrolcü türüdür. Bu kontrolcüde maliyet fonksiyonu, durum vektörü ve kontrol girdi vektörleri kullanılarak oluşturulmaktadır. Durum vektörü ve kontrol girdi vektörlerinin, sırasıyla Q ve R parametreleri kullanılarak maliyet fonksiyonundaki ağırlıkları ayarlanabilir. Bu maliyet fonksiyonunu minimize eden bir durum uzayı geri bildirim kontrolcüsü tasarlanmıştır. Bu kontrolcünün kazanç parametresi K ile ifade edilmekte olup, bu parametre Ricatti diferansiyel denkleminin çözülmesi ile elde edilmektedir. LQR kontrolcü için gerçek uydu verilerinin kullanıldığı bir benzetim tasarlanmıştır. Bu benzetimde elde edilen kazanç parametresi incelenerek sistemin kararlı olduğu ve verilen kuaterniyon referanslarını takip ettiği görülmüştür. Buna ek olarak LQR kontrolcünün Q ve R parametrelerinin sistem performansına etkisi incelenmiştir. Bu bağlamda, Q ve R parametrelerinin

değerleri değiştirilerek sistem çıktısı değerlendirilmiştir. Q ve R parametresinin 1 olduğu durumun uygun parametreler olarak seçilmesi kararlaştırılmıştır. Diğer durumlarda sistem çıktısında kalıcı hal hatası gözlemlenmiştir. Kontrol girdisini belirli değerler arasında kısıtlamak amacıyla, sisteme bir "satürasyon" bloğu konmuştur kontrolcü çıktısı, $\pm$ 0.02 Nm ile sınırlandırılmıştır. Satürasyon olduğu durumlarda sitem basamak yanıtının değişimi incelenmiş ve oturma süresinin uzadığı görülmüştür. Bu satürasyon, doğrusal olmayan bir etken olduğundan LQR çözümünde bulunmamaktadır ve optimal çözüm değildir.

İkinci kontrol algoritması olarak Model Öngörülü Kontrol kullanılmıştır. MPC, LQR ile aynı temele dayansa da belirgin farkları bulunmaktadır. MPC kontrolcü, sistem dinamiğinin kullanıldığı bir "öngörü" modeline bağlı olarak çalışır. Durum uzayı sistemi denklemleri kullanılarak öngörü modelinin nasıl elde edildiği açıklanmıştır. MPC'nin LQR'da olduğu gibi benzer bir maliyet fonksiyonu kullandığı gösterilmiştir. MPC'nin en ayırt edici özelliği, optimizasyon problemine kısıtlama yerleştirme olanağıdır. Bu kısıtlamalar sayesinde, sistemin durum vektörü, kontrol girdisi ve sistem çıktısı gibi parametreler belirli değer aralıklarında kısıtlanabilir ve bu şekilde optimizasyon çözümü yapılabilir. Sistem, kısıtlar ve maliyet fonksiyonu ile çözülmeye çalışıldığında, bir Karesel Programlama Problemi çözümü yapılmakta olduğundan bahsedilmiştir. Bu Karesel Programlama Problemi, sayısal olarak çözülerek optimal kontrol girdisi elde edilmektedir. LQR'dan farklı olarak, MPC kontrolcüde oluşturulan öngörü modeli ve optimizasyon çözümü, sistem çıktısına göre her zaman adımında güncellenmektedir. MPC kontrolcü için de gerçek bir uydudan alınan atalet momenti verisinin yanı sıra LQR'da olduğu gibi kontrol sinyali $\pm$ 0.02 Nm aralığında kısıtlanmış, yapılan bir benzetim ile MPC kullanıldığı durumda sistemin kuaterniyon elemanlarını doğru şekilde takip ettiği görülmüştür. Aynı zamanda kontrol sinyali de paylaşılmış ve kısıtlamanın başarıyla gerçekleştiği gösterilmiştir.

LQR ve MPC kontrolcülerin karşılaştırılması için bir benzetim yapılmıştır. Bu benzetimde, belirli bir saniyeden sonra uyduya bir basamak bozucu sinyal verilmiştir, farklı kontrolcülerin kuaterniyon çıktıları incelenmiştir. Çıktılar incelendiğinde, MPC kontrolcünün verilen bozucu torkunu başarıyla sönümlediği ve sistemin tekrar referansa ulaştığı görülmüştür. Öte yandan LQR kontrolcünün, verilen sabit bozucu tork karşısında bu bozucuyu sönümleyemediği görülmüştür. Sayısal olarak da kontrolcü performanslarının karşılaştırılması amacıyla kök ortalama kare hataları hesaplanmıştır. Bu hata değeri, her iki kontrolcü için de bozucu uygulandıktan sonra sistem çıktısı ve referans sinyali arasında hesaplanmış; LQR için 0.01, MPC için 0.0029 değerine ulaşılmıştır. Bu noktada MPC kontrolcünün, bozucu varlığında LQR'a kıyasla gösterdiği performansın çok daha üstün olduğu ortadadır.

Tez çalışmasının çıktılarının kapsamlı şekilde genişletilmesi için öneriler verilmiştir. Bunlar arasında Doğrusal Karesel İntegrator (LQI), Doğrusal Karesel Gaussian (LQG) gibi gelişmiş kontrolcü tasarımları kullanmak bulunmaktadır. Bunun yanı sıra, uyarlanabilir kontrol, kayan kipli kontrol gibi doğrusal olmayan kontrol tasarımları da tasarlanabilir. Böylelikle sistemde integratör kullanımı, ölçüm gürültüsü ve belirsizliklerin eklenmesi mümkün olacaktır. Kalman Filtresi, Parçacık Filtresi gibi kestiriciler de sistemin ölçüm gürültüsünün modellenmesi ve çıktıların kestirimi amacıyla kullanılabilir. Böylelikle bozucuların var olduğu durumda da gelişmiş, gürbüz kontrolcüler tasarlanabilir. Bir diğer adım olarak, işlemci çevrim benzetimi (SIL) donanım çevrim benzetimi (HIL) gibi adımlar kullanılarak bir model tabanlı

kontrol yaklaşımı izlenebilir. HIL düzenekleri tasarlanarak sistemin kontrolcüsünün gömülü bir donanımda çalışması sağlanıp, gerçek sistemdeki kontrolcü performansı incelebilir.

# 1. INTRODUCTION

Satellites are used for a variety of purposes, such as communication, observation (both Earth and space), navigation and so on. One of the main and most crucial subsystems of a satellite is its attitude determination and control system (ADCS). ADCS of a satellite enables it to orient from one attitude to another one as well as stabilize around an orientation. Engineers may design an ADCS that has an enormous pointing accuracy, 7 arc-seconds for Webb Space Telescope for instance [1]. To ensure these high pointing accuracy requirements, advanced sensors and control techniques are utilized for attitude control systems.

The first satellites are launched in the early 1950s. The first satellites were primitive and did not require any attitude control system. However, with the growth of complex missions and satellites, advanced attitude control became apparent. Attitude control is a complex problem since there are many factors that may affect the orientation of the satellite such as gravity [2], solar radiation [3], atmospheric drag [4] and satellite's flexible body [5]. To overcome attitude control problems, different actuator types are used such as reaction wheels, magnetic torquers, control moment gyros and jet thrusters.

One of the main actuators that are used in attitude control systems is reaction wheels. To execute orientation change, one needs to change the angular velocity of the reaction wheel resulting in angular acceleration. This angular acceleration may also be interpreted as torque around the center of mass. By the conservation of the angular momentum principle, the satellite starts to rotate around the opposite direction of produced torque. Another main actuator used in attitude control systems is magnetic torquers, also called magnetorquers. One can generate an artificial magnetic field inside of the satellite by giving voltage to the coils of magnetorquers, this magnetic field interacts with Earth's magnetic field so that torque will be generated as a result of

this magnetic interaction. The actuator choice might depend on many factors such as the desired level of performance, cost of the mission and size/mass of the satellite.

## 1.1 Purpose of Thesis

The main purpose of this thesis is to design and analyze a satellite attitude control system by using different optimal control techniques. The main strategy is to design controllers that can compensate the error between desired attitude and the actual attitude of the satellite. As the controller, Linear Quadratic Regulator (LQR) and Model Predictive Controller (MPC) methods are explained in detail. Tuning of controller parameters and comparison of controllers in terms of disturbance rejection performances are performed.

## 1.2 Literature Review

Various studies have been performed on satellite attitude control with advanced methods, LQR and MPC in particular. LQR is a control method based on the solution of an optimization problem where the cost function to be minimized consists of states, inputs, and their weights. In [6], Ofodile et. al. design an attitude controller for a 3U CubeSat, "ESTCube2/3". They present the nonlinear satellite model and linearize it. They propose different control algorithms, in which one of them is an LQR controller. They compare different control algorithms. Navabi and Hosseini design a quaternion-based attitude controller for a satellite that is actuated by reaction wheels. After constructing nonlinear spacecraft dynamics, they use input-output feedback linearization to obtain the linear spacecraft model and they apply the LQR controller for attitude control purposes [7]. In [8], Fakoor et. al. focus on designing sliding mode and LQR controllers for a 4 degree-of-freedom flexible satellite. Tuning of parameters is performed by the Particle Swarm Optimization method. They use normalized integral square error as the cost function of the particle swarm optimization algorithm. Yang designs an LQR controller for a nonlinear spacecraft model. The author develops a linearized spacecraft model by using three components of the quaternion, which makes the system fully controllable. After that, the author develops an LQR design and utilizes this design on a nonlinear spacecraft model [9].

The model predictive controller, on the other hand, is alike the LQR controller in terms of being the solution to an optimization problem. However, in MPC, there exists a model that predicts ongoing states and inputs/outputs of the system. The control move is updated at every state according to the prediction model and system measurements. Control input is determined with respect to the minimization of the cost function that is similar to the cost function used in the LQR controller. Various studies are presented on MPC-based attitude control of the satellites. In [10], Mamarasulov et. al. construct a linear time-variant CubeSat nanosatellite model and design an MPC controller to control the attitude of this spacecraft. They use Taylor expansion linearization to obtain the linearized time-variant state space model of the satellite. The authors also elaborate on the usage of the Kalman filter for sensor fusion purposes. At the end of the paper, they compare the MPC controller with the LQR controller. Wood et. al. design a magnetorquer-based attitude controller for a satellite. They use an MPC controller to control the attitude of the satellite and compare their results with a well-known controller type: PD controller [11]. Mirshams and Khosrojerdi propose a method called "tube-based MPC" to stabilize the attitude of an under-actuated satellite in 3-axis [12]. More studies on MPC-based satellite attitude control can be found in [13, 14].

### 1.3 Outline of Thesis

Outline of the thesis is as follows: In the second chapter, definitions required to construct the mathematical model of the satellite are presented. Reference frames, Euler angles, and quaternions are presented. The third chapter consists of the construction of the satellite mathematical model under the dynamics and kinematics subsections. Disturbances acting on an orbiting satellite are also modeled. Additionally, a nonlinear state space model is presented and linearization of this nonlinear state space model is performed. To investigate the performance of linearization, comparison between nonlinear and linear models are executed. At the end of this chapter, satellite linear state space is obtained. The fourth chapter comprises the elaboration of controller designs. LQR and MPC controllers are explained in detail. Tuning of controller parameters is examined and two controller methods are compared.

The last chapter concludes the study with significant outcomes and proposes studies that may be investigated for ongoing research.

# 2. CONCEPTUAL BACKGROUND

## 2.1 Reference Frames

In this section, the reference frames that are used to explain the motion of a satellite are introduced. Firstly, it is essential to understand the concepts of "inertial" and "non-inertial" reference frames.

An inertial reference frame is a frame that has no acceleration. This is valid for both linear and angular accelerations. On the other hand, a non-inertial reference frame is a frame that has acceleration. One can apply Newton's mechanical laws for an inertial frame whereas Newton's laws are not applicable for a non-inertial reference frame [15]. For instance, if one writes equations of motion for a rotating frame (which means existence of angular acceleration) there will exist an extra force called "Coriolis force" which needs to be implemented into the equations of motion.

### 2.1.1 Earth-centered Earth-fixed (ECEF) reference frame

In the ECEF coordinate system, the origin is at the center of the Earth. z-axis points towards the north pole. x-axis points towards the intersection of (0,0) latitude and longitude. The y-axis is completed by the right-hand rule [16]. This reference frame rotates with the Earth. Thus, it is a non-inertial reference frame. [17]

### 2.1.2 Earth-centered inertial (ECI) reference frame

ECI reference frame has the origin at the center of the earth. z-axis points towards the north pole. x-axis points towards the vernal-equinox plane. The y-axis is completed by the right-hand rule [18]. ECI is an inertial frame. Thus, it is fixed in the universe and does not rotate.

### 2.1.3 Local-vertical local-horizontal reference frame (LVLH)

This frame is one of the main reference frames explaining a satellite's orientation. In the LVLH, the origin is at the center of mass of the satellite. z-axis points towards the center of mass of the Earth. The y-axis is perpendicular to the orbital frame and The x-axis completes the right-hand rule.

### 2.1.4 Body-fixed reference frame

The body-fixed reference frame is alike the LVLH frame. The origin is at the center of the mass of the satellite. The is no standard representation of the axis for this frame. It is common to choose the principle axis of inertia as the frame axes [19]. Visual representations of ECI, LVLH and body-fixed frames are given in Fig. 2.1.

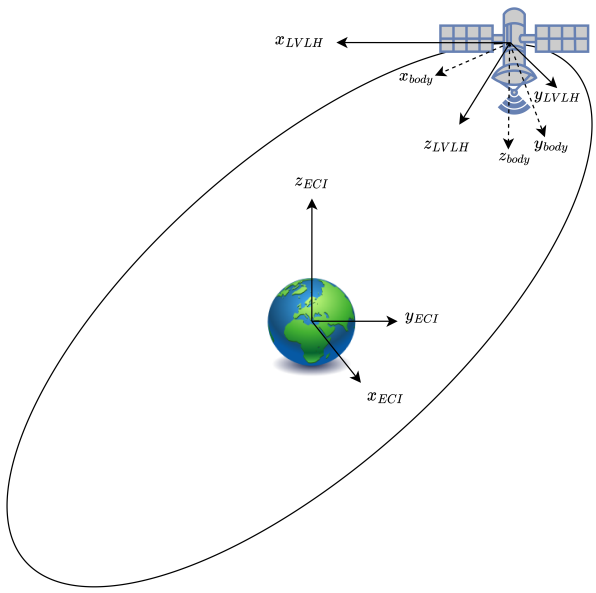


**Figure 2.1 :** Visualization of ECI, LVLH and body-fixed frames.

## 2.2 Euler Angles

The attitude of a satellite can be represented as the angle of the body-fixed frame with respect to an inertial frame. Let us rotate the body-fixed frame around an inertial frame. This rotation can be represented in different sequences. For instance, one can rotate the frame around x-y-z, respectively to reach the current angular state of the satellite. It

is also possible to rotate with different sequences as well. When the rotation sequence is z-y-x (also called 3-2-1), it is called as the "Euler angle". The rotation around z-y-x are called "**Yaw** ($\psi$)", "**Pitch** ($\theta$)" and "**Roll** ($\phi$)" angles, respectively. Let reference coordinate system be represented by $\gamma$ and target body-fixed coordinate system is $\gamma''$ Visualization of rotation sequence for coordinate transformations are given in Fig. 2.2.

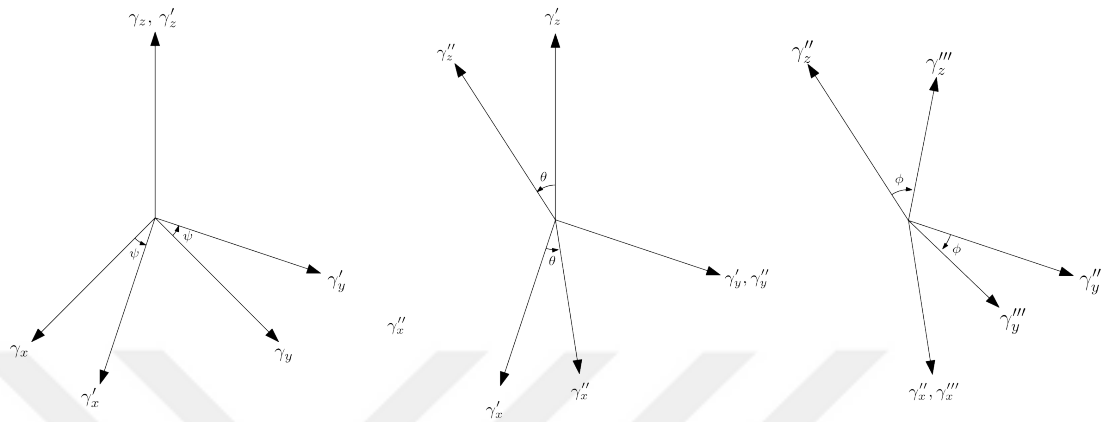


**Figure 2.2 :** Euler angle coordinate transformation sequence.

As shown in Fig 2.2, the sequence starts with a $\psi$ deg rotation around the z-axis, which corresponds to a rotation matrix:

$$A_3 = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

where $c$ and $s$ stand for cosine and sine, respectively. The second rotation consists of rotation around the y-axis of the new frame, $\gamma'$:

$$A_2 = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \tag{2.2}$$

The rotation matrix around the x-axis of the new frame, $\gamma''$ is:

$$A_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \tag{2.3}$$

The final transformation matrix that consists of the Euler angle rotation sequence is:

$$A_{DCM} = A_1 A_2 A_3 = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\phi s\psi + s\phi s\theta c\psi & c\phi c\psi + s\phi s\theta s\psi & s\phi c\theta \\ s\phi s\psi + c\phi s\theta c\psi & -s\phi c\psi + c\phi s\theta s\psi & c\phi c\theta \end{bmatrix} \tag{2.4}$$

The matrix $A_{DCM}$ is also called the "Direction Cosine Matrix (DCM)" which is also another representation to describe the satellite attitude. There exist different DCMs according to different sequences of rotation. It is possible to obtain the Euler angles from the DCM $A$ by using basic relations that can be extracted from Eq. 2.4 as:

$$\begin{aligned} \phi &= tan^{-1}\left(\frac{A_{23}}{A_{33}}\right) \\ \theta &= -sin^{-1}(A_{13}) \\ \psi &= tan^{-1}\left(\frac{A_{12}}{A_{11}}\right) \end{aligned} \tag{2.5}$$

One of the main disadvantages of using the Euler angle is there happens singularity when $\theta = \pm 90$ deg. which needs to be avoided while representing the attitude.

## 2.3 Quaternions

Quaternions are the other options to represent the attitude of a satellite. Let us construct a unit sphere in Fig. 2.3 and specify two points $p_1$ and $p_2$ represented as vectors $\vec{p}_1$ and $\vec{p}_2$ on the surface of the sphere. Let $\vec{v}$ be a vector on the sphere surface perpendicular to $\vec{p}_1$ and $\vec{p}_2$, namely their cross product. It can be said that rotation from $\vec{p}_1$ to $\vec{p}_2$ can be achieved by the rotation around the vector $\vec{v}$ with an angle $\alpha$. This rotation can be represented by using quaternions.

Quaternion is a number system that was first described by William Rowan Hamilton [20]. Quaternions have 3 vectoral and one scalar element just like given in Fig. 2.3 and can be represented as:

$$q = \cos\left(\frac{\alpha}{2}\right) + \sin\left(\frac{\alpha}{2}\right)\vec{v} \tag{2.6}$$

Which can be also described as scalar and vectoral parts. Let us arrange Eq. 2.6 as:

$$q = q_0 + q_1 i + q_2 j + q_3 k \tag{2.7}$$

where the element $q_0$ is the scalar element; $q_1$, $q_2$ and $q_3$ are the vectoral elements represented as imaginary numbers.

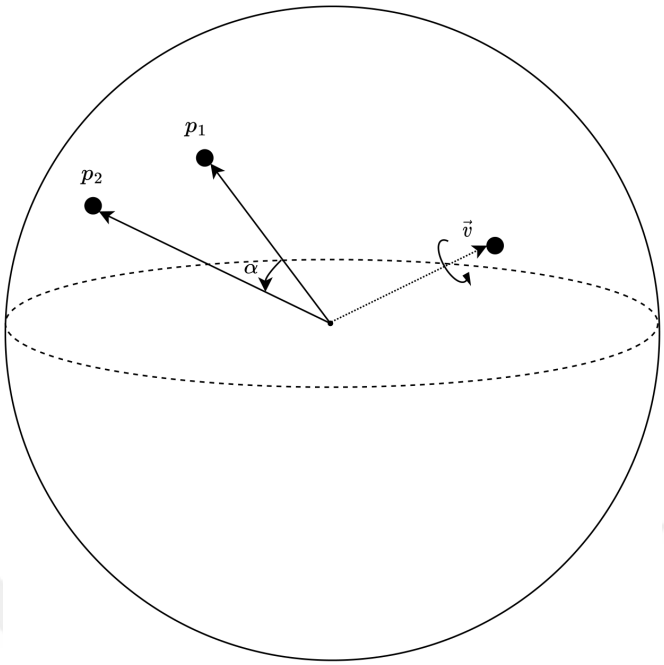


**Figure 2.3 :** Visual representation of quaternions.

Some definitions related to quaternions need to be given. Adjoint ($\bar{q}$), norm ($||q||$) and inverse ($q^{-1}$) of a quaternion can be calculated as [21]:

$$\bar{q} = [q_0 - q_1 - q_2 - q_3] \tag{2.8}$$

$$||q|| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{2.9}$$

$$q^{-1} = \frac{\bar{q}}{||q||} \tag{2.10}$$

Additionally,

$$i^2 = j^2 = k^2 = ijk = -1 \tag{2.11}$$

$$
\begin{aligned}
ij &= k \\
jk &= i \\
ki &= j
\end{aligned}
\tag{2.12}
$$

## 2.4 Conversion Between Euler Angles and Quaternions

It is possible to convert Euler angles to quaternions and vice versa. The proper transformation from Euler angles to quaternions can be written as [22]:

$$
\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) \end{bmatrix}
\tag{2.13}
$$

Similarly, conversion from Euler angles to quaternions can be represented as:

$$
\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{2q_0q_1+2q_2q_3}{q_0^2-q_1^2-q_2^2+q_3^2}\right) \\ \sin^{-1}\left(2q_0q_2 - 2q_3q_1\right) \\ \tan^{-1}\left(\frac{2q_0q_3+2q_1q_2}{q_0^2+q_1^2-q_2^2-q_3^2}\right) \end{bmatrix}
\tag{2.14}
$$

## 2.5 Skew-Symmetic Matrix

A matrix is said to be skew-symmetric if its negative transpose is equal to itself, namely:

$$
A = -A^T
\tag{2.15}
$$

Let us define two vectors $M$ and $N$ with basis vectors $\hat{e}_1$, $\hat{e}_2$ and $\hat{e}_3$:

$$
\begin{aligned}
M &= M_1\hat{e}_1 + M_2\hat{e}_2 + M_3\hat{e}_3 \\
N &= N_1\hat{e}_1 + N_2\hat{e}_2 + N_3\hat{e}_3
\end{aligned}
\tag{2.16}
$$

Cross-product of two vectors can be written as:

$$
\begin{aligned}
M \times N &= \begin{vmatrix} \hat{e}_1 & \hat{e}_2 & \hat{e}_3 \\ M_1 & M_2 & M_3 \\ N_1 & N_2 & N_3 \end{vmatrix} \\
&= \hat{e}_1 \left(M_2 N_3 - M_3 N_2\right) - \hat{e}_2 \left(M_1 N_3 - M_3 N_1\right) + \hat{e}_3 \left(M_1 N_2 - M_2 N_1\right)
\end{aligned}
\tag{2.17}
$$

which can be represented in $\hat{e}$ basis as:

$$
M \times N = \begin{bmatrix} M_2 N_3 - M_3 N_2 \\ M_3 N_1 - M_1 N_3 \\ M_1 N_2 - M_2 N_1 \end{bmatrix} = \begin{bmatrix} 0 & -M_3 & M_2 \\ M_3 & 0 & -M_1 \\ -M_2 & M_1 & 0 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} \tag{2.18}
$$

Equation 2.18 can be arranged as:

$$
M \times N = M_{\times} N \tag{2.19}
$$

where,

$$
M_{\times} = \begin{bmatrix} 0 & -M_3 & M_2 \\ M_3 & 0 & -M_1 \\ -M_2 & M_1 & 0 \end{bmatrix} \tag{2.20}
$$

Thus, it can be said that the cross-product operator can be represented as a matrix as given in Eq. 2.20. Notice that this matrix satisfies the condition given in Eq. 2.15, which is the condition for being a skew-symmetric matrix. Thus, the matrix $M_{\times}$ is called a skew-symmetric cross-product matrix.

# 3. SATELLITE EQUATIONS OF MOTION

Satellite equations of motion can be examined under the titles: kinematics, dynamics and actuator and disturbance modeling. In Fig. 3.1, open loop diagram of a satellite attitude system is given which clarifies the input-output relation and subsystems between them.

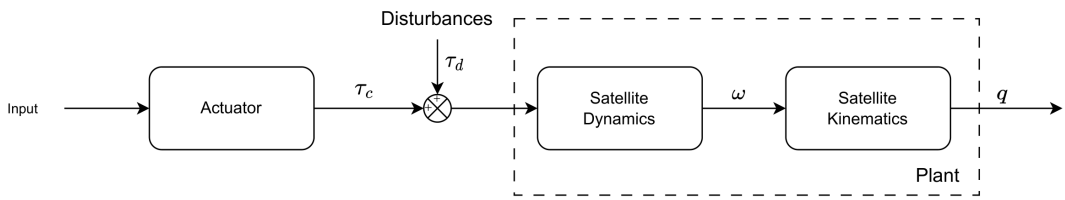


**Figure 3.1 :** Open loop block diagram.

As the input, voltage, or Pulse Width Modulation (PWM) signal (for the DC motor of reaction wheels), is given into the actuator subsystem that will produce the command for generated control torque in which it can be represented as $\tau_c$ the actuated system.

The disturbance torque also needs to be included in the system dynamics since there occurs disturbances during the operation of satellites such as aerodynamic force (in low orbit) gravity gradient, solar pressure and so on.

The total force acting on a satellite can be calculated by adding control torque and disturbance torques. Total torque will be included as the input for the satellite dynamics subsystem.

The satellite dynamics subsystem will produce the corresponding angular velocity with respect to time for given input torques acting on the satellite.

The angular velocity state will be used as input for the kinematics subsystem to produce the final angular state of the satellite. This final state can be obtained by different representations such as Euler angles, quaternions and so on.

The overall system can be interpreted as to be having the voltage as the input and angular state as the output. One can design an open-loop simulator by constructing all the equations/subsystems given in this section.

### 3.1 Satellite Dynamics

Dynamics is the part that deals with the force acting on the satellite. One can yield the angular velocity state of the satellite by taking the overall force acting on the satellite as the input. The force acting on a satellite can be examined as internal force and external forces (disturbance).

The satellite dynamics equation is basically based on Newton's second law for angular motion, which can be described as: "the total external moment acting on a satellite around its center of mass is equal to the time derivative of the angular momentum" This can be formulated in inertial frame as [23]:

$$T_e = \dot{h_I} \tag{3.1}$$

where $T_e$ is the total external moment acting on the satellite and $h_I$ is the angular momentum represented in the inertial frame, where the subscript $I$ stands for the word "inertial".

To examine the motion in the body frame, one needs to apply proper transformations since it is known that the body frame is fixed to the satellite. Thus, it is a non-inertial frame (See. Section 2.1). The transformation from inertial frame to the body-fixed can be applied by adding a new term as explained in [23], which converts Eq. 3.1 as to be:

$$T_e = \dot{h}_s + \omega \times h_s \tag{3.2}$$

where $h_B$ is the angular momentum vector of the satellite expressed in the body-fixed frame and $\omega$ is the angular velocity vector of the satellite.

It is also known that the external torque can be composed of disturbance torque $\tau_d$ and control torque $\tau_c$. Additionally, the angular momentum of a body can be expressed as the multiplication of the moment of inertia times angular velocity. For a satellite, this can be defined as:

$$h_s = I_s \omega \tag{3.3}$$

where $I_s$ is the moment of inertia matrix of the satellite.

Applying these conditions to Eq. 3.2 yields:

$$\tau_d + \tau_c = I_s \dot{\omega} + \omega \times I_s \omega \tag{3.4}$$

Leaving the angular velocity term alone in Eq. 3.4 yields the dynamics model for an inertial pointing spacecraft [24]:

$$\dot{\omega} = I_s^{-1} \left( -\omega \times I_s \omega + \tau_d + \tau_c \right) \tag{3.5}$$

Equation 3.5 represents the final version of the satellite dynamics equations. In this equation, the input is control torque and disturbance torque whereas the output is satellite angular velocity. Thus, one can obtain the angular velocity from the inputs by only having the knowledge of the satellite moment of inertia matrix, which clarifies the dynamics subsystem given in Fig. 3.1.

### 3.2 Satellite Kinematics

Kinematics deals with motion without considering the effect of force. For the satellite motion, kinematics equations help to convert angular velocity obtained by dynamics equations into convenient angles.

For a satellite, kinematics equations might be written in different ways. In this section, the equations are constructed in the way of Euler angles and quaternions.

Euler angle-based kinematics equations differ for different rotation sequences. Since the base rotation sequence is chosen as 3-2-1 Euler angle sequence for this thesis, the kinematics equations for this sequence can be written as [25]:

$$\begin{aligned} \dot{\phi} &= \omega_1 + (\omega_2 \sin\phi + \omega_3 \cos\phi) \tan\theta \\ \dot{\theta} &= \omega_2 \cos\phi - \omega_3 \sin\phi \\ \dot{\psi} &= (\omega_2 \sin\phi + \omega_3 \cos\phi) sec\theta \end{aligned} \tag{3.6}$$

One can write Eq. 3.6, in matrix form as:

$$\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \frac{1}{\cos\theta} \begin{bmatrix} 0 & \sin\phi & \cos\phi \\ 0 & \cos\phi\sin\theta & -\sin\phi\cos\theta \\ \cos\theta & \sin\phi\sin\theta & \cos\phi\sin\theta \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{3.7}$$

By using Eq. 3.7, it is possible to obtain Euler angles as roll, pitch and yaw by taking the angular velocity vector as the input. However, the biggest disadvantage of using Euler angles to represent kinematics is that the existence of singularity. Notice that the term $1/\cos\theta$ approaches infinity when $\theta = \pm\pi/2$, which destroys the equation. Thus, when using Euler angle-based kinematics, one needs to avoid the pitch angle to be $\pm\pi/2$ radians.

To avoid the singularity of Euler angle kinematics, it is possible to write quaternion-based kinematics. For a satellite, quaternion-based kinematics equation is written as [24]:

$$\dot{q} = \frac{1}{2}\Omega q \tag{3.8}$$

where

$$\Omega = \Omega(\omega) = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \tag{3.9}$$

It is also possible to express Eq. 3.8 as:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{3.10}$$

By using kinematics equations expressed in quaternion form, one can obtain the quaternions by taking the angular velocity as input.

Equation 3.8 can also be separated into two parts as vector and scalar elements of the quaternion. Let $g$ be the vector that represents the vector elements of $q$, namely $\vec{g} = [q_1 q_2 q_3]^T$, which is also known as the Gibbs vector. Equation 3.8 can be written as:

$$\begin{bmatrix} \dot{g} \\ \dot{q}_0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\omega_\times g + q_4 I_{3\times3}\omega \\ -\omega^T g \end{bmatrix} \tag{3.11}$$

where $I_{3\times3}$ is 3×3 identity matrix, $\omega_\times$ represents the skew-symmetric cross product matrix of angular velocity:

$$\omega_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{3.12}$$

## 3.3 Disturbance Modeling

There are various disturbance forces acting on a satellite during its operation. The disturbance types depend on the mission of the satellite such as its shape, altitude onboard elements and so on. Let us examine a satellite that operates at Low Earth Orbit (LEO). The most important disturbances can be examined as aerodynamic and gravity gradient forces at these altitudes.

### 3.3.1 Gravity gradient

Due to the unsymmetrical shape of the satellite, different gravitational force will act on different points of it, which will cause a disturbance torque around the center of mass. This disturbance torque is also known as gravity gradient torque and can be approximated as a linear equation [23]:

$$\begin{aligned} G_x &= \frac{3\mu}{R^3}\left(I_{zz} - I_{yy}\right)\phi \\ G_y &= \frac{3\mu}{R^3}\left(I_{zz} - I_{xx}\right)\theta \\ G_z &= 0 \end{aligned} \tag{3.13}$$

where $G_x$, $G_y$ and $G_z$ are the gravity gradient torque elements for x,y and z directions, $\mu$ is Earth's gravitational parameter, $R$ is the radius for circular orbits, $I_{xx}$, $I_{yy}$ and $I_{zz}$ are the diagonal elements of satellite moment of inertia matrix.

### 3.3.2 Aerodynamic drag

The LEO orbit still consists of air. Satellites orbiting at LEO will be subject to a drag force due to the presence of the atmosphere. This drag force $F_d$ can be calculated by the well known drag equation [26]:

$$F_d = \frac{1}{2}\rho v^2 c_D A_{ref} \tag{3.14}$$

where $\rho$ is the air density at that altitude, $v$ is the linear velocity vector, $c_D$ is the drag coefficient and $A_{ref}$ is the reference area of the satellite that faces incoming air.

## 3.4 Nonlinear State Space Model

One can collect the whole equations of motion explained in previous sections in state space representation. It is possible to introduce the dynamical behavior of a system by defining particular states, inputs and outputs. State space equations for a system can be written as:

$$\dot{x} = F(x,u) \tag{3.15}$$

where $u$ is input and $x$ is the state vector of the system.

Let us define the satellite state vector as to be consisting of satellite angular velocity an quaternion elements:

$$x = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \tag{3.16}$$

As explained in equations of motion in previous sections, the overall nonlinear state space model can be written as in Eq. 3.15:

$$\begin{bmatrix} \dot{\omega} \\ \dot{q}_0 \\ \dot{g} \end{bmatrix} = \begin{bmatrix} I_s^{-1}\left(-\omega_\times I_s \omega + \tau_c + \tau_d\right) \\ -\omega^T g \\ 0.5\left(-\omega_\times g + q_4 I_{3x3}\omega\right) \end{bmatrix} \tag{3.17}$$

In this system, the inputs are control torque ($T_c$) and external torque, $T_e$. Output is the quaternion elements.

### 3.4.1 Model validation

To proceed further on the system, let us validate our model by conducting a set of tests. Nonlinear state space equations were constructed as given in Eq. 3.17. To test the model, let us give an input torque to the system around the x, y and z axis, respectively. The constructed Simulink model is given in Fig. 3.2. Content of MATLAB

Function blocks "SCmodel_7state" and "quat2Euler" are given in Appendices A.1, A.2 respectively.

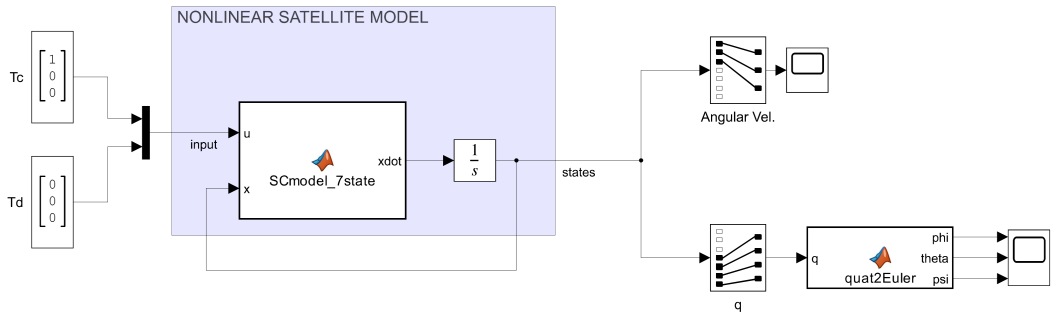


**Figure 3.2 :** Nonlinear state space model constructed in Simulink.

1 Nm of torque around the x-axis was given to the system. The angular position and velocity outputs of the system are given in Figs. 3.3 and 3.4. When a constant input torque is applied to the system, it is expected to get a rotating satellite around the torque-applied axis. This is shown in Fig. 3.3 as expected. Since the torque is given continuously, the rotation rate must increase, which is proven in Fig. 3.4. Thus, the time between two rotation periods must decrease by time, which can be also observed in Fig. 3.3.

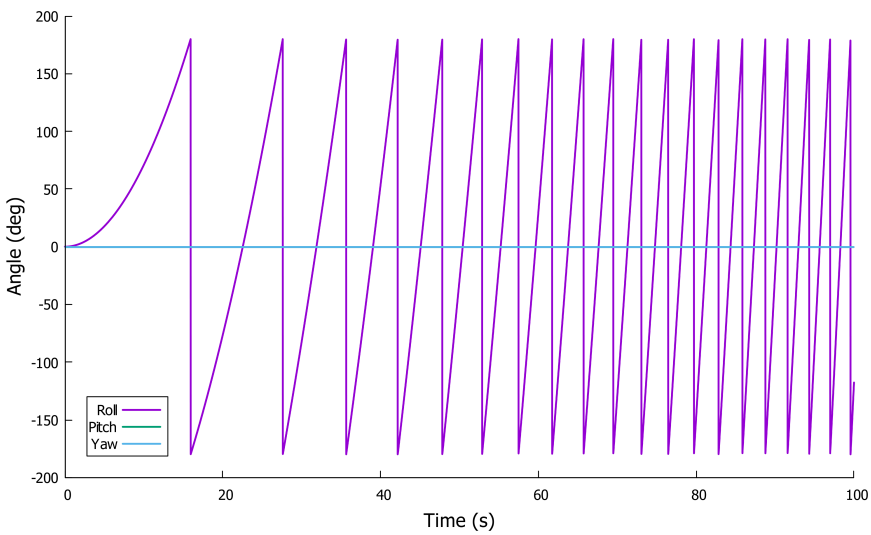


**Figure 3.3 :** Variation of angles for a given test input.

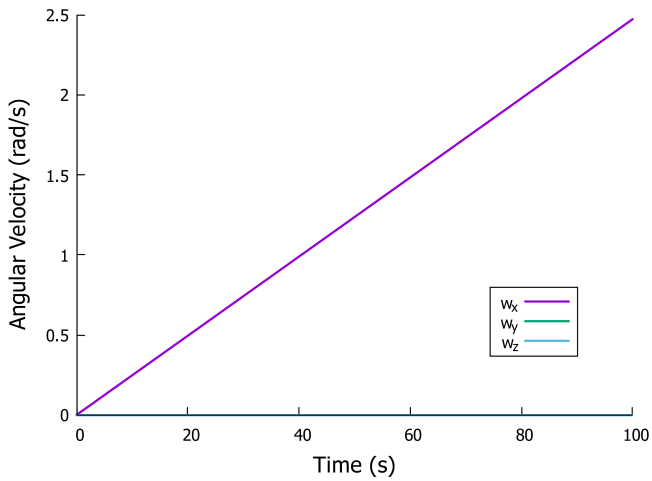


**Figure 3.4 :** Variation of angular velocity for a given test input.

Since the model gives expected results for the given input test, one can say that it is safe to proceed with this nonlinear model.

### 3.4.2 Dimension reduction of the nonlinear state space model

Let us rewrite Eq. 2.9 for normalized quaternion:

$$1 = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{3.18}$$

which shows that the scalar element of a normalized quaternion can be written as a combination of quaternion vector elements:

$$q_0 = \sqrt{1 - q_1^2 - q_2^2 - q_3^2} \tag{3.19}$$

By using Eq. 3.19, one can vanish the state $q_0$ given in Eq. 3.17. All other states that consist of $q_0$ element can be written in terms of vector elements as stated in Eq. 3.19.

Let us rearrange Eq. 3.10 as to be having only vector element states:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{3.20}$$

which can be expressed as:

$$\dot{g} = Q\omega \tag{3.21}$$

Let us investigate the determinant of the matrix $Q$. This can be calculated as:

$$\begin{aligned}\det Q &= q_0\left(q_0^2+q_1^2\right)+q_3\left(q_3q_0-q_1q_2\right)+q_2\left(q_3q_1+q_0q_2\right)\\ &= q_0\left(q_0^2+q_1^2+q_2^2+q_3^2\right)\end{aligned} \tag{3.22}$$

Using relation given in Eq. 3.18 yields:

$$\det Q = q_0 \tag{3.23}$$

According to this result, the determinant of the matrix $Q$ is zero when $q_0 = 0$. This condition is satisfied when $\alpha = \pm\pi$ rad (see Eq. 2.6). Thus reduced quaternion has also singular point (like in the Euler angles) when the angle reaches $\pm$ 180 degrees since it is not full rank in these cases. However, most of attitude control applications stay within the safe area.

In summary, reduced state space equations of motion can be written as:

$$\dot{x} = \begin{bmatrix}\dot{\omega}\\ \dot{g}\end{bmatrix} = \begin{bmatrix} I_s^{-1}\left(-\omega_\times I_s\omega+\tau_c+\tau_d\right)\\ \frac{1}{2}Q\omega\end{bmatrix} \tag{3.24}$$

The number of states is now reduced from 7 to 6. This enables the acceleration of computational power. Furthermore, when the system is linearized, the classical controller methods can be applied to the reduced system since it is now fully controllable and stability analysis can be performed [27].

### 3.5 Linearizaton of Equations of Motion

The equations of motion were explained by using nonlinear equations. However, working on a linear system enables the usage of classical control theory, stability analysis and many more advantages. This section describes the linearization of nonlinear satellite equations.

It is possible to approach a nonlinear system linearly around an operating point. As an example let us investigate the function $f = \sin(x)$. It is well known that around 0 degrees, for small angle approximation, the function $\sin(x)$ can be approximated linearly as $f(x) = x$. The visualization of two functions is given in Fig. 3.5 As shown

in Fig. 3.5 the portion of functions that lay within around $\pm 0.5$ rad (corresponding $\pm 20$ degrees) are nearly identical. Thus, for small angles, one can use $f(x) = x$ to calculate the sine of the angle.

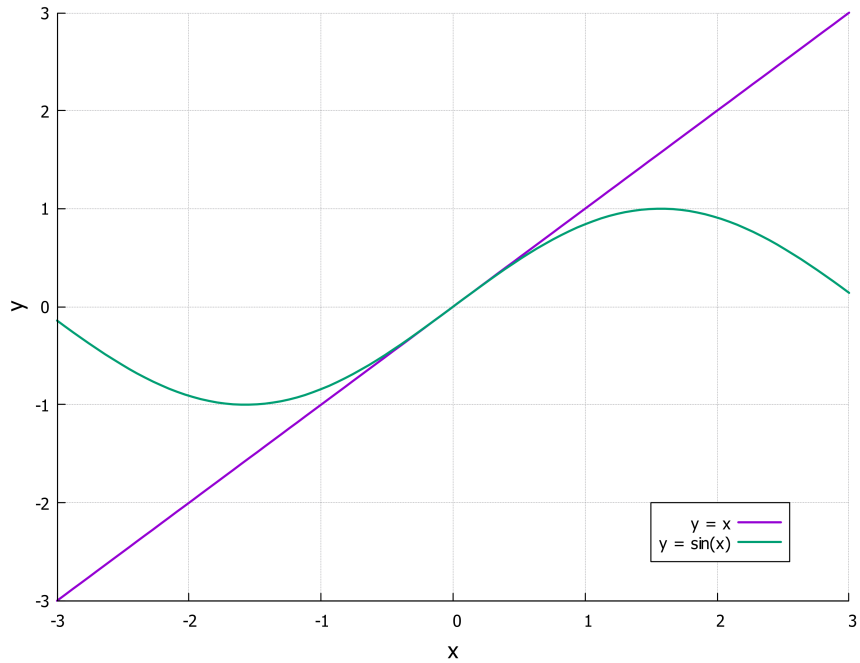


**Figure 3.5 :** Plots of $f(x) = x$ and sine function.

Linearization of a nonlinear function can be achieved by using a well-known method: Taylor Expansion.

Let us express a nonlinear state space system:

$$\dot{x} = f(x,u) \tag{3.25}$$

The Taylor expansion around an operating point, $(x_0, u_0)$ can be written as :

$$f(x,u) \approx f(x_0,u_0) + \left.\frac{\partial f}{\partial x}\right|_{x=x_0}(x-x_0) + \left.\frac{\partial f}{\partial u}\right|_{u=u_0}(u-u_0) + H.O.T. \tag{3.26}$$

where $H.O.T.$ stands for higher order terms. Let us define:

$$
\begin{aligned}
\delta x &\equiv x - x_0 \\
\delta u &\equiv u - u_0 \\
\delta \dot{x} &\equiv f(x,u) - f(x_0,u_0) \\
A &\equiv \left.\frac{\partial f}{\partial x}\right|_{x=x_0} (x - x_0) \\
B &\equiv \left.\frac{\partial f}{\partial u}\right|_{u=u_0} (u - u_0)
\end{aligned}
\tag{3.27}
$$

By ignoring higher order terms, Eq. 3.26 becomes:

$$
\delta \dot{x} = A\delta x + B\delta u \tag{3.28}
$$

The matrices A and B might involve more than 1 row and column when the number of states are more than one, which it is then called the "Jacobian Matrix", which can be defined for states as:

$$
J_A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_q}{\partial x_1} & \frac{\partial f_q}{\partial x_2} & \cdots & \frac{\partial f_q}{\partial x_n} \end{bmatrix} \tag{3.29}
$$

where $n$ is the number of states and $q$ is the number of functions. For input $u$, the Jacobian matrix $J_B$ can be written as:

$$
J_B = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_m} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_q}{\partial u_1} & \frac{\partial f_q}{\partial u_2} & \cdots & \frac{\partial f_q}{\partial u_m} \end{bmatrix} \tag{3.30}
$$

where $m$ is the number of inputs. The derivative term can be calculated analytically for simple equations, whereas it needs to be executed numerically for complex cases.

### 3.5.1 Analytical linearization

Let us rewrite the nonlinear state space equations by ignoring disturbance torques and taking the control torque as input, namely $\tau_c = u$:

$$\dot{x} = \begin{bmatrix} \dot{\omega} \\ \dot{g} \end{bmatrix} = \begin{bmatrix} I_s^{-1}\left(-\omega_\times I_s \omega + u\right) \\ \frac{1}{2} Q \omega \end{bmatrix} \tag{3.31}$$

It is possible to define the functions $f_1$ and $f_2$ to calculate the Jacobian:

$$\begin{aligned} f_1 &\equiv \dot{\omega} \\ f_2 &\equiv \dot{g} \end{aligned} \tag{3.32}$$

The Jacobian matrices for states and inputs can be calculated as:

$$J_A = \begin{bmatrix} \frac{\partial f_1}{\partial \omega} & \frac{\partial f_1}{\partial g} \\ \frac{\partial f_2}{\partial \omega} & \frac{\partial f_2}{\partial g} \end{bmatrix} \tag{3.33}$$

$$J_B = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{bmatrix} \tag{3.34}$$

Let us determine the stationary states $x_0$ and inputs $u_0$ to be linearized as:

$$\begin{aligned} x_0 &= \begin{bmatrix} \omega_{10} & \omega_{20} & \omega_{30} & q_{10} & q_{20} & q_{30} \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \tag{3.35}$$

$$\begin{aligned} u_0 &= \begin{bmatrix} u_{1_0} & u_{2_0} & u_{3_0} \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \tag{3.36}$$

the elements of Jacobian matrices can be calculated as:

$$\left.\frac{\partial f_1}{\partial \omega}\right|_{x=x_0} = \left.\frac{\partial}{\partial \omega} I_s^{-1}\left(-\omega_\times I_s \omega + u\right)\right|_{x=x_0} = 0_{3\times 3} \tag{3.37}$$

$$\left.\frac{\partial f_1}{\partial g}\right|_{x=x_0} = \left.\frac{\partial}{\partial g} I_s^{-1}\left(-\omega_\times I_s \omega + u\right)\right|_{x=x_0} = 0_{3\times 3} \tag{3.38}$$

$$\left.\frac{\partial f_2}{\partial \omega}\right|_{x=x_0} = \left.\frac{\partial}{\partial \omega}\left(\frac{1}{2} Q \omega\right)\right|_{x=x_0} = \left.\frac{1}{2} Q\right|_{x=x_0} \tag{3.39}$$

Referring to Eq. 2.9, when the vector elements of quaternion are zero (given as stationary point), the scalar element of quaternion becomes 1. Thus, the Jacobian element given in Eq. 3.39 becomes:

$$Q\Big|_{x=x_0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \frac{1}{2} I_{3\times 3} \tag{3.40}$$

$$\frac{\partial f_2}{\partial g}\Big|_{x=x_0} = \frac{\partial}{\partial \omega}\left(\frac{1}{2} Q\omega\right)\Big|_{x=x_0} = 0_{3\times 3} \tag{3.41}$$

For the input, Jacobian elements can be written as:

$$\frac{\partial f_1}{\partial u}\Big|_{u=u_0} = \frac{\partial}{\partial u} I_s^{-1}\left(-\omega_\times I_s \omega + u\right)\Big|_{u=u_0} = I_s^{-1} \tag{3.42}$$

$$\frac{\partial f_2}{\partial u}\Big|_{u=u_0} = \frac{\partial}{\partial u}\left(\frac{1}{2} Q\omega\right)\Big|_{u=u_0} = 0_{3\times 3} \tag{3.43}$$

Finally, the linearized state space equations become:

$$\begin{bmatrix} \dot{\omega} \\ \dot{g} \end{bmatrix} = \begin{bmatrix} 0_{3\times 3} & 0_{3\times 3} \\ \frac{1}{2} I_{3\times 3} & 0_{3\times 3} \end{bmatrix} \begin{bmatrix} \omega \\ g \end{bmatrix} + \begin{bmatrix} I_s^{-1} \\ 0_{3\times 3} \end{bmatrix} u \tag{3.44}$$

Let us check the controllability of this final equation.

The controllability matrix for 6 states can be written as:

$$M = \begin{bmatrix} B & AB & A^2B & A^3B & A^4B & A^5B \end{bmatrix} \tag{3.45}$$

The matrix M can easily be constructed via MATLAB symbolic manipulation (Appendix A.3) It is found as:

$$\begin{bmatrix} 1/I_{s_1} & 0 & 0 & 0 & 0 & 0 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0 & 1/I_{s_2} & 0 & 0 & 0 & 0 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0 & 0 & 1/I_{s_3} & 0 & 0 & 0 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0 & 0 & 0 & 1/(2I_{s_1}) & 0 & 0 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0 & 0 & 0 & 0 & 1/(2I_{s_2}) & 0 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0 & 0 & 0 & 0 & 0 & 1/(2I_{s_3}) & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \end{bmatrix} \tag{3.46}$$

where the satellite moment of inertia vector is assumed as:

$$\begin{bmatrix} I_{s_1} & 0 & 0 \\ 0 & I_{s_2} & 0 \\ 0 & 0 & I_{s_3} \end{bmatrix} \tag{3.47}$$

From Eq. 3.46, one can see the that rank of matrix $M$ is 6, which is equal to the number of states. Thus, the linearized state space is fully controllable, which means one can manipulate all the states by changing the inputs.

To observe the quaternion states, one can choose the output matrix $C$ as:

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.48}$$

Thus, the state space for output becomes:

$$y = [C] \begin{bmatrix} \omega \\ g \end{bmatrix} \tag{3.49}$$

where the feedforward matrix, $D$ is assumed as zero.

### 3.5.2 Numerical linearization

Referring to Eq. 3.26, when linearizing a system, one needs the calculate the derivative to obtain the Jacobian term of the Taylor series. In the Analytical Linearization section, it is explained to get derivatives analytically. However, it might not be possible to get analytical derivatives when the system is too complex. To execute this, one can use the method of numerical linearization. Let us introduce the "Central Difference Method" to get the derivative terms numerically.

The central difference method is based on obtaining the derivative by basically calculating the slope of the line that passes through the point of the derivative. The visual representation of the method is given in Fig. 3.6. The slope of the line (which is approximately derivative) given in Fig. 3.6 can be calculated as:

$$f'(x) \approx \frac{f(x+\delta x) - f(x-\delta x)}{(x+\delta x) - (x-\delta x)} \tag{3.50}$$

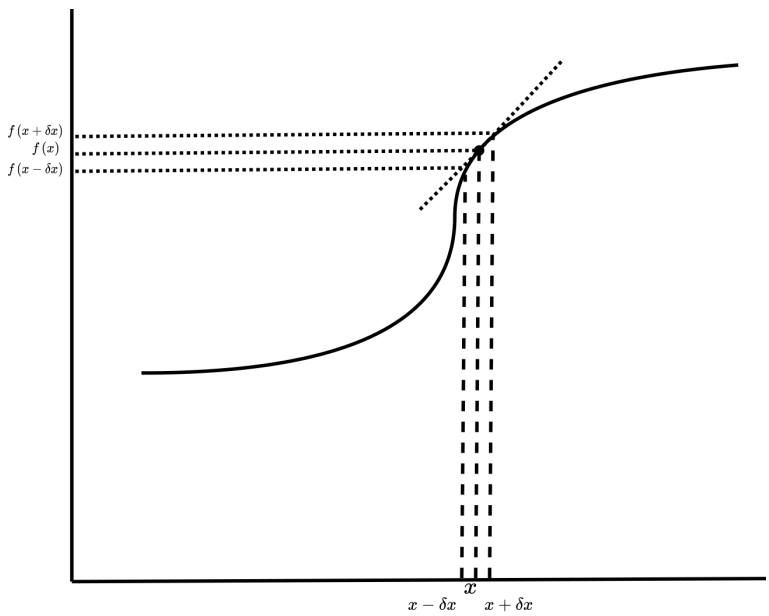


**Figure 3.6 :** Visualization of the central difference method.

One can obtain the numerical derivatives by using this method for complex models. It is also possible to apply it to the satellite model.

### 3.6 Comparison of Nonlinear and Linear Models

To proceed further, let us compare the nonlinear and linear models to validate the linearized equations. This is required since from now on, linearized model will be used for controller design and analysis. Remember that the linearized model was given in Eq. 3.44. The only parameter to construct this model is the satellite moment of inertia matrix. Let us choose the satellite moment of inertia matrix as the parameters of a real satellite taken from [28] as $I_s = diag([40.45, 42.09, 40.36])$.

A Simulink model was established to compare the models. The model is given in Fig 3.7. As the input, a band-limited white noise torque around the roll axis was given to the system with parameters of noise power 0.01 and a sample time of 0.1 seconds. To reduce the complexity of the figure, only the roll angle is shown. The output of nonlinear and linear models are given in Fig. 3.8. As shown by Fig. 3.8 the models are nearly identical when the angle is smaller than approximately 50 deg. Thus, it can be said that the linear model is quite successful for small angle approximation and can be used as a representation of the nonlinear model reliably.

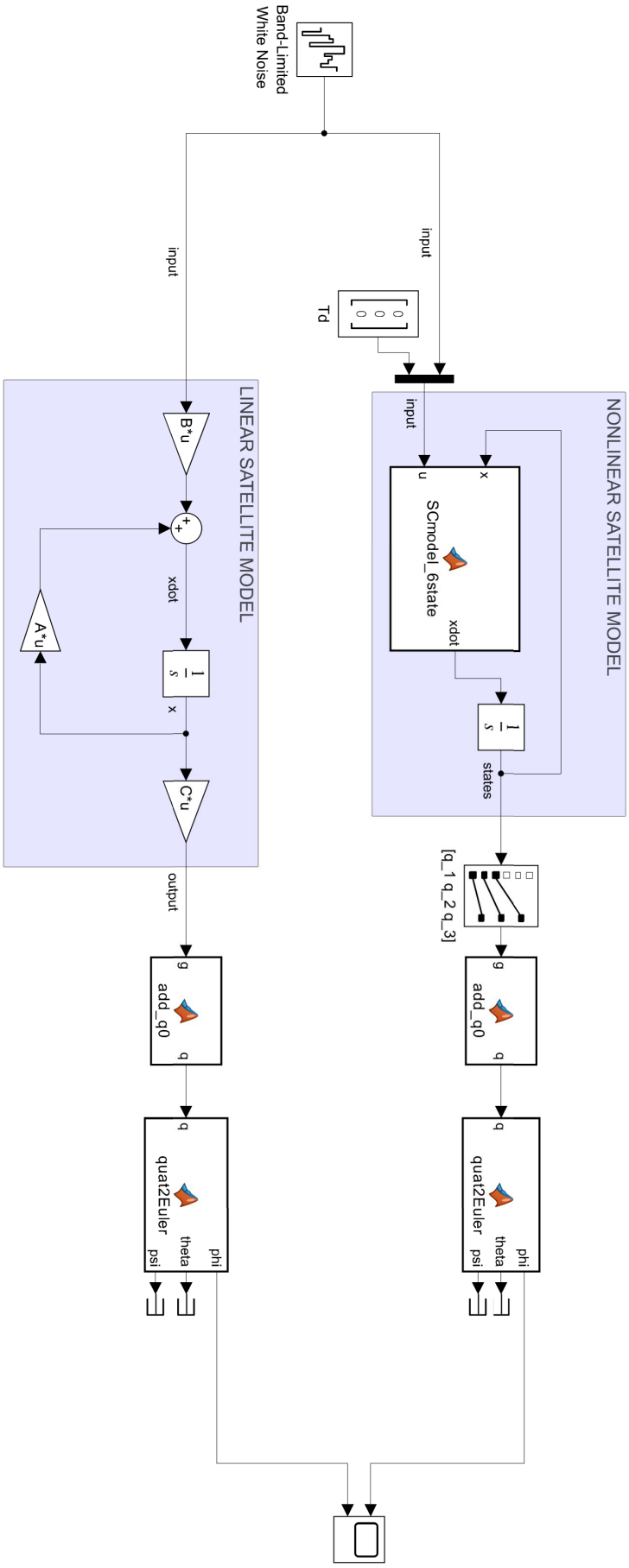


**Figure 3.7 :** Nonlinear-linear model comparison on Simulink.

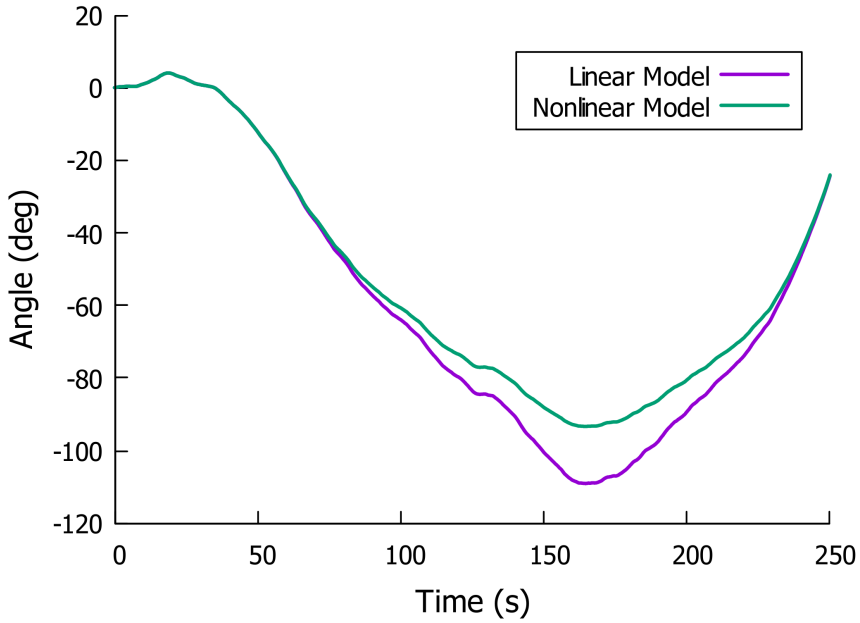


**Figure 3.8 :** Output of nonlinear and linear models for given input.

# 4. CONTROLLER DESIGN

In previous chapters, basic definitions and equations of motion to construct the satellite attitude control structure were explained. This chapter consists of explanation of different controllers that are used to compensate the error between the reference and the output angle. Referring to Fig. 3.1, The basic closed-loop control structure for a satellite attitude control system is given in Fig. 4.1. Actuator dynamics are ignored in this diagram. The controller basically takes the error between the reference and the output. Then, it produces proper control input for the dynamics system, that is control torque for the satellite attitude control system. The dynamics and kinematics blocks are used to produce the quaternion output to visualize the orientation state of the satellite.

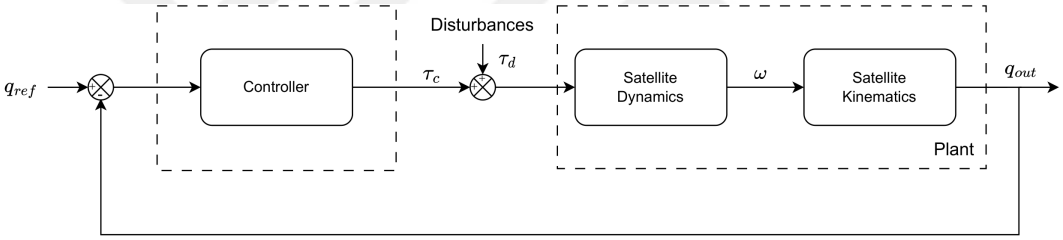


**Figure 4.1 :** Closed loop control diagram.

## 4.1 Linear Quadratic Regulator

Linear Quadratic Regulator (LQR) problem is based on the solution of an optimization problem. The formulation of an optimization problem can be expressed as:

$$minimize_x \quad J(x)$$

$$Subject\ to \quad g(x) \leq 0 \quad h(x) = 0$$

where $J(x)$ is called the cost function to be minimized, $g(x)$ is inequality constraint and $h(x)$ is the equality constraint. The number of constraints can be more than one as well as there might be no constraints. It is possible to apply this optimization problem to the control theory as well. By using this method, one can generate different problems such as fuel minimization, time minimization, and so on. Optimal control algorithms

are used in many fields such as planetary landing [29], vertical rocket landing [30], and so on.

Let us construct a state space system:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{4.1}$$

where $A$ is the state matrix, $B$ is the input matrix, $C$ is the output matrix, $D$ is the feed-forward matrix.

One can construct a cost function to be minimized as:

$$J = \int_0^\infty (x^T Qx + u^T Ru)dt \tag{4.2}$$

Where $Q_{n\times n}$ and $R_{m\times m}$ are positive definite matrices that determine the weights of tracking error and control input, respectively. The control law that satisfies the state space equation given in Eq. 4.1 and minimizes the cost function given in Eq. 4.2 is determined as:

$$u = -Kx \tag{4.3}$$

where

$$K = R^{-1}B^T P \tag{4.4}$$

$P_{n\times n}$ is found by solving the Riccati differential equation [6]:

$$0 = PA + A^T P - Q + PBR^{-1}B^T P \tag{4.5}$$

Once the state space, $Q$ and $R$ matrices are given, the steps of solving the LQR problem can be sorted as:

1. Solve the Ricatti differential equation and get $P$.
2. Found gain (K) from Eq. 4.12.

3. Use $K$ as the gain of state feedback control, given in Eq. 4.3.

Let us design a basic LQR controller by using the state space model explained in Eq. 3.44. The closed loop diagram of the LQR controller is given in Fig. 4.2. The state space model is explained as the open loop plant. The control law is explained in 4.3.

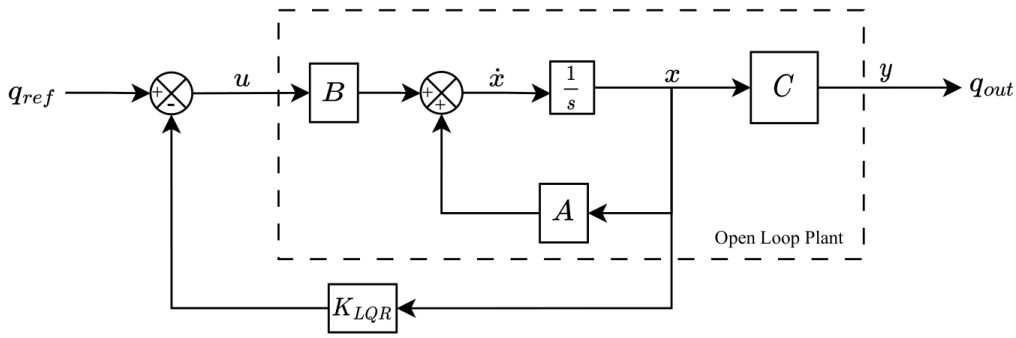


**Figure 4.2 :** LQR control diagram.

One can design a linear state space model by only knowing the satellite moment of inertia. Let us use $I_s$ as the same as given in Section 3.6. By using given $I_s$, the state space becomes:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{4.6}$$

where

$$A = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ \frac{1}{2}I_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{4.7}$$

$$B = \begin{bmatrix} I_s^{-1} \\ 0_{3\times3} \end{bmatrix} \tag{4.8}$$

$$C = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{4.9}$$

The weights Q and R are determined as identity matrices, namely the weight of 1.

$$Q = [I_{6\times6}] \tag{4.10}$$

$$R = [I_{3\times3}] \tag{4.11}$$

Inserting given A, B, Q and R matrices and solving the Riccati equation yields the LQR gain value, which is the gain that minimizes the cost function given in Eq. 4.2:

$$K_{LQR} = \begin{bmatrix} 6.4382 & 0 & 0 & 1 & 0 & 0 \\ 0 & 6.5643 & 0 & 0 & 1 & 0 \\ 0 & 0 & 6.4312 & 0 & 0 & 1 \end{bmatrix} \tag{4.12}$$

The stability of a closed loop state-space system for LQR can be found by finding the eigenvalues of $(A - BK_{LQR})$ which is obtained by giving the control input as $u = -K_{LQR}x$ as mentioned in LQR control problem.

For the gain value found in Eq. 4.12, $eig(A - BK_{LQR})$ is found as:

$$eig(A - BK_{LQR}) = \begin{bmatrix} -0.0796 + j0.0776 \\ -0.0796 - j0.0776 \\ -0.0797 + j0.0777 \\ -0.0797 - j0.0777 \\ -0.078 + j0.0761 \\ -0.078 - j0.0761 \end{bmatrix} \tag{4.13}$$

This result gives the closed loop poles. As given in Eq. 4.13, all of the poles lie on the left of the s-plane, which indicates that the closed loop system is stable.

Let us make a simulation for this LQR solution. The reference quaternion vector elements $[q_1 q_2 q_3]$ are chosen as 0.1, 0.2 and 0.3, respectively. The output of the simulation is given in Fig. 4.3.

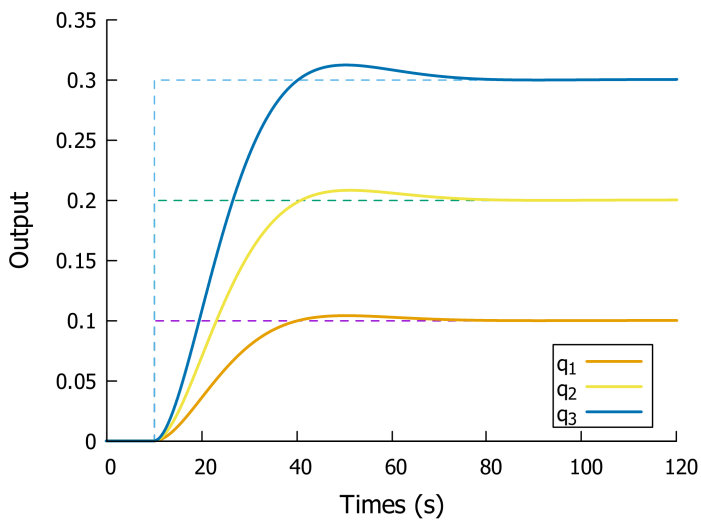


**Figure 4.3 :** LQR simulation output with given references.

As shown in Fig. 4.3, the satellite is able to track the given quaternion references accurately with zero steady-state error. Additionally, a custom reference signal has been produced to investigate the performance of LQR controller on tracking. To reduce the complexity of the graph, only $q_1$ custom reference has been produced. The output of $q_1$ element with reference is given in Fig. 4.4, which shows that the satellite is able to track the given quaternion element accurately.

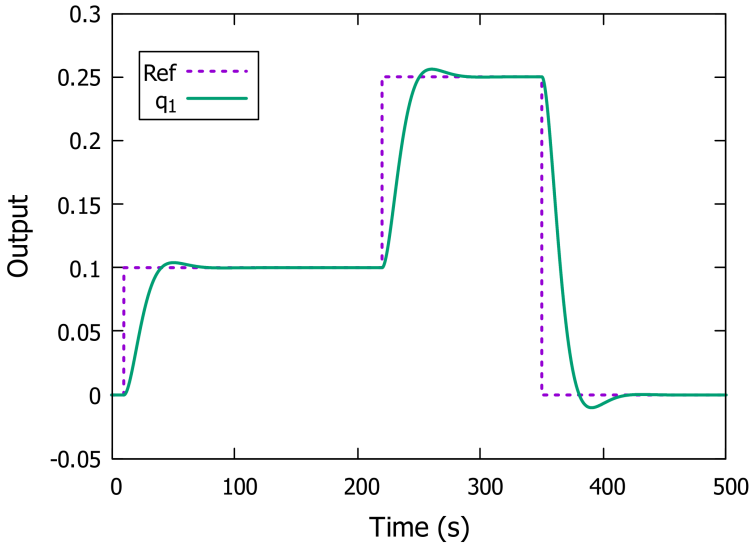


**Figure 4.4 :** LQR simulation output for a custom reference signal.

### 4.1.1 Controller tuning

The LQR control has two important parameters to be tuned: Q and R. One can investigate the effect on the Q parameter. To investigate that, let us keep R as the identity matrix, namely 1 and alter the Q parameter. The output of $q_1$ for different Q values is given in Fig. 4.5. As shown in this figure, one can see that a steady state error occurs when the Q value is lower or greater than 1. Thus, as an intuitive approach, one can set the Q parameter as 1.

It is also possible to examine the effect of R on the performance. Let us keep the Q value as 1 and alter the R value. The output of $q_1$ element for different R values is given in Fig. 4.6. This figure also shows that the values of R other than 1 have also steady-state error. Thus, one can say that as an intuitive approach, R must be set to 1.

It is known that Q and R parameters are the values that "penalize" the parameters of the cost function. Referring to the cost function to be minimized, the Q value has the effect on $x$, namely states. Thus, increasing the Q value means penalizing changes in

states, which means the minimum change in the states. Similarly, the R parameter has the effect on $u$, namely control input. Increasing the R parameter means penalizing the control input. Thus, increasing R means using less "energy".

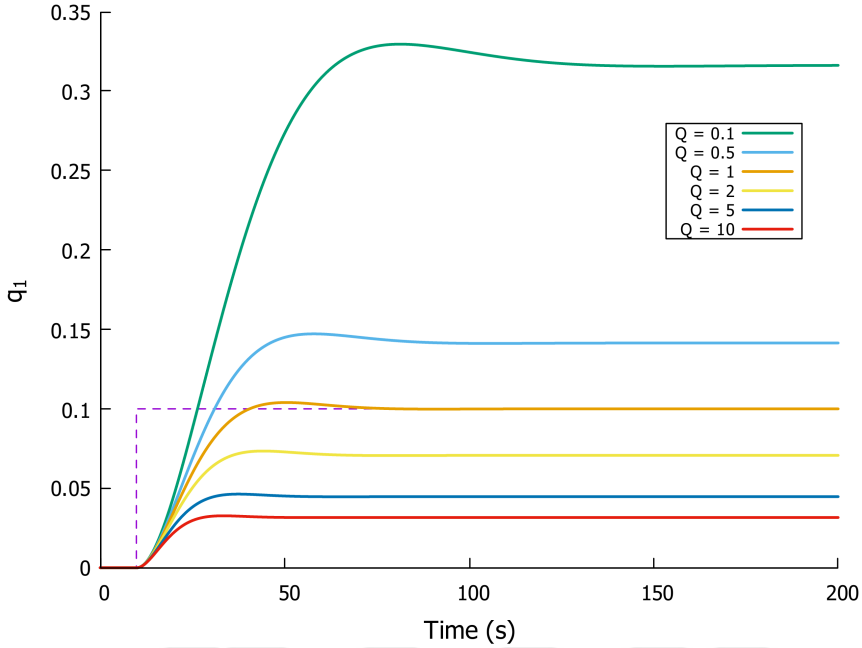


**Figure 4.5 :** LQR simulation output different Q values.

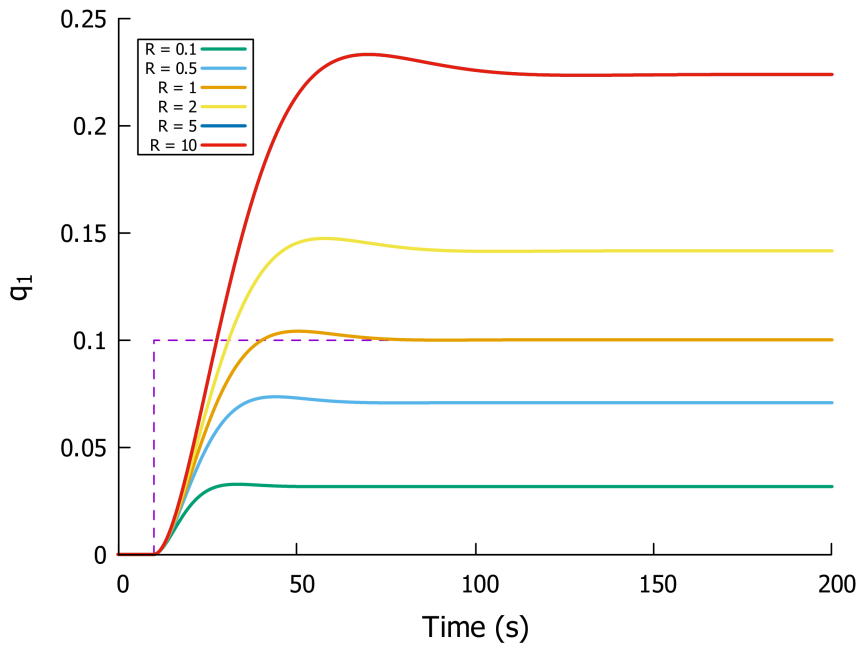


**Figure 4.6 :** LQR simulation output different R values.

It is also possible to include the constraints to the LQR solution. Since the LQR solution is executed for an unconstrained optimization problem, one can include a "saturation block" to the input of the dynamics subsystem to limit the control torque

values. The saturation block is a nonlinear type of block and is not included in the solution of the optimization problem. It is added subsequently. Thus, the saturation included LQR is not the "optimal" solution to this attitude control problem.

Let us saturate the control torque between $\pm$ 0.02 Nm and give a quaternion reference to the system as to be $q_1 = 0.2$. Outputs of saturated and unsaturated solutions of LQR are given in Fig. 4.7. Additionally, the control inputs for saturated and unsaturated cases are given in Fig. 4.8, which shows that the saturation block works fruitfully, the input is limited between $\pm$0.02 Nm.

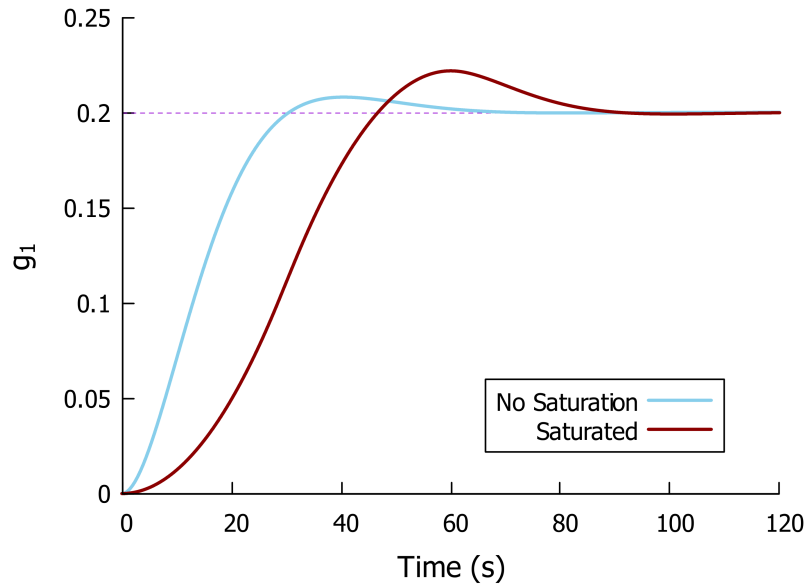


**Figure 4.7 :** Effect of controller saturation on $q_1$

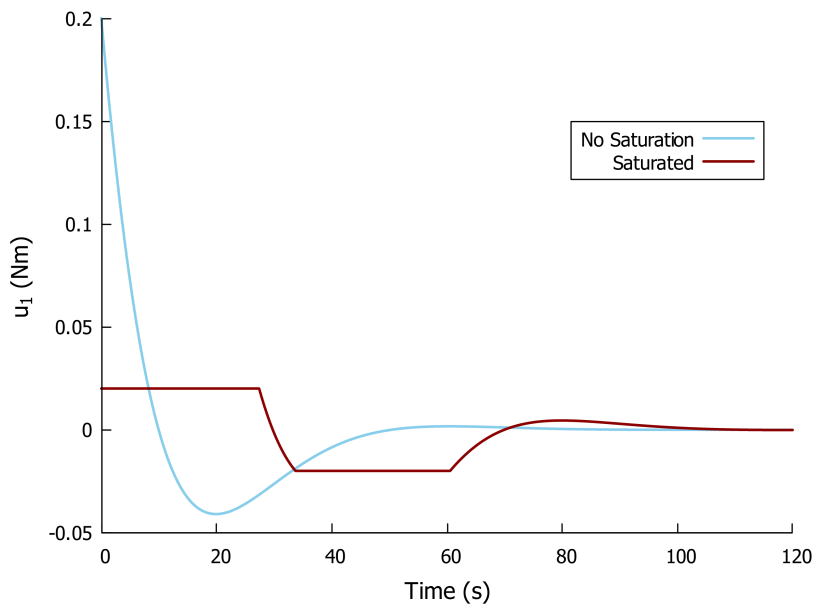


**Figure 4.8 :** LQR control torque with and without saturation.

## 4.2 Model Predictive Controller

Model Predictive Control (MPC) is also an optimal control solution. One of the most important differences between MPC and LQR is that the optimization problem is solved at each time step. MPC is based on the "prediction" of the next states and control inputs of the models. Thus, one needs a strong, successful model to use that control algorithm since the control inputs are updated according to the system model besides system measurements.

MPC has also two important parameters as Prediction Horizon, $N_p$ and control horizon $N_c$. These parameters determine the amount of prediction for the given system.

Consider a state-space system in discrete time:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{4.14}$$

Where $A_{n\times n}$ is the state matrix $B_{m\times m}$ is the control matrix and $C_{p\times n}$ is the output matrix where $p$ is the number of outputs.

Let us iterate the state space system given in (4.14) one step:

$$\begin{aligned} x(k+2) &= Ax(k+1) + Bu(k+1) \\ &= A(A(x(k) + Bu(k)) + Bu(k+1) \\ &= A^2x(k) + ABu(k) + Bu(k+1) \end{aligned} \tag{4.15}$$

The next step will be:

$$x(k+3) = Ax(k+2) + Bu(k+2) \tag{4.16}$$

inserting Eq. 4.15 into this yields:

$$x(k+3) = A^3x(k) + A^2Bu(k) + ABu(k+1) + Bu(k+2) \tag{4.17}$$

Let us bring Eqs. 4.15, 4.16 and 4.17 together to observe the pattern:

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) \\
x(k+2) &= A^2x(k) + ABu(k) + Bu(k+1) \\
x(k+3) &= A^3x(k) + A^2Bu(k) + ABu(k+1) + Bu(k+2)
\end{aligned}
\tag{4.18}
$$

As shown in Eq. 4.18, one can "predict" the ongoing states with the equation:

$$
\begin{bmatrix} x(k+1) \\ x(k+2) \\ x(k+3) \\ \vdots \\ x(k+N_p) \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^{N_p} \end{bmatrix} x(k) +
\begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ A^2B & AB & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ A^{N_p-1}B & A^{N_p-2}B & A^{N_p-3}B & \cdots & B \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N_p-1) \end{bmatrix}
\tag{4.19}
$$

which can be written in a compact form:

$$
x_{\rightarrow} = Fx(k) + Hu_{\rightarrow}
\tag{4.20}
$$

Where $N_p$ is the prediction horizon. One can write the prediction equations by using Eq. 4.19 to predict the future states and control inputs of the system. In this system, the prediction horizon is the same as the control horizon. Since there is no issue concerning computational burden, the control horizon is kept constant. By multiplying predicted states with the output Matrix, $C$, one can also obtain the predicted outputs since $y = Cx$. The cost function to be minimized for the MPC controller is similar to LQR control:

$$
\min J \qquad \sum_{k=0}^{N_p-1} x_k^T Q x_k + u_k^T R u_k
\tag{4.21}
$$

subject to:

$$
\begin{aligned}
x_{k+1} &= Ax_k + Bu_k \qquad k = 1, 2, \ldots N_p - 1 \\
u_{min} &\leq u_k \leq u_{max}
\end{aligned}
$$

where the matrices $Q$ and $R$ are the weights of states and inputs, respectively. The block diagram of the MPC-controlled satellite system is given in Fig. 4.9. As shown in the block diagram, in the controller section, there is an optimizer that solves the optimization problem (given in 4.21) in every time step to find the feasible solution, which is control torque for the satellite. The optimizer works with the prediction model and predicts the ongoing states of the system. Furthermore, predictions are also updated according to system output, $q_{ref}$ for this case. The produced control torque is used in the satellite dynamics subsystem to obtain the angular velocity vector of the satellite, whereas the angular velocity vector is then used to obtain the quaternion state of the satellite.

One of the most powerful sides of MPC is its constraint-handling capability. This means that the user can provide specific constraints such as interval of control input (which might be dc motor voltage constraint for a reaction wheel), and constraints on states and outputs. For instance, one can also constrain the output quaternions to make sure that the satellite is operating under the given angle interval. An unconstrained MPC is also a controller type that has no constraints such as the classical LQR controller. Unconstrained MPC has an analytical solution [31] whereas to solve constrained MPC problems, one needs to define the optimization problem as a "quadratic programming" problem and solve it numerically. Numerous open-source libraries can be used to solve quadratic programming problems in many different languages.

Let us also perform a simulation to investigate the performance of the MPC controller. The simulation parameters are given in Table 4.1. As distinct from the LQR controller, MPC has the capability of constraint handling. Thus, one can determine constraints for control torque. Let us determine this control torque constraint as given in Table 4.1. Reference quaternion vector is chosen as $q_{ref} = [0.08, 0.05, 0.03]$. Under these conditions step response of quaternion outputs, $q_{out}$ is given in Fig. 4.10.

**Table 4.1 :** MPC simulation parameters.

| Parameter | Symbol | Value | Unit |
|---|---|---|---|
| Satellite moment of inertia | $I_s$ | $diag([40.45, 42.09, 40.36])$ | $kgm^2$ |
| Control torque constraint | $T_{c_{max}}$ | $\pm 0.02$ | $Nm$ |
| Initial angular velocity | $\omega_0$ | [0 0 0] | $rad/s$ |
| Initial quaternion elements | $g_0$ | [0 0 0] | - |

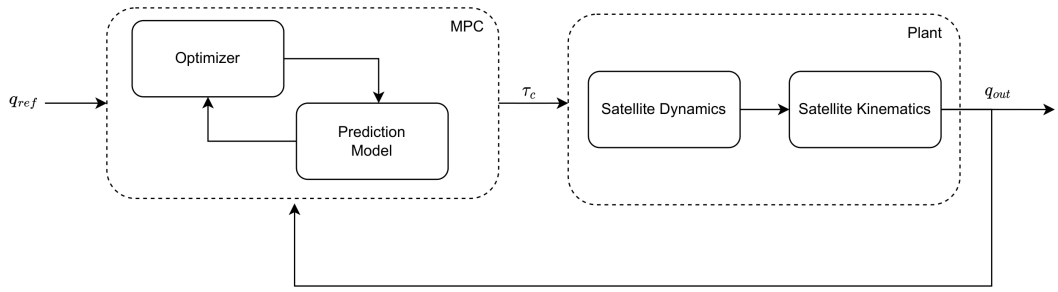


**Figure 4.9 :** MPC block diagram.

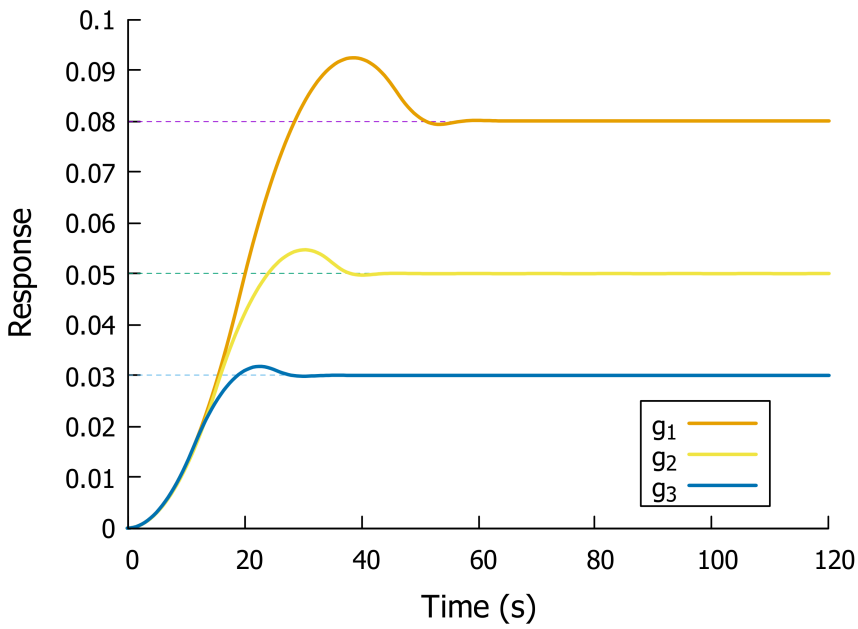


**Figure 4.10 :** Step response of the MPC controller.

Additionally, the control torque input plot for the case of Fig. 4.10 is given in Fig. 4.11. Fig. 4.10 shows that the satellite is able to track given references accurately. Furthermore, Fig. 4.11 shows that the input constraint is satisfied accordingly between $\pm$ 0.02 Nm. This is the solution to a quadratic programming problem with input constraints. So the optimal solution also includes the constraints as well. The constraints are not included subsequently, contrary to the LQR controller.

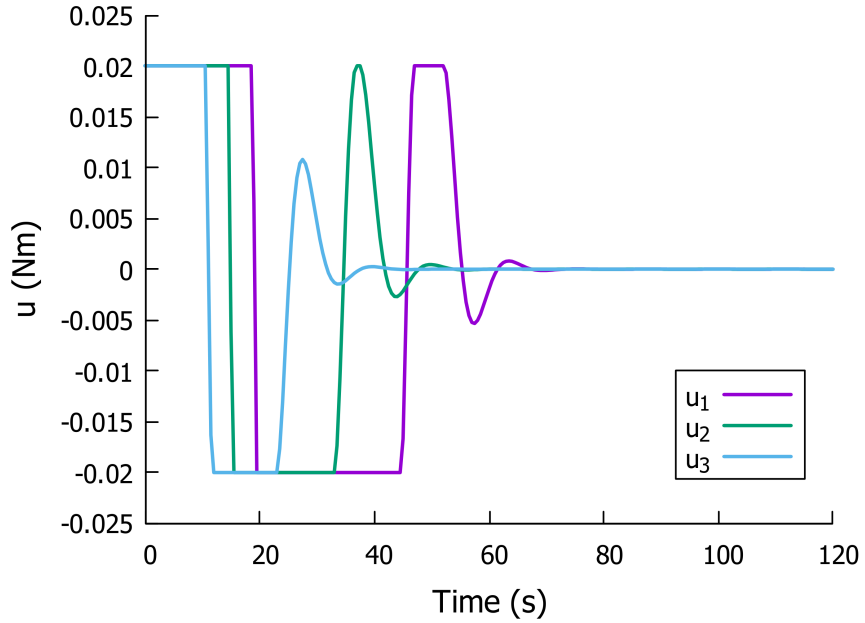


**Figure 4.11 :** Control torques for MPC controller.

## 4.3 Comparison of Controllers

One can compare LQR and MPC controllers with respect to their disturbance rejection performances. To evaluate this, a step (constant) output disturbance was given into the system. The reference quaternion element, $g_1$ was taken as 0.1. Only $g_1$ was examined to reduce the complexity of the graph. The disturbance was given into the system at 70th second and the magnitude is determined as 0.01. The output response of LQR and MPC controllers are given in Fig. 4.12.

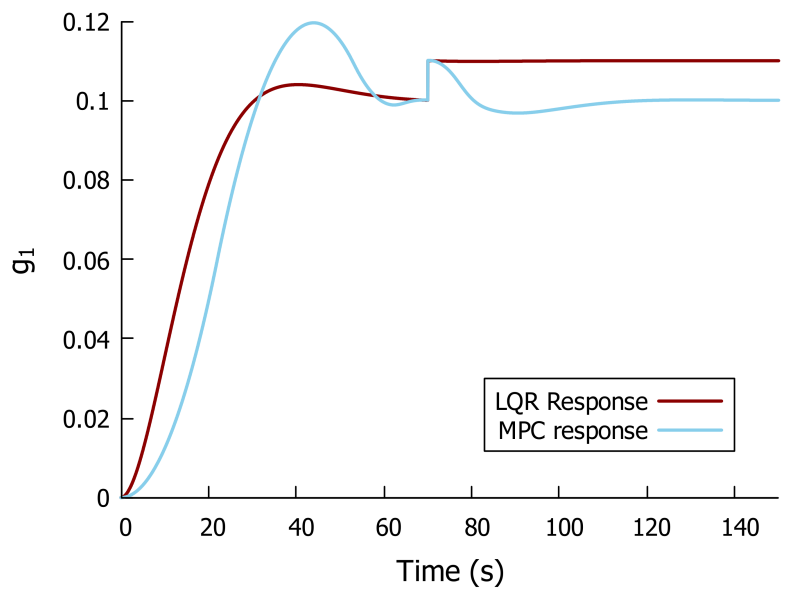


**Figure 4.12 :** Disturbance rejection performance of LQR and MPC controllers.

It is obvious that the MPC controller successfully compensates the disturbance whereas the LQR controller fails to do so. Since the disturbance is given as an output disturbance, a sudden change in the output occurs, as expected. The disturbance rejection performance of the controllers can be also compared by using a mathematical approach, Root mean square error (RMSE). RMSE values of the controllers between the output and reference were calculated after the disturbance was applied. RMSE value for the LQR controller was found as 0.01 whereas this value is 0.0029 for the MPC controller, which is exceptionally lower than LQR.

# 5. CONCLUSIONS AND RECOMMENDATIONS

In this thesis, design, analysis and optimization of controllers for a satellite attitude control system have been explored. The main purpose was to investigate improving the performance and robustness of attitude control systems. To achieve this, LQR and MPC controllers were investigated and compared in terms of disturbance rejection performances.

In the initial chapters, the required definitions to design a satellite attitude control system were introduced. Satellite mathematical model were explained as dynamics and kinematics subsystems. The nonlinear system dynamics were linearized around the operating point. The linearization process was explained in detail and a comparison of nonlinear and linear models was executed.

The controller design mainly focused on optimal control methods. Initially, LQR controller was introduced with tracking performance evaluation. The tracking performance was also investigated with the tuning of cost function weights. It has been observed that the cost function weights Q and R have a significant effect on the system output. It was also observed that the LQR is capable of stabilizing the attitude of the satellite and tracking the given reference accurately. Additionally, an investigation of the MPC controller was executed in detail. Model Predictive Controller has the advantage of predicting the future states and outputs by the prediction model as well as it has the capability of constraint handling. Then, LQR and MPC controllers were compared in terms of their disturbance rejection performances. It was observed that the MPC controller was able to compensate the disturbance successfully and brings the state to the reference quaternion whereas the LQR controller was not able to compensate the error between reference and output after the disturbance was applied.

Future research could explore the implementation of advanced control algorithms such as Linear Quadratic Integrator, Linear Quadratic Gaussian, Nonlinear control and so on. Advanced estimators may also be developed such as Kalman Filter, and Particle Filter to include the measurement noise into the system model and enhance the control

models in the presence of this noise. Additionally, model-based design techniques such as processor-in-the-loop, and hardware-in-the-loop can be developed in embedded hardware to achieve more realistic results.

This thesis contributes the body knowledge for satellite attitude control design and provides profitable insights for the engineers and researchers employed in satellite missions. It also aims to provide intuition on selecting and optimizing appropriate control parameters.

## REFERENCES


[1] **Meza, L.**, **Tung, F.**, **Anandakrishnan, S.**, **Spector, V. and Hyde, T.** (2005). Line of sight stabilization of James Webb space telescope, *27th Annual AAS Guidance and Control Conference*, AAS-05-002.

[2] **Zhou, K.**, **Huang, H.**, **Wang, X. and Sun, L.** (2017). Magnetic attitude control for Earth-pointing satellites in the presence of gravity gradient, *Aerospace Science and Technology*, *60*, 115–123.

[3] **Ibrahim, S.A. and Yamaguchi, E.** (2019). Comparison of solar radiation torque and power generation of deployable solar panel configurations on nanosatellites, *Aerospace*, *6*(5), 50.

[4] **Xu, G.**, **Tianhe, X.**, **Chen, W. and Yeh, T.K.** (2011). Analytical solution of a satellite orbit disturbed by atmospheric drag, *Monthly Notices of the Royal Astronomical Society*, *410*(1), 654–662.

[5] **Xiao, B.**, **Yin, S. and Kaynak, O.** (2016). Attitude stabilization control of flexible satellites with high accuracy: An estimator-based approach, *IEEE/ASME Transactions on Mechatronics*, *22*(1), 349–358.

[6] **Ofodile, I.**, **Ehrpais, H.**, **Slavinskis, A. and Anbarjafari, G.** (2019). Stabilised LQR control and optimised spin rate control for nanosatellites, *2019 9th International Conference on Recent Advances in Space Technologies (RAST)*, IEEE, pp.715–722.

[7] **Navabi, M. and Hosseini, M.** (2017). Spacecraft quaternion based attitude input-output feedback linearization control using reaction wheels, *2017 8th International Conference on Recent Advances in Space Technologies (RAST)*, IEEE, pp.97–103.

[8] **Fakoor, M.**, **Nikpay, S. and Kalhor, A.** (2021). On the ability of sliding mode and LQR controllers optimized with PSO in attitude control of a flexible 4-DOF satellite with time-varying payload, *Advances in Space Research*, *67*(1), 334–349.

[9] **Yang, Y.** (2012). Analytic LQR design for spacecraft control system based on quaternion model, *Journal of aerospace engineering*, *25*(3), 448–453.

[10] **Mamarasulov, J.**, **Holaza, J. and Gulan, M.** (2021). Design of an MPC based attitude control system for a CubeSat nanosatellite, *2021 23rd International Conference on Process Control (PC)*, IEEE, pp.266–271.

[11] **Wood, M.**, **Chen, W.H. and Fertin, D.** (2006). Model predictive control of low earth orbiting spacecraft with magneto-torquers, *2006 IEEE Conference*

*on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, IEEE, pp.2908–2913.

[12] **Mirshams, M. and Khosrojerdi, M.** (2016). Attitude control of an underactuated spacecraft using tube-based MPC approach, *Aerospace Science and Technology*, *48*, 140–145.

[13] **Hu, Y.**, **Lu, Z.**, **Ling, K.**, **Liao, W. and Zhang, X.** (2022). Multiplexed MPC attitude control of a moving mass satellite using dual-rate piecewise affine model, *Aerospace Science and Technology*, *128*, 107778.

[14] **Walsh, A.**, **Di Cairano, S. and Weiss, A.** (2016). MPC for coupled station keeping, attitude control, and momentum management of low-thrust geostationary satellites, *2016 American Control Conference (ACC)*, IEEE, pp.7408–7413.

[15] **Walter, U.**, (2018). Reference Frames, Springer International Publishing, Cham, pp.661–675, `https://doi.org/10.1007/978-3-319-74373-8_13`.

[16] **Cai, G.**, **Chen, B.M. and Lee, T.H.**, (2011). Coordinate Systems and Transformations, Springer London, London, pp.23–34, `https://doi.org/10.1007/978-0-85729-635-1_2`.

[17] **Svartveit, K.**, (2003), Attitude determination of the NCUBE satellite.

[18] **Du, W.**, **Wie, B. and Whorton, M.** (2008). Dynamic modeling and flight control simulation of a large flexible launch vehicle, *AIAA Guidance, Navigation and Control Conference and Exhibit*, p.6620.

[19] **Lavezzi, G.**, **Grøttey, M.E. and Ciarcià, M.** (2019). Attitude control strategies for an imaging cubesat, *2019 IEEE International Conference on Electro Information Technology (EIT)*, IEEE, pp.149–155.

[20] **Ed., S.W.R.H.L.P.F.H.M.R.S. and or Corr. M., D.H.** (1844). II. On quaternions; or on a new system of imaginaries in algebra, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *25*(163), 10–13, `https://doi.org/10.1080/14786444408644923`, `https://doi.org/10.1080/14786444408644923`.

[21] **Diebel, J.** *et al.* (2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors, *Matrix*, *58*(15-16), 1–35.

[22] **Fresk, E. and Nikolakopoulos, G.** (2013). Full quaternion based attitude control for a quadrotor, *2013 European control conference (ECC)*, IEEE, pp.3864–3869.

[23] **Sidi, M.J.** (1997). *Spacecraft dynamics and control: a practical engineering approach*, volume 7, Cambridge university press.

[24] **Yang, Y.** (2012). Spacecraft attitude determination and control: Quaternion based method, *Annual Reviews in Control*, *36*(2), 198–219.

[25] **Nudehi, S.S.**, **Farooq, U.**, **Alasty, A. and Issa, J.** (2008). Satellite attitude control using three reaction wheels, *2008 American control conference*, IEEE, pp.4850–4855.

[26] **Lee, D.**, **Springmann, J.**, **Spangelo, S. and Cutler, J.** (2011). Satellite dynamics simulator development using lie group variational integrator, *AIAA Modeling and Simulation Technologies Conference*, p.6430.

[27] **Yang, Y.** (2010). Quaternion based model for momentum biased nadir pointing spacecraft, *Aerospace Science and Technology*, *14*(3), 199–202.

[28] **Horri, N. and Hodgart, S.** (2003). Attitude stabilization of an underactuated satellite using two wheels, *IEEE Aerospace conference*, IEEE, pp.2629–2635.

[29] **Acikmese, B.**, **Casoliva, J.**, **Carson, J. and Blackmore, L.** (2012). G-fold: A real-time implementable fuel optimal large divert guidance algorithm for planetary pinpoint landing, *Concepts and Approaches for Mars Exploration*, *1679*, 4193.

[30] **Liu, X.** (2019). Fuel-optimal rocket landing with aerodynamic controls, *Journal of Guidance, Control, and Dynamics*, *42*(1), 65–77.

[31] **Maciejowski, J.M.** (2001). *Predictive control with constraints*, Addison-Wesley Educational, Boston, MA.

**APPENDICES**

**APPENDIX A.1 :** MATLAB Function block: "SCmodel_6state"
**APPENDIX A.2 :** MATLAB Function block: "quat2Euler"
**APPENDIX A.3 :** MATLAB Code: Controllability Check Matrix

**APPENDIX A.1: MATLAB Function block: "SCmodel_7state"**

```
function xdot = SCmodel_7state(u,x)

% xdot (1:3) = dw/dt
% xdot (4:6) = dg/dt

w = [x(1); x(2); x(3)];  % Angular velocity state
q = [x(4); x(5); x(6); x(7)];  % Quaternion state

Tc = u(1:3);  % Control torque
Td = u(4:6);  % Disturbance torque


S = [0    -w(3)  w(2);%Skew symmetric cross product matrix
     w(3)    0   -w(1);
    -w(2)   w(1)    0];


Omega = [ 0    -w(1)   -w(2)   -w(3);
          w(1)    0      w(3)   -w(2);
          w(2)  -w(3)     0      w(1);
          w(3)  -w(2)   -w(1)    0];


Is = diag([40.45 42.09 40.36]);

xdot = zeros(length(x),1); % Initialization

xdot(1:3) = Is\(-S*Is*w + Tc + Td);  % Satellite dynamics
xdot(4:7) = 0.5*Omega*q; % Satellite kinematics
end
```

**APPENDIX A.2: MATLAB Function block: "quat2Euler"**

```
function [phi, theta, psi] = quat2Euler(q)

q0 = q(1);
q1 = q(2);
q2 = q(3);
q3 = q(4);

phi   = atan2(2*q0*q1 + 2*q2*q3, q0^2 - q1^2 - q2^2 ...
+ q3^2);
theta = asin(2*q0*q2 - 2*q3*q1);
psi   = atan2(2*q0*q3 + 2*q1*q2, q0^2 + q1^2 - q2^2 ...
- q3^2);

phi = phi*180/pi;
theta = theta*180/pi;
psi = psi*180/pi;

end
```

**APPENDIX A.3: MATLAB Code: Controllability Check Matrix**

```
clear;
clc;

syms I_1 I_2 I_3


Is = [I_1   0   0;  %Satellite moment of inertia matrix
      0   I_2  0;
      0    0  I_3];


A = [ zeros(3,3)   zeros(3,3);
     0.5*eye(3,3)  zeros(3,3)];

B = [inv(Is); zeros(3,3)];


M = [B A*B A^2*B A^3*B A^4*B A^5*B]  % Controllability matrix
```

**CURRICULUM VITAE**

**Name Surname** **:** Emre SAYIN

**PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:**

- **Sayin, E. & Bayezit, I.** 2022. Model-Based Design of a Satellite Attitude Control System Using 2 DoF Reaction Wheel Testbed. *International Graduate Research Symposium (IGRS' 22)*, June 1–3, 2022 Istanbul, Turkiye.