**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY**

# HARDWARE IN THE LOOP SYSTEM DEVELOPMENT FOR MODELING AND CONTROL OF MULTIROTOR VEHICLES

**M.Sc. THESIS**

**Muhsin HANÇER**

**Department of Aeronautical and Astronautical Engineering**

**Aeronautical and Astronautical Engineering Programme**

**AUGUST 2017**

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE

ENGINEERING AND TECHNOLOGY

# HARDWARE IN THE LOOP SYSTEM DEVELOPMENT FOR MODELING AND CONTROL OF MULTIROTOR VEHICLES

M.Sc. THESIS

Muhsin HANÇER
(511141146)

Department of Aeronautical and Astronautical Engineering

Aeronautical and Astronautical Engineering Programme

Thesis Advisor: Asst. Prof. Dr. İsmail BAYEZİT

AUGUST 2017

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

# MULTİ ROTORLU HAVA ARAÇLARININ MODELLENMESİ VE KONTROLÜ İÇİN DONANIM ÇEVRİMLİ BENZETİM SİSTEMİ TASARIMI

YÜKSEK LİSANS TEZİ

Muhsin HANÇER
(511141146)

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

Tez Danışmanı: Asst. Prof. Dr. İsmail BAYEZİT

AĞUSTOS 2017

**Muhsin HANÇER**, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 511141146 successfully defended the thesis entitled "**HARDWARE IN THE LOOP SYSTEM DEVELOPMENT FOR MODELING AND CONTROL OF MULTIROTOR VEHICLES"**, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

**Thesis Advisor :** **Asst. Prof. Dr. İsmail BAYEZİT** ..............................
Istanbul Technical University

**Jury Members :** **Prof. Dr. Cengiz HACIZADE** ..............................
Istanbul Technical University

**Asst. Prof. Dr. İlker ÜSTOĞLU** ..............................
Yıldız Technical University

**Asst. Prof. Dr. İsmail BAYEZİT** ..............................
Istanbul Technical University

**Date of Submission :** **18 August 2017**
**Date of Defense :** **21 August 2017**

***To my wife and daughter,***

## FOREWORD

First and foremost, I would like to gratefully thank my supervisor Asst. Prof. Dr. İsmail Bayezit for his guidance, encouragement and kindness. I highly appreciate Prof. Dr. Mahmut Reyhanoğlu for his great guidance and valuable suggestions and contributions on my thesis and academic life. All the İTU professors and lectures who have helped me to develop my academic skills.

A very special thanks to Mr. Rahman Bitirgen for his truly great help in this work. And a great appreciation to all my friend in campus life, specifically, Mr. Muhammet Öztürk, Mr. Onur Altınçelik, Mr. Müslim Çakır, Mr. Muhammet Şakir Kaya, and Mr. Ahmet Gökay Öztürk. I wish to express my appreciation to Mr. Afşin Baran, Mr. Kubilay Cebeci, and Model Based Design and Control Laboratory workers about their contribution on my thesis.

Moreover, I would like to thank my precious country of Türkiye for funding me as a research assistance during the my academic life in the programme of ÖYP.

Last but not least, I present the thesis to my beloved family and parents for their endless love and unwavering support throughout my life.

August 2017 Muhsin HANÇER
Aeronautical and Astronautical Engineer

## TABLE OF CONTENTS

**Page**

## ABBREVIATIONS

| | |
|---|---|
| **HIL** | : Hardware-In-The-Loop |
| **UAV** | : Unmanned Air Vehicles |
| **VTOL** | : Vertical Take-Off and Landing |
| **3 DOF** | : Three Degree of Freedom |
| **ADC** | : Attitude Determination and Control |
| **CAD** | : Computer Aided Drafting |
| **MBDL** | : Model Based Design Laboratory |
| **3D** | : Three Dimensional |
| **CW** | : Clockwise |
| **CCW** | : Counterclockwise |
| **PWM** | : Pulse-width Modulation |
| **RPM** | : Rotation Per-Minute |
| **RPS** | : Rotation Per-Second |
| **ESC** | : Electronic Speed Controller |
| **PID** | : Proportional-Integral-Derivative |
| **FLC** | : Feedback Linearized Controller |

## SYMBOLS

| | |
|---|---|
| $\vec{\xi}_{\mathbf{i}}$ | : Inertial Frame Vector |
| $\mathbf{X_i}, \mathbf{Y_i}, \mathbf{Z_i}$ | : Inertial Frame Vector Components |
| $\vec{\xi}_{\mathbf{v}}$ | : Vehicle Frame Vector |
| $\mathbf{X_v}, \mathbf{Y_v}, \mathbf{Z_v}$ | : Vehicle Frame Vector Components |
| $\vec{F}_{\mathbf{b}}$ | : Inertial Frame Vector |
| $\mathbf{X_b}, \mathbf{Y_b}, \mathbf{Z_b}$ | : Body Frame Vector Components |
| $\vec{\eta}_{\mathbf{v}}$ | : Body Euler Angle Vector |
| $\theta, \phi, \psi$ | : Body Euler Angle Vector Components |
| $\omega$ | : Angular Velocity of Motor |
| $\mathbf{K_t}$ | : Thrust Coefficient |
| $\mathbf{K_d}$ | : Drag Coefficient |
| $\rho$ | : Air Density |
| $\tau_\phi, \tau_\theta, \tau_\psi$ | : Roll, Pitch, Yaw Torques |
| $u, v, w$ | : Linear Velocity Components |
| $p, q, r$ | : Angular Velocity Components |
| $\mathbf{m}$ | : mass |
| $\mathbf{g}$ | : gravity |
| $\mathbf{T}$ | : Propeller Thrust |
| $\mathbf{D}$ | : Propeller Drag |
| $\mathbf{S}$ | : Scale Reaction Force |
| $\mathbf{d}$ | : Propeller Diameter |
| $\vec{F}_{\mathbf{G}}$ | : Gravity Force Vector |
| $U$ | : Control Input |
| $\mathbf{L_{yb}}$ | : Moment Arm of Back Motor to Center of Mass |
| $\mathbf{L_{xf}}$ | : Moment Arm of Front Motor to Center of Mass |
| $\Omega$ | : Body Angular velocity |
| $I_{\mathbf{motor}}$ | : Moment of Inertia of Motor |
| $m_{\mathbf{prop}}$ | : Mass of Propeller |
| $L_{\mathbf{prop}}$ | : Diameter of Propeller |
| $m_{\mathbf{motor}}$ | : Mass of Motor |
| $r_{\mathbf{motor}}$ | : Radius of Motor |
| $L$ | : Lagrangian Energy |
| $K$ | : Kinetic Energy |
| $P$ | : Potential Energy |
| $I$ | : Moments of Inertia |
| $I_{\mathbf{xx}}, I_{\mathbf{yy}}, I_{\mathbf{zz}}$ | : Components of Moments of Inertia |
| $\vec{M}^{\mathbf{B}}$ | : Total Body Moments Vector |

## LIST OF TABLES

**Page**

## LIST OF FIGURES

**Page**

# HARDWARE IN THE LOOP SYSTEM DEVELOPMENT FOR MODELING AND CONTROL OF MULTIROTOR VEHICLES

## SUMMARY

Unmanned air vehicles were designed to accomplish specific missions thats are dangerous to be fulfilled by man power. Nowadays the big companies want to reduce worker payments in companies budgets, so they tended to robotic applications. Not only big cargo but also restaurants or fast food companies are planning to carry out courier services via quadrotors for delivering cargo to costumers. Main reason for this changing is development of electronic devices, especially sensors, reduction of the prizes, and easily arrival UAVs parts. So, UAVs are going to be modern flying robots in the near future in all of the areas in mind.

The topic of the thesis is development of hardware in the loop system to model and control of multirotor vehicles. Other words, we want to model and control multirotor vehicles on a testbed. Why testbed? The testbed gives some advantages. Reducing time consuming such as, eliminating the necessity of going to flight area, cheap workable conditions, and reduction the material and part breakage are most of them.

This works is a sample of Model Based Design Process. Model Based Design (MDB) is a process that accelerating prototyping with cost-effective in a short time. MBD process phases are definition of requirement, mathematical modeling of the system(simulation), rapid prototyping, code generation,software in the loop (SIL), processor in the loop (PIL), hardware in the loop (HIL) and validation. First phase is definition of the system or product properties and requirements. The generating mathematical model of the system under boundary conditions effecting the system directly or indirectly in the simulation environment.

Regarding to the mathematical model, rapid prototyping and code generation phase is coming to test the system on the simulations carry out on the computer environment that is the most important point to reduce cost and time consume. Software in the lop simulations are carried out to analyzing generated codes on the host computer. Processor in the loop is analyzing the system on target CPU on the embedded controller to reduction of the processor errors.

Hardware-in-the-loop (HIL) simulation is a type of real-time simulation to test designed controller with real plant or system input out put via embedded hardware. HIL simulation shows how controller responds, in real time, to realistic virtual simulations. HIL can be use to determine if real physical system (plant) model is valid or not. The Gyroscopic testbed was designed for HIL tests of the quadrotor (IRIS+) for validating and implementing the designed different controller coefficients in real time environments.

The HIL testbed is designed in order to how a quadrotor attitude control test easily. Before preliminary design, literature search is done and different test beds are found for both quadrotor testbed and motor propeller testbed. Then working principle

of gyroscope is gave inspiration to design testbed. All of the designing processes are carried out on the computer aided drawing environment. Then production and assembly phase are done. Next problem is fixing the quadrotor on the testbench. A support part is designed and produced, then quadrotor vehicle is assembled on the testbench. The motor-propeller testbed is designed with same procedure of HIL testbench.

Mathematical modeling of a quadrotor is second chapter of the thesis. Before the implementation of the controller, the simulation of the system give foreknowledge about behavior of the system. The mathematical model of the quadrotor is needed to simulate. Newton-Euler and Euler-Lagrange methods are used to derivate nonlinear model of the system. In the mathematical model, thrust and drag coefficients are used for refer propellers reaction to angular velocities. Some assumptions are taken into account during the modeling.

The third chapter of the thesis is related with quadrotor and testbeds parameters estimation. IRIS+ quadrotor vehicle has four motor, one pair rotate clockwise, other rotate counterclockwise with propellers which are self locked type. IRIS+ quadrotor has cross type frame architecture that angle between front arms is one hundred twenty degree but back angle between back arms is one hundred forty degree.

Cross type quadrotor has basically two advantages over plus configuration. first, motor need less power for pitch and roll motion and remaining power can be used for increasing translational motion. Second advantage is providing clear vision for camera with obtuse angle in between front propellers. The reference coordinate frames are defined related to cross type frame quadrotors. The quadrotor moments of inertias are founded by experimental studies.Blade element theory is used for calculating the propellers thrust and drag coefficients.

Mathematical model of the HIL testbed is derived via Euler Lagrange energy method. The inner and outer circles affect the motion of the quadrotor seriously so differences of angels of the circles and quadrotor are taken into account to effect on moment of inertias.

Designing flight controller for simulation and implementation is the last section of the thesis. Main aim of the section is comparison PID and feedback linearization methods outputs both in simulation and real time environments. In simulation, PID and Feedback Linearization Controller is applied on the both nonlinear model of quadrotor and HIL testbench nonlinear model in SIMULINK. PID responses are figured as $\phi$, $\theta$, $\psi$, and $Z$ parameters of the systems outputs. Feedback Linearization controller designed to eliminate nonlinear terms to stabilize the quadrotor vehicle attitude motion with different $\lambda$ coefficients. The last part of the simulations is designing feedback linearization controller for HIL testbed with optimum $\lambda$ coefficients of $\phi$, $\theta$, and $\psi$. $\lambda$ of $Z$ is't needed because there isn't any motion in Z direction.

Implementation of the PID and Feedback Linearization controllers are verified on the Pixhawk open-source autopilot hardware. Pixhawk code is redesigned to implement controllers in real time. The flight data is graphed in MATLAB to visualize motion with changes of angles for both PID and Feedback Linearization controller. Main difference between the results of the PID and Feedback Linearization controller is seen on the response of outputs. PID controller is tracking the signal with higher overshoots and bigger oscillations then Feedback Linearization controller response. But PID is faster than Feedback Linearization Controller. Both of the controller are stayed in effect of moments of inertias of HIL testbed circles. It is easily seen in long time delay on the time response especially in $\psi$ angle.

Future works are planned to continue in this works line. Especially different nonlinear controller methods are to implement on airvehicles. Moreover, code generation on the autopilot hardware are planned on the UAVs.

# MULTİ ROTORLU HAVA ARAÇLARININ MODELLENMESİ VE KONTROLÜ İÇİN DONANIM ÇEVRİMLİ BENZETİM SİSTEMİ TASARIMI

## ÖZET

İnsansız hava araçları, insan sağlığı için tehlikeli görevleri yerine getirmek için tasarlanmış araçlardır. Günümüzde büyük bütçeli şirketler çalışanlarına ayırdıkları bütçeyi en küçük düzeye çekmek için robotik uygulamalara karşı eğilim içindedirler. Sadece büyük kargo şirketleri değil retoranlar veya fast food şirketleri müşterilerine kargo teslimatını quadrotorlar vasıtası ile ulaştırmayı planlamaktadırlar. Bu değişimin en büyük sebebleri ise elektronik cihazlardaki hızlı gelişim özelliklede sensörlerdeki, fiyatların ucuzlaması ve insansız hava araçlarının yedek parçalarına erişmedeki kolaylığın artması söylenilebilir. Kısaca insansız hava araçları yakın zamanda hayal edebileceğiniz her alanda uçan robotlar olarak karşınıza çıkabilir.

Tez başlığı, multirotorlu hava araçlarının modellenmesi ve kontrolü için donanım çevrimli benzetim sistemi tasarımı olarak seçilmiştir. Başka bir ifade ile bir hava aracının test düzeneği üzerinde modellenmesi ve kontrolünün yapılması olarak ifade edilebilir. Neden test düzeneğine ihtiyacımız var? Test düzenekleri birçok avantaj sunmaktadır. Zaman israfını azaltmak örneğin, uçuş testi için uçuş alanına gitme gereksinimini azaltması gibi, ucuz, güvenirliği yüksek çalışma şartları sunması ve malzeme ve yedek parça hasarlarını en aza indirerek maliyeti düşürmek olarak sıralayabiliriz.

Bu çalışmayı Model Tabanlı Tasarım (MTT) Sürecine örnek olarak sunabiliriz. Model tabanlı tasarım, en etkili, verimli, kısa sürede hızlı prototip geliştirme sürecine denir. MTT sürecinin fazlarının ve gereksinimlerin tanımlanması, matematiksel olarak sistemin modellenmesi, hızlı prototipleme, kod jenerasyonu, yazılımsal ve işlemcisel döngü ve validasyon olarak sıralayabiliriz. İlk evre, istenilen sistem yada ürünün gereksinimlerinin tanımlanması olarak ifade edilebilinir. Belirlenen sınır şartları altında sistemin dolaylı yada direk benzetim ortamında matemetiksel modelinin üretilmesi kısmını oluşturmaktadır.

Sonraki aşama ise, matematiksel modele göre, hızlı prototipleme ve kod jenerasyonudur. Bu aşamada üretilen kodlar, bilgisayar ortamında sistem modeli üzerinde test edilerek zaman ve maliyet israfı en düşük düzeye çekilmektedir. Yazılımsal döngüde, benzetim ortamında üretilen kodlar host bilgisayar üzerinde test edilmektedir. İşlemcisel döngüde ise gömülü sistemler kulanılarak hedef işlemci üzerinde test edilerek işlemci kaynaklı hatalar düzeltilmektedir.

Donanımsal çevrimli benzetim (DÇB) ise gömülü sistemler aracılığı ile gerçek sistem üzerinde sistem giriş ve çıkışları kullanılarak gerçek zamanalı benzetim çalışmalarını ifade etmektedir. Donanım çevrimli benzetim kontrolcünün nasıl tepki verdiğini, gereçek zamanlı olarak test etme fırsatını sunar.

DÇB test düzeneği, bir quadrotor aracının yönelme kontrolünü daha kolay test etmek için tasarlandı. Ön tasarım yapılmadan önce, motor-pervane test düzeneğinde

olduğu gibi literatürdeki test düzenekleri araştırıldı. Daha sonra ise jiroskop çalışma prensibinden ilham alınarak şuanki test düzeneği tasarlandı. Bütün tasarım süreçleri bilgisayar destekli çizim ortamlarında gerçekleştirildi. Akabinde üretim ve montaj süreçleri gerçekleştirildi. Bir sonraki problem ise quadrotorun test düzeneğine bağlanması oldu. Bu probleme çözüm olarak bir ara plaka tasarlandı ve üç boyutlu yazıcıda üretilip, quadrotor, test düzeneğine bağlandı. Motor-pervane test düzeneği de DÇB test düzeğindeki tasarım, üretim ve montaj süreçlerine benzer olarak üretildi.

Matematiksel modelinin çıkarılması, ikinci bölümün ana başlığını oluşturmaktadır. Kontrolcünün implementasyonun yapılmsından önce, benzetim ortamında sistem davranışı incelendi. Sistemin nonlinear matematik modeli elde edilmesinde Newton-Euler ve Euler-Lagrange metodları kullanıldı. Matematik modelde, motor ve pervanelerin açısal hızlarından elde edilen itki ve sürükleme katsayıları kullanıldı. Tez içinde detaylı olarak anlatılan kabuller doğrultusunda matematik model tamamlandı.

Üçüncü kısımda, test düzeneklerinin üretilmesinin yanında quadrotor ve test düzeneklerinin parametrelerin incelenmesi ve tahmini gerçekleştirildi. IRIS+ quadrotor dört motor ve pervaneye sahip bir araç olmakla birlikte, birer çift motor ve pervane saat yönünde dönmekte, diğer çiftler ise saat yönünün tersi istikamette dömektedirler. Ayrıca pervaneler kendinden kilitlenen bir yapıya sahiptir. IRIS+ quadrotor aracı ana gövde olarak cross tip olup ön motorlar arasındaki açı yüzyirmi derece,arka motorlar arasındaki açı ise yüz kırk derecedir.

Cross tip ana gövde, Plus tip gövdeye göre bazı avantajları bulunmaktadır. İlk olarak motorlar, yunuslama ve yalpalama hareketi için daha az enerji gereksinimi olurken öteleme hareketi için daha fazla kuvvet üretebilmektedirler. Diğer önemli avantajı ise motorlar arasındaki açının büyük olması nedeni ile kamera için daha temiz ve geniş çekim alanı kalmaktadır, zira pervaneler genelde kameranın bakış alanına girmektedir. Referans kordinat sistemleri cross tip gövdeye göre tanımlandı. Quadrotorun atalet momentleri deneysel metodlarla elde edildi. Pervane itki ve sürükleme katsayıları Blade Element Teorisine göre hesaplandı.

DÇB test düzeneğinin matematik modeli Euler-Lagrange enerji metodu kullanılarak elde edildi. Dış ve iç çemberlerin atalet momentleri quadrotorun hareketini ciddi olarak etkilmekte ve yönelme hareketlerinde sistemin toplam atalet momenti, açılara bağlı olarak değişkenlik göstermekte olup, bu etki model içinde ifade edildi.

Bu tezin son kısmında uçuş kontrolcülerin tasarlanması, benzetim ortamlarında (Matlab SIMULINK) test edilip, DÇB test düzeneği üzerinde implementasyonu yapıldı. Bu kısmın ana amacı ise PID ve Geribeslemeli Linearizasyon Metodlarının çıktılarının hem benzetim hemde gerçek zamanlı olarak uçuş kartı üzerinde karşılaştırılmasıdır. Benzetim ortamında, PID ve Geribeslemeli Linearizasyon kontrolcü, quadrotorun nonlinear modeli üzerinde test edilmiştir. DÇB test düeneğinin nonlinear modeli kullanarak Geribeslemeli Linearizasyon kontrolcü test edildi. PID kontrolcü, çıktı olarak $\phi$, $\theta$, $\psi$ ve $Z$ parametreleri verecek şekilde benzetim yapıldı. Fakat, test düzeneği öteleme hareketi yapma yeteneği bulunmadığı için Geribeslemeli Linearizasyon kontrolcü $\phi$, $\theta$ ve $\psi$ çıktı olarak oluşturuldu ve $\lambda$ katsayıları optimize edildi. Benzetim bölümün son kısmını, Geribeslemeli Linearizasyon kontrolcünün DÇB test düzeneğinin nonlinear modeli üzerinde, $\phi$, $\theta$ ve $\psi$ parametreleri için yedi farklı $\lambda$ katsayıları optimize edildi.

Açık kaynak yazılıma sahip Pixhawk Otopilot kartı üzerinde, tasarlanan PID ve Geribeslemeli Linearizasyon kontrolcülerinin implementasyonu ve validasyonu yapıldı. Pixhawk içerisinde çalışan kodlar gerçek zamanlı uçuş için gerekli kısımları tekrar düzenlendi ve tasarlandı. Hem PID hemde Geribeslemeli Linearizasyon kontrolcülerinden kaydedilen açı değişimlerini gösteren uçuş verileri Matlab programında grafik olarak oluşturuldu.

PID ve Geribeslemeli Linearizasyon kontrolcülerinin arasındaki en belirgin farklar ise sistem çıktılarında görüldü ve yorumlandı. PID kontrolcü yüksek aşım oranlarına ve salınımlara sahipken, Geribeslemeli Linearizasyon kontrolcü daha az aşım ve salınım oluşturduğu görüldü. Fakat PID kontrolcü çok daha hızlı yanıt verirken, Geribeslemeli Linearizasyon kontrolcü daha yumuşak ama yavaş yani uzun sürede istenilen konuma geldiği görüldü. Her iki kontrolcüde çemberlerin oluşturduğu yüksek atalet momentine karşı istenilen kontrol sinyalini takip etmede zorlandıkları görüldü. Özelliklede $\psi$ açısın takibinde bu açık olarak gözlendi.

Gelecek çalışmalarda, test düzeneği üzerinde Pixhawk otopilot kullanılarak, farklı nonlinear kontrol methdları test edilmesi planlanmaktadır. Ayrıca, Pixhawk kontrol kartı üzerinde kod jenerasyonu yapılması konusunda da hedefler bulunmaktadır.

# 1. INTRODUCTION

Unmanned Air Vehicles(UAV) are designed to operate via remote controllers or autonomous algorithms to fulfill specific flight missions. UAVs accomplish their missions with autonomous mode, image taking and processing, communication etc. Recent development on the sensor and electronics technology, easily reaching and cheap mechanical and electronic parts triggered increasing attention on the UAVs. Lots of applications was developed and usage of the UAVs are increasing day by day subject to this conditions nowadays. Especially specific missions, thats are dangerous to be fulfilled by man, such as security, surveillance, detection and follow-up of enemies and targets, border patrolling, traffic control, damage detection after natural disasters, investigation of crime scene and agricultural applications.

First use of UAVs were during 1st World War at first quarter of 19th century with radio controlled. At first years main aim is reduction of loss well-trained manpower at high fatality military operations. During the last thirty years, utility of UAVs increased relative to the technological development depending on high reliability. Small budget requirement and minimizing the loss of manpower are main advantages for increased attentions of UAVs. In military area many countries have UAVs for reconnaissance and spotting. But, just little countries have capability of shooting and destruction targets with UAVS equipped with missiles.

## 1.1 Quadrotor Vehicles

The categorization of the UAVs is complex because of lots of diversity but generally UAVs are divided into two group with regard to fixed or rotary wings according to aerodynamic features. First is fixed wing UAVs, generally called aeroplane or simply plane, generates lift via fixed mounted wing on the body of the plane. Second part is rotary wing UAVs, generally called helicopters or multi-rotors, generates lift via rotary propeller. Quadrotor vehicles, coming from multirotor family, have four motors and propellers to maneuver three axes. Also, quadrotors are generally classified into plus

(+), cross (X), (V), and (H) types according to the frame shape. In this work, IRIS+ quadrotor vehicle, has a cross (X) type frame architecture, is used to experimental works and simulations.

### 1.1.1 Literature review

In the literature, lots of works are done about quadrotor modeling and control. Because of the aims of this work which are nonlinear or linear control and mathematical modeling of cross type quadrotors, searching on the literature is focused on controlling quadrotors and testing on a testbench.

#### 1.1.1.1 Modeling and controlling quadrotor vehicles

Patric and his friends established a dynamic flight model and designed feedback, backstepping and, sliding mode nonlinear controllers on the quadrotor vehicle to simulate and validate the theoretical analysis [9]. Marcelo generated mathematical model, identified experimentally and controlled both linear controller such as PID, LQ besides modern robust mixed-sensitivity $H_\infty$ and $\mu$-synthesis with D-K iteration under parametric model uncertainty [10]. Wei derived six degree of freedom dynamic model of quadrotor to use for designing and implementation of the linear PID attitude controller implemented both in simulation and real time on the hardware controller. He used embedded code generation via developed open-source software setup to enable control software prototyping [11]. Bouabdallah modeled, designed and controlled microquadrotor with based on Lyapunov theory, PID, LQ, sliding mode, and backstepping methods for attitude and altitude control with simulations and validations [12]. Seung and his friends designed new type quadrotor and identified physical properties. Thy showed high maneuvering capabilities of the vehicle via embedded flight controller using PID coefficients [13].

Bouabdallah and Siegwart derived quadrotor model with the different aerodynamic coefficients. They described integral backstepping control approach for control of attitude, altitude and position of autonomous quadrotor to takeoff, landing and collision avoidance [14]. Pounds, Mahony and Corke developed a larger quadrotor vehicle than other vehicles in reseach at same time. They focused on pitch and roll motion control with high sensitivity and disturbance rejection capabilities [15]. Xu and Özgüner designed new fligth controller of an autonomous quadrotor based on sliding mode control to eliminate model errors and parametric uncertainties and disturbances [16]. An and friends designed a second order geometric sliding mode attitude observer for angular velocity estimation of quadrotor vehicle [5]. They presented that results of second order geometric sliding mode attitude observer is better than taraditional quaternion based sliding mode observer on the real time testbed. Lebedev derived six degree of freedom mathematical model, designed, and produced prototype of the quadrotor. He simulated the pole placement method firstly then validated on the testbench to compare simulation and validation results [17]. Altuğ, Ostrowski and Mahony estimated the position of the quadrotor with a ground camera. They designed mode-based feedback linearization and sliding mode controller to simulate and validate with experimental works [18]. Bresciani carried out Newton-Euler modeling and control of dynamics of quadrotor on simulation and real test platform with PID controller [19].

Reyhanoglu, Damen, and MacKunis presented a sliding mode controller on the quadrotor vehicle. They used sliding mode observer to estimate angular velocities and proved asymptotic regulation of the quadrotor to control attitude based Lyapunov analysis [20].

#### 1.1.1.2 Quadrotor testbeds at literature

Many researchers designed different kind of testbeds for experimental works on the controlling of quadrotor vehicle. Some of them are presented in this subsection.

Bernstein, Kayacan, and Prach studied an experimental evaluation of the forward-propagating Riccati equation control. They used nonlinear dynamics of Quanser 3 DOF Hover experimental testbench, as seen in Figure 1.1-a, to update feedback gains in real time and compared with conventional linear quadratic regulator [1].Nicol, Macnab, and Ramirez-Serrano worked on a direct approximate adaptive controller for an experimental prototype quadrotor, as seen in Figure 1.1-b, via Cerebellar Model Arithmetic Computer nonlinear approximators. They tested quadrotor performance with changing total weights of the vehicle via updating adaptive parameters [2]. Alexis, Nikolakopoulos, and Tzes designed and verificated a Contrained Finete Time Optimal controller for attitude control of quadrotor on the

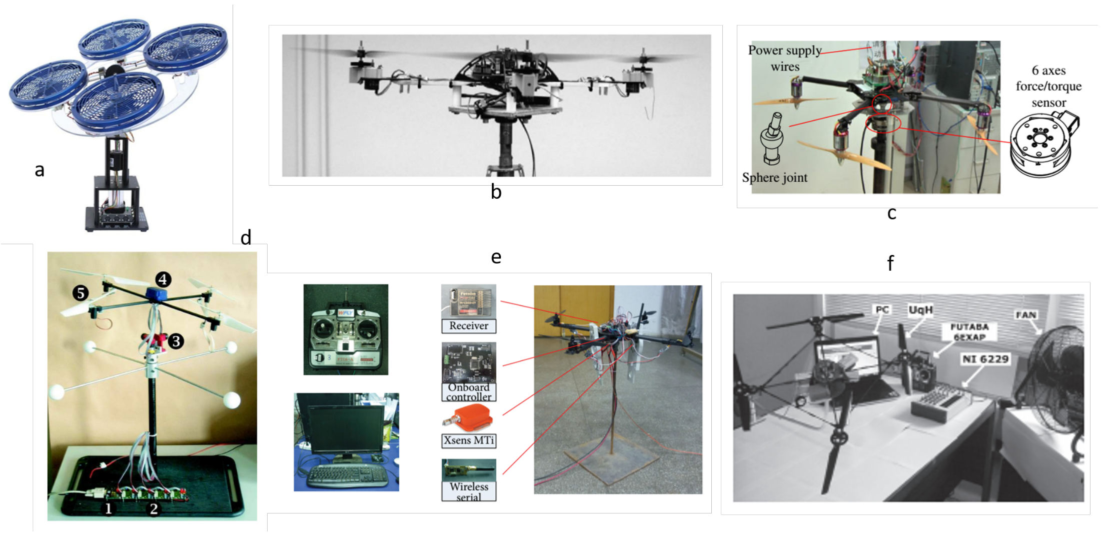

**Figure 1.1** : Quadrotor Test Beds; a [1], b [2], c [3], d [4], e [5], f [6].

testbech. Experimental set up, as seen in Figure 1.1-f, consisted of a fan for wind-gust disturbances and Draganfly Vti quadrotor on board Xsen IMU [6]. Yu and Ding tested and validated 6DOF flight controller on a quadrotor testbed as seen in Figure 1.1-c. Quadrotor vehicle mounted both 6 axes force/torque sensor (3 axes force and 3 axes torque) and a sphere joint to let vehicle rotate in 3 axes. On board they used Xsens MTI IMU sensor for sensor-fusion to experimentally control the vehicle via nonlinear trajectory linearization control technique [3]. An, Li, Wang, and Ma controlled a quadrotor using second order geometric sliding mode Attitude observer on a testbench as seen in Figure 1.1-e with comparing traditional quaternion based sliding mode observer. They used a experimental testbed which composed of wireless reviver, on board controller and Xsens IMU to control of attitude [5]. Bouabdallah, Murrieri and Siegwart used testbench, as seen in Figure 1.1-d, composing of universal joint to enable to rotate 3 axes and Xsens IMU to validate controlling methods [4].

## 1.2 Thesis Outline

**Chapter I** provides literature reviews on the UAVs history, studies about quadrotor controlling, designed experimental testbench for multirotor vehicles, and outline of the thesis.

**Chapter II** presents mathematical modeling of cross frame type quadrotor vehicle with Newton-Euler and Euler-Lagrange methods.

**Chapter III** elaborates on the identification of the quadrotor (IRIS+) properties which are IRIS+ electronic equipment, frame type configuration with dimensions, motor and propeller thrust and drag coefficients, experimental calculation of moments of inertias of quadrotor vehicle, and reference frames. Designing and production of motor-propeller performance testbench and designing, production and mathematical modeling of Hardware In The Loop Aerospace testbed processes are expressed.

**Chapter IV** shows about designing quadrotor flight control principles and evaluation of simulations and implementations results. Design PID and feedback linearization controller both simulation environment and implementation on the Pixhawk autopilot of quadrotor vehicle on the HIL testbed with the methodology of tuning coefficients. In addition, chapter IV presents the analysis of simulation results and the flight data recorded from the Iris+ flight tests about the attitude controller's performance. Finally, conclusion and future research work are elaborated.

# 2. MATHEMATICAL MODELING OF CROSS TYPE QUADROTOR

## 2.1 Mathematical Modeling Of Quadrotor Vehicle

Mathematical modeling of quadrotor is important in order to design and tune controller gains. Modeling approaches [12], [11], [21], [22] and [23] are used for generating the equations of quadrotor motions in this chapter. Generally quadrotors are categorized related to the frame type and defined with the reference coordinate frame. Quadrotors are generally used as a frame of cross or plus configuration as seen in Figure 2.1. In this work, cross frame type quadrotor (IRIS+) vehicle is used for simulation and experimental studies. Cross type quadrotor has basically two advantages over plus configuration: First, motors need less power for pitch and roll action and remaining power can be used for increasing translational motion. Second advantage is providing clear vision for camera with obtuse angle in between front propellers. Frame configurations of cross and plus type quadrotor vehicles and coordinates frames of IRIS+ quadrotor can be illustrated as in Figure 2.1 and 2.2. In order to obtain equations of motion, well adjustment of coordinate frame is needed. The aforementioned frames are inertial, vehicle, and body frames as illustrated in Figure 2.2.

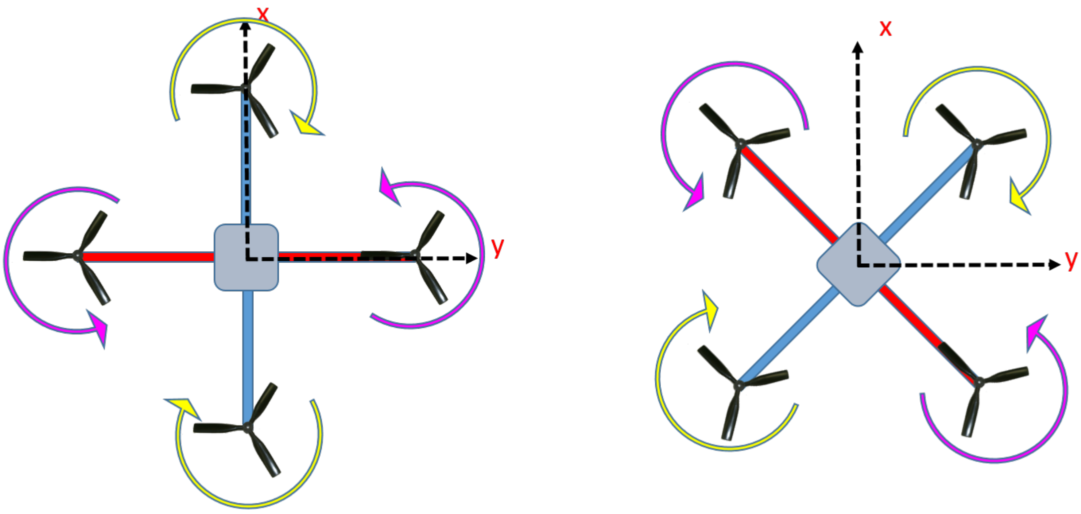

**Figure 2.1** : Plus and cross type of quadrotor vehicle frame configurations.

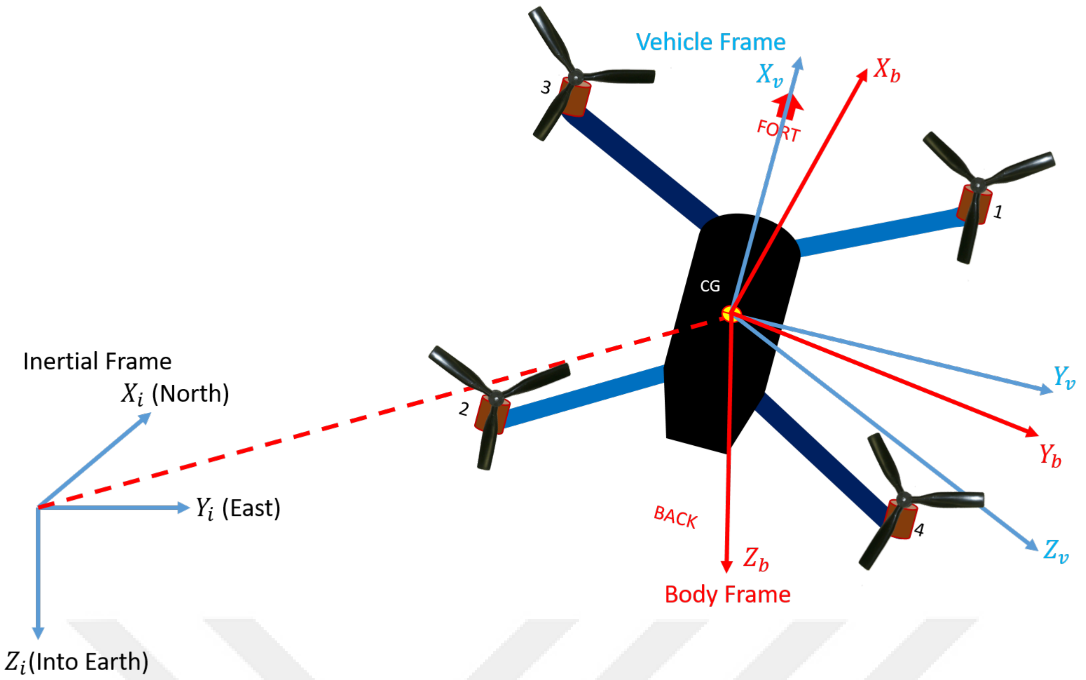

**Figure 2.2** : Coordinate frames of IRIS+ quadrotor.

The inertial frame is earth centered at the vehicle motion starting point and has orthogonal three axes $\xi_i$ =($X_i$, $Y_i$, $Z_i$). Vehicle frame is centered at center of gravity of the quadrotor $\xi_v$ =($X_v$, $Y_v$, $Z_v$). This frame is used for translational motion of the vehicle with respect to inertial frame. The origin of the body coordinate frame, $\xi_b$ =($X_b$, $Y_b$, $Z_b$) is related with rotational motion with respect to vehicle frame at the center of the gravity of the quadrotor. X axis points out to the front of the quadrotor with roll motion *phi* ($\phi$) angle, Y axis always points out from right of the quadrotor with pitch motion *theta* ($\theta$) angle and Z axis always points down from center of the gravity of the quadrotor with yaw motion *psi* ($\psi$) angle.

Quadrotor has two types of motion; linear translational and rotational motion. Quadrotor can be represented as a point mass during translational motion;

$$\begin{aligned}
\vec{\xi}_b &= \begin{bmatrix} X_b & Y_b & Z_b \end{bmatrix}^T \\
\vec{\xi}_i &= \begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix}^T \\
\vec{\eta}_b &= \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T
\end{aligned} \tag{2.1}$$

Linear and angular velocities are as follows:

$$\begin{aligned}\vec{V}_b &= \begin{bmatrix} u & v & w \end{bmatrix}^T \\ \vec{\eta}_b &= \begin{bmatrix} p & q & r \end{bmatrix}^T\end{aligned} \tag{2.2}$$

These vectors are used for deriving the main components of mathematical modeling of the quadrotor vehicle needed to transform from one coordinate frame to another one.

Transformation matrix for Roll motion can be described by;

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \tag{2.3}$$

Transformation matrix for Yaw motion can be described by;

$$R_\psi = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

Transformation matrix for Pitch motion can be described by;

$$R_\theta = \begin{bmatrix} cos\theta & 1 & -sin\theta \\ 0 & 0 & 1 \\ sin\theta & 0 & cos\theta \end{bmatrix} \tag{2.5}$$

Combination of three matrix to Direction Cosine Matrix to transform angular motion from body frame to inertial frame is obtained as (s refers to sine, c refers to cosine, t refers to tan);

$$R_i^b = \begin{bmatrix} s\theta c\psi & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\theta s\psi & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{2.6}$$

$$R^b = R_i^b \cdot R^i \tag{2.7}$$

$$\begin{bmatrix} X^b \\ Y^b \\ Z^b \end{bmatrix} = \begin{bmatrix} s\theta c\psi & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\theta s\psi & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} X^i \\ Y^i \\ Z^i \end{bmatrix} \tag{2.8}$$

Rotational motion matrix for angular velocities in body frame;

$$\omega_b = R_\phi \begin{bmatrix} 0 \\ 0 \\ \dot{\phi} \end{bmatrix} + R_\phi R_\theta \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_\phi R_\theta R_\psi \begin{bmatrix} \dot{\psi} \\ 0 \\ 0 \end{bmatrix} \tag{2.9}$$

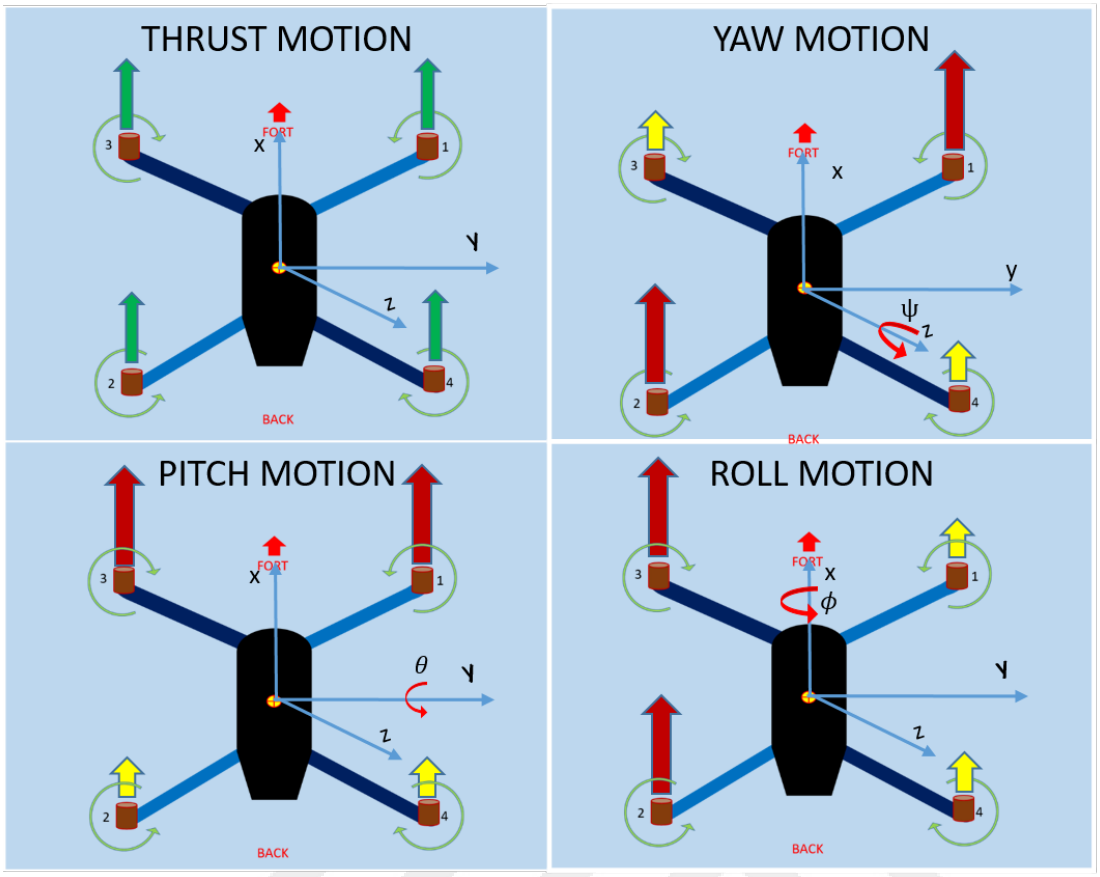

**Figure 2.3** : Motions of quadrotor.

$$\omega_b = \begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 1 & -s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.10}$$

Therefore, transformation of angular velocities from body frame to inertial frame is as follows;

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 1 & s\phi \frac{1}{c\theta} & c\phi \frac{1}{c\theta} \end{bmatrix} \begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} \tag{2.11}$$

After kinematic and dynamic equations of the quadrotor have to be driven, the dynamic equations give translational linear motion information. Linear acceleration to position data is obtained with aerodynamic, gravity and gyroscopic forces. Blade element theory is used to calculation of the aerodynamic forces and moments [24]. Blades are mounting to motor and turning with same angular velocity around motor shaft axes. For aerodynamic force and moment calculation propeller drag and thrust coefficient are used. As seen in the Figure 2.3, there are four flight actions which are thrust, yaw, pitch and roll motions.

Firstly, thrust maneuver is vertical motion with four motor are balanced respect to the center of gravity along the z axis. Total moments of the forces are zero. There are three different missions guided as for thrust maneuver. If total forces are equal to weight, this is hovering maneuver. Other is taking off maneuver occurring while total forces are bigger than total thrust. Last maneuver is landing occurring with total forces are smaller than total weight. The thrust force can be defined as;

$$\begin{aligned} T &= -K_t(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2) \\ T &= -K_t U_1 \end{aligned} \tag{2.12}$$

$K_t$ is thrust coefficient of the propellers. $U_1$ is thrust control input for cross type quadrotors.

Second mission is yaw motion. This motion occours by differences of rational speed between the counter rotating pair of propellers. In the Figure 2.3, first and second motors are turning same direction, third and fourth motors are rotating opposite direction around z axis with yaw angle of $\psi$.

$$\begin{aligned} \tau_\psi &= -K_d(\omega_1^2+\omega_2^2-\omega_3^2-\omega_4^2) \\ \tau_\psi &= -K_d U_4 \end{aligned} \tag{2.13}$$

$K_d$ is drag coefficient of the propellers. $U_4$ is yaw control input for cross type quadrotors.

Third mission is pitch motion turning around y axis with angle of $\theta$. If first and third motors thrusts bigger than second and fourth motors, the quadrotor goes to backward otherwise first and third motors thrusts are smaller than others, quadrotor goes to forward motion.

$$\begin{aligned} \tau_\theta &= -K_t(L_{xf}(\omega_1^2+\omega_3^2)-L_{xb}(\omega_2^2+\omega_4^2)) \\ \tau_\theta &= -K_t U_3 \end{aligned} \tag{2.14}$$

$U_3$ is pitch control input for cross type quadrotors.

Fourth mission is roll motion which quadrotor rotate to right side or left side via roll angle of $\phi$ rotating around x axis. If the desired motion direction is right side, the thrust of left motors third and second thrusts are higher than first and fourth motor thrusts or for left side motion first and fourth motors thrusts are bigger that others.

$$
\begin{aligned}
\tau_\phi &= -K_t(L_{yf}(\omega_3^2 - \omega_1^2) + L_{yb}(\omega_2^2 - \omega_4^3)) \\
\tau_\phi &= -K_t U_2
\end{aligned}
\tag{2.15}
$$

$U_2$ is roll control input for cross type quadrotors. The following matrix shows relationship between control input and angular velocities of each motors with propeller coefficients.

$$
\begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} -K_t & -K_t & -K_t & -K_t \\ K_t L_{yf} & -K_t L_{yb} & -K_t L_{yf} & K_t L_{yb} \\ -K_t L_{xf} & K_t L_{xb} & -K_t L_{xf} & K_t L_{xb} \\ -K_d & -K_d & K_d & K_d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}
\tag{2.16}
$$

The Gravity effect on the quadrotor direction is along to z axis into earth center. The vector notations of Gravity ca be given as follows:

$$
\vec{F_G} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ mg \end{bmatrix}
\tag{2.17}
$$

The gravitational force acting on the quadrotor's center of gravity in the body frame, it needs to be converted by multiplying the rotation matrix with the gravitational force vector in the inertia coordinate frame;

$$
\begin{aligned}
\vec{F_G^b} &= \vec{P_i^b}\vec{F_G^i} \\
\vec{F_G^b} &= \begin{bmatrix} -mgs\theta \\ mgc\theta s\phi \\ mgc\theta c\phi \end{bmatrix}
\end{aligned}
\tag{2.18}
$$

The rotation of propeller and motor rotor generate gyroscopic effect on the quadrotor in body coordinate frame. Gyroscopic force is in Eq. 2.19 ;

$$
\begin{aligned}
&\vec{G} = I_{rotor}\left(\omega_b \times \begin{bmatrix}0\\0\\1\end{bmatrix}\right)\Omega = I_{rotor}\left(\begin{bmatrix}\omega_{bx}\\\omega_{by}\\\omega_{bz}\end{bmatrix} \times \begin{bmatrix}0\\0\\1\end{bmatrix}\right)\Omega \\
&\Omega = \omega_1 + \omega_2 + \omega_3 + \omega_4 \\
&I_{rotor} = \frac{1}{12} m_{prop} L_{prop}^2 + \frac{1}{4} m_{motor} r_{motor}^2
\end{aligned}
\tag{2.19}
$$

### 2.1.1 Euler-Lagrange method

The Euler-Lagrange method is used to model the flight dynamics of the quadrotor [25]. The assumptions for modelling are;

- The structure is rigid and symetric along the X axis
- The origin of body fixed frame coincides with the center of gravity of the quadrotor.
- The propellers has rigid structure and no blade flapping
- The relationship between blade's speed and thrust and drag forces are linear.
- External forces are not considered.
- The hub force, ground effect and gyro effects are not taken into account.

As mentioned in Equation 2.1 translational and rotational motion vectors are structured in Eq. 2.20 for Lagrangian Energy Equation.

$$
\begin{aligned}
q &= [\xi \quad \eta]^T \\
L &= K - P
\end{aligned}
\tag{2.20}
$$

A Lagrangian is obtained by modeling the energy of system, where the difference between kinetic and potential energy is calculated. While potential energy is related only altitude but kinetic energy is related both translational and rotational velocities of the quadrotor. K is refer to kinetic energy and P is refer to potential energy of the system during the flight of quadrotor. The Lagrangian equation is as follows;

$$
\begin{aligned}
L(q,\dot{q}) &= \frac{1}{2}\dot{\xi}^T m \dot{\xi} + \frac{1}{2}\dot{\eta}^T m \dot{\eta} \\
L(q,\dot{q}) &= \frac{1}{2}m(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + \frac{1}{2}m(I_{xx}\dot{\phi}^2 + I_{yy}\dot{\theta}^2 + I_{zz}\dot{\psi}^2) - mgz
\end{aligned}
\tag{2.21}
$$

Moments of inertia of the System is represented with I.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{2.22}$$

The Euler-Lagrange equation eith external generalised force is used [26] in Eq. 2.23.

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = F \tag{2.23}$$

All the external forces acting on he quadrotor is represented with F. F has translational $F_\xi$ and rotational $\tau$ parts.

$$\begin{aligned} F_\xi &= \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^T \\ \tau_\xi &= \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T \\ F &= \begin{bmatrix} F_\xi & \tau \end{bmatrix}^T \end{aligned} \tag{2.24}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \\ I_{xx}\ddot{\phi} + \dot{I_{xx}}\dot{\phi} \\ I_{yy}\ddot{\theta} + \dot{I_{yy}}\dot{\theta} \\ I_z\ddot{\psi} + \dot{I_{yy}}\dot{\psi} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.25}$$

The translational force is;

$$F_\xi = \begin{bmatrix} -mg \cdot s\theta \\ mg \cdot c\theta \cdot s\phi \\ mg \cdot c\theta \cdot c\phi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -K_t \cdot U_1 \end{bmatrix} = \begin{bmatrix} -mg \cdot s\theta \\ mg \cdot c\theta \cdot s\phi \\ mg \cdot c\theta \cdot c\phi - K_t \cdot U_1 \end{bmatrix} \tag{2.26}$$

T is total thrust force as expressed before in Eq. 2.12. $F = m\vec{a}$ is used for getting acceleration information to velocity and position data. Torque is equal to product of moment of inertia and angular acceleration of the body. Torque gives attitude informations $p, q, r$ for designing attitude controller. The gravitational force and motors thrust are used for total force.

Total body moments are;

$$\sum \vec{M}^B = \begin{bmatrix}\tau_\phi & \tau_\theta & \tau_\psi\end{bmatrix}^T$$
$$\sum \vec{M}^B = \begin{bmatrix}-K_t \cdot U_2 \\ -K_t \cdot U_3 \\ -K_d \cdot U_4\end{bmatrix} \tag{2.27}$$

Thus, the Euler-Lagrange equation,

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \\ I_{xx}\ddot{\phi} + \dot{I}_{xx}\dot{\phi} \\ I_{yy}\ddot{\theta} + \dot{I}_{yy}\dot{\theta} \\ I_z\ddot{\psi} + \dot{I}_{yy}\dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ mg \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -mg \cdot s\theta \\ mg \cdot c\theta \cdot s\phi \\ mg \cdot c\theta \cdot c\phi - K_t \cdot U_1 \\ -K_t \cdot U_2 \\ -K_t \cdot U_3 \\ -K_d \cdot U_4 \end{bmatrix} = F \tag{2.28}$$

At hover postion, angular rates can be neglected. Therefore;

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} -g \cdot s\theta \\ g \cdot c\theta \cdot s\phi \\ \frac{1}{m}\left[mg(c\theta \cdot c\phi - 1) - K_t \cdot U_1\right] \end{bmatrix}$$
$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{1}{I_{xx}} K_t \cdot U_2 \\ \frac{1}{I_{yy}} K_t \cdot U_3 \\ \frac{1}{I_{zz}} K_t \cdot U_4 \end{bmatrix} \tag{2.29}$$

### 2.1.2 Newton-Euler method

Forces and moments dynamics equations acting on the quadrotor are expressed in Eqs. 2.12, 2.13,2.14, 2.13 and 2.18. The basic principle of the Newton-Euler Method is;

$$\begin{bmatrix} F^b \\ \tau^b \end{bmatrix} = \begin{bmatrix} mI_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I \end{bmatrix} \begin{bmatrix} \dot{V^b} \\ \ddot{\eta}_b \end{bmatrix} + \begin{bmatrix} \dot{\eta}_b \times (mV^b) \\ \dot{\eta}_b \times (I\dot{\eta}_b) \end{bmatrix} \tag{2.30}$$

Rotational Equations with Newton-Euler Method is in Eq. 2.31.

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ I_{rotor}\Omega \end{bmatrix} = \begin{bmatrix} K_t \cdot U_2 \\ K_t \cdot U_3 \\ K_d \cdot U_4 \end{bmatrix} \tag{2.31}$$

Expanding that, leads to,

$$\begin{bmatrix} I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\phi} \\ I_{zz}\ddot{\phi} \end{bmatrix} + \begin{bmatrix} qI_{zz}r - rI_{zz}q \\ rI_{xx}p - pI_{xx}r \\ pI_{yy}r - rI_{yy}p \end{bmatrix} + \begin{bmatrix} qI_{rotor}\Omega \\ -pI_{rotor}\Omega \\ 0 \end{bmatrix} = \begin{bmatrix} K_t \cdot U_2 \\ K_t \cdot U_3 \\ K_d \cdot U_4 \end{bmatrix} \tag{2.32}$$

Translational equations with Newton-Euler Method is in Eq. 2.33.

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}=\begin{bmatrix}0\\0\\mg\end{bmatrix}+\begin{bmatrix}s\theta c\psi & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi\\ s\theta s\psi & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi\\ -s\theta & c\theta s\phi & c\theta c\phi\end{bmatrix}\begin{bmatrix}0\\0\\-K_t\cdot U_1\end{bmatrix} \tag{2.33}$$

Rotational and translational equations with Newton-Euler Method is in Eq. 2.34.

$$\begin{aligned}
\ddot{x} &= (s\phi s\psi + c\psi s\theta c\phi)\frac{K_t U_1}{m}\\
\ddot{y} &= (-c\psi s\phi + s\psi s\theta c\phi)\frac{K_t U_1}{m}\\
\ddot{z} &= -g + (c\theta c\phi)\frac{K_t U_1}{m}\\
\ddot{\phi} &= \frac{I_{yy}-I_{zz}}{I_{xx}}qr + \frac{K_t U_2}{I_{xx}}\\
\ddot{\theta} &= \frac{I_{zz}-I_{xx}}{I_{yy}}pr + \frac{K_t U_3}{I_{yy}}\\
\ddot{\psi} &= \frac{I_{xx}-I_{yy}}{I_{zz}}qp + \frac{K_d U_4}{I_{zz}}
\end{aligned} \tag{2.34}$$

# 3. HARDWARE IN THE LOOP TEST PLATFORM

Model Based Design (MDB) is a process accelerating with cost-effective prototyping at a short time. MBD process phases are definition of requirement, mathematical modeling of the system (simulation), rapid prototyping, code generation, software in the loop (SIL), processor in the loop (PIL), hardware in the loop (HIL) and validation. First phase is definition of the system or product properties and requirements. The generating mathematical model of the system under boundary conditions effecting the system directly or indirectly in the simulation environment. Then regarding to the mathematical model, rapid prototyping and code generation phase is coming to test the system on the simulations carry out on the computer environment that is the most important point to reduce cost and time consume. Software in the loop simulations are carried out to analyzing generated codes on the host computer. Processor in the loop is analyzing the system on target CPU on the embedded controller to reduction of the processor errors. Hardware-in-the-loop (HIL) simulation is a type of real-time simulation to test designed controller with real plant or system input out put via embedded hardware. HIL simulation shows how controller responds, in real time, to realistic virtual stimuli. HIL can be use to determine if real physical system (plant) model is valid or not.

The Gyroscopic testbed was designed for HIL tests of the quadrotor (IRIS+) for validating and implementing the designed different controller coefficients in real time environments. In this chapter, designing, production and experimental setup processes of the HIL quadrotor vehicle and motor-propeller testbeds were presented.

## 3.1 Motor And Propeller Performance Testbed

Determining the physical parameters of the quadrotor vehicle sometimes needs special testbeds. This parameters are moments of inertia matrix and propeller drag and thrust coefficients. Propeller coefficients are needed to calculate thrust, yaw, pitch and roll

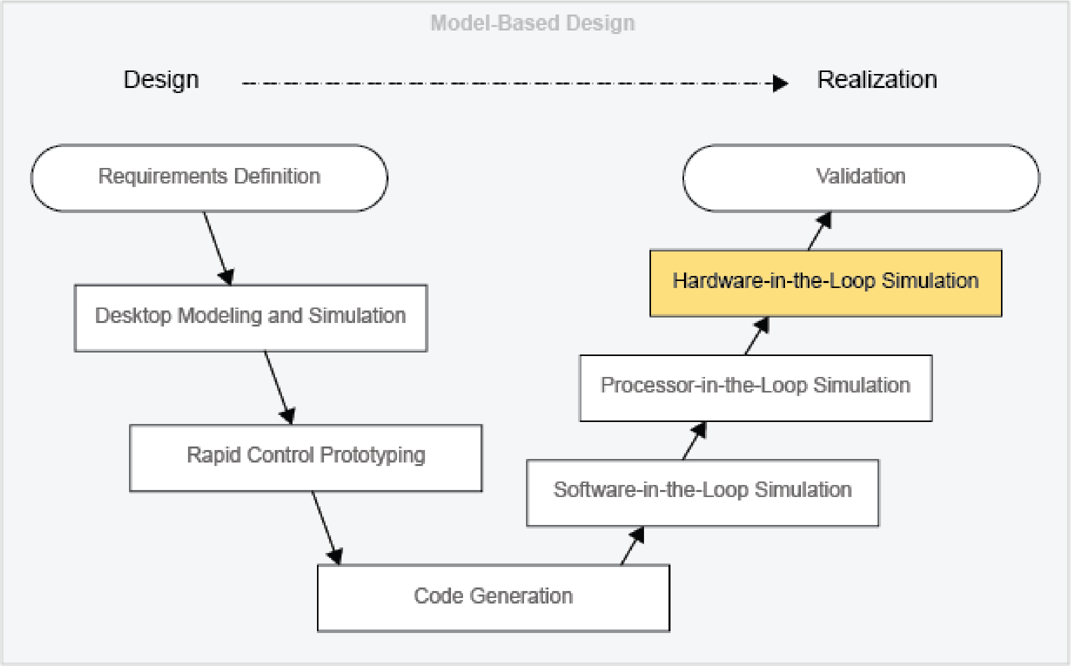
Model-Based Design
Design
Realization
Requirements Definition
Desktop Modeling and Simulation
Rapid Control Prototyping
Code Generation
Software-in-the-Loop Simulation
Processor-in-the-Loop Simulation
Hardware-in-the-Loop Simulation
Validation

**Figure 3.1** : Model based design with Hadware-In-The-Loop Phase [27].

motion as seen in Eqs. 2.12, 2.13, 2.14 and 2.15. So, for the propeller state estimation $K_t$ and $K_d$ are important parameters.

The motor-propeller performance testbed was designed on the CAD, produced with 3D printer assembled with T-motor MN 2213 950 Kv and 9.5 × 4.5 propellers as seen in Figure 3.2. The motor were connected to Lipo Battery via a 20 Amp Electronic Speed Controller (ESC) unit and controlled by Arduino UNO with Pulse-width modulation (PWM) signals.

For experimental setup, the requirements are a scale to measure thrust or drag and a computer to verify the PWM signal generated by potentiometer on the breadboard connected to Arduino UNO board. Th PWM signals were read via program of the Arduino UNO reading the analog outputs of the potentiometer signal and writing to the serial port. The motor generates rotation per minute (RPM) with respect to PWM signal as throttle pedal. RPM values are measured with digital tachometer as seen in Figure 3.2 part e with manually. The trust coefficient $K_t$ and drag coefficient $K_d$ formula are [28] as followings;

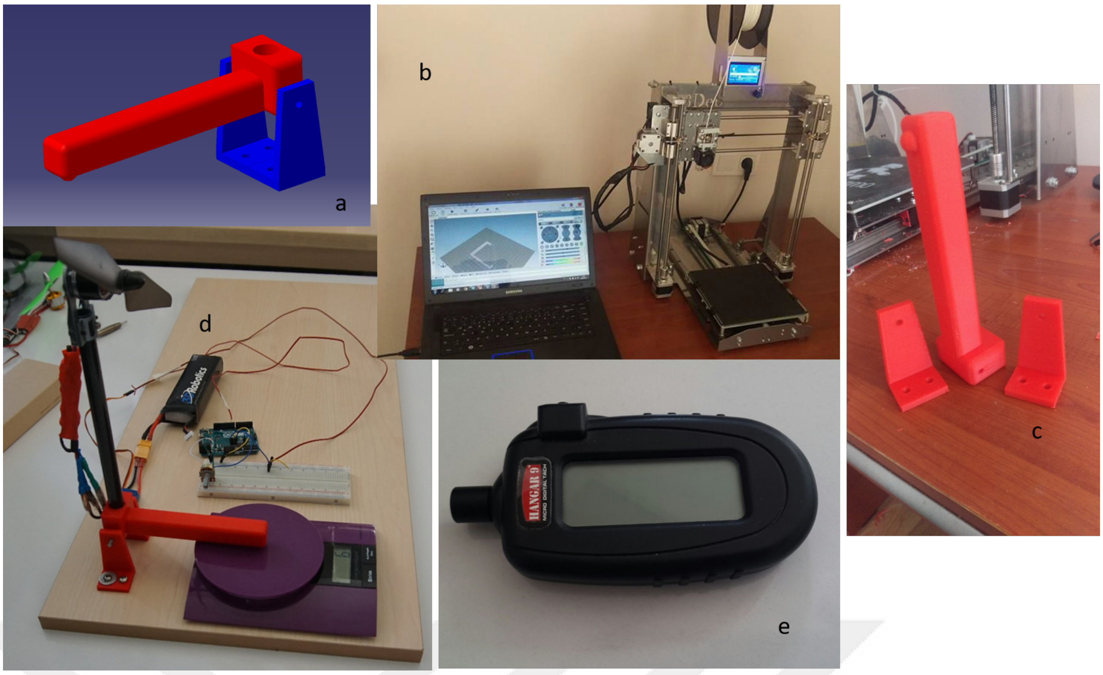

**Figure 3.2** : Motor-propeller performance testbed, a. CAD drawings, b. 3D printer, c. Produced parts, d. Assembly, e Micro digital tachometer.

$$
\begin{aligned}
K_t &= \frac{T}{\rho n^2 d^4} \\
K_d &= \frac{D}{\rho n^2 d^5}
\end{aligned}
\tag{3.1}
$$

n is refer to revolution per-second (RPS), d is diameter of the propeller. So RPM values is converted to RPS values to calculate $K_t$ and $K_d$.

### 3.2 IRIS+ quadrotor vehicle

IRIS+ is a unmanned aerial vehicle that has a capability of Vertical Take-Off and Landing (VTOL) as seen in Figure 3.3. The Iris+ is designed and built by 3D Robotics Company. It can be fitted with a camera to perform aerial photography and mapping with way point navigation ability. The IRIS+ can fly nearly 16 minutes with on board 5100 mAh lipo battery and payload up to 400 g by means of four MN2213 950 kV DC motors. It has Pixhawk Autopilot thats gives opportunity to developer access and develop open source codes. This is the main factor to use for control implementation in this work. Remote controller

The IRIS+ has remote controller with on-screen telemetry for instant or logged data and self-tightening propellers for easy assembly. The Tarot T-2D brushless gimbal

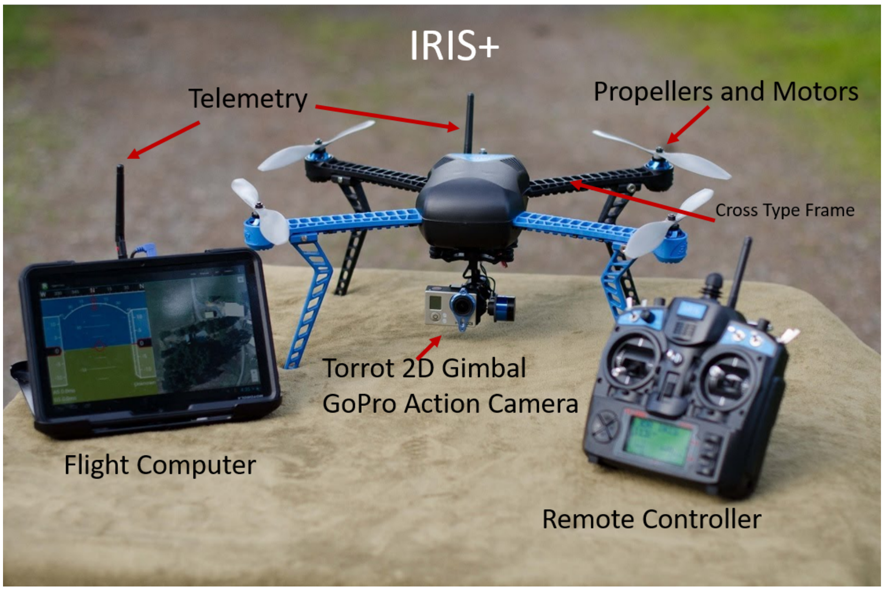

**Figure 3.3** : IRIS+ ready to fly kit [7].

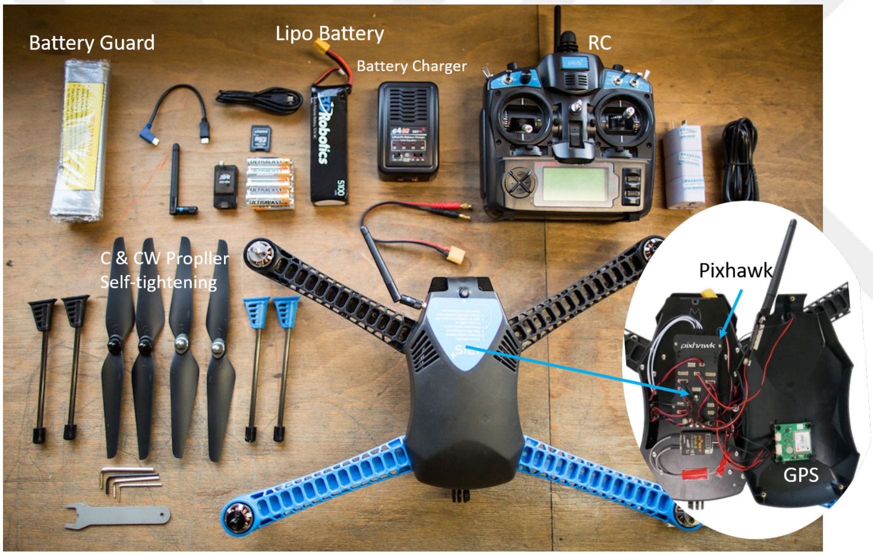

**Figure 3.4** : IRIS+ hardware [7].

uses cutting-edge two-axis stabilization technology to ensure stable video in any flight condition.

### 3.2.1 Pixhawk autopilot

IRIS+ has a Pixhawk Autopilot as a hardware as seen in Figure 3.5. It originated from the Pixhawk Project of the Computer Vision and Geometry Lab of ETH Zurich (Swiss Federal Institute of Technology) and Autonomous Systems Lab. Pixhawk is designed by the open hardware development team in collaboration with 3D Robotics and ArduPilot Group. Pixhawk, targeting at providing autopilot hardware to the academic, hobby and industrial communities with low costs and high availability, is open-hardware project. Pixhawk is suitable for fixed wing, multi rotors, helicopters, cars, boats and any other robotic platform that can move. It is targeted towards high-end research, amateur and industry needs and combines the functionality of the combination PX4FMU with PX4IO. PX4IO is the Input/Output module and PX4FMU is a high-performance autopilot-on-module [29], [30], [8].

Specifications [30]:

- Processor
  - 32bit STM32F427 Cortex M4 core with FPU
  - 168 MHz, 256 KB RAM, 2 MB Flash
  - 32 bit STM32F103 failsafe co-processor
- Sensors
  - MPU 6000 3-axis accelerometer / 3-axis gyroscope
  - MEAS MS5611 barometric pressure sensor
  - ST Micro L3GD20H 16 bit 3-axis gyroscope
  - ST Micro LSM303D 14 bit 3-axis accelerometer / 3-axis magnetometer
- Interfaces
  - PPM sum signal input and RSSI (PWM or voltage) input.
  - 14 PWM / Servo outputs.
  - I2C & SPI & 3.3 and 6.6V ADC inputs.
  - MicroSD card for high-rate logging over extended periods of time.
  - Internal micro USB port and external micro USB port extension.

The main reason of the choosing IRIS+ as a experimental test vehicle is having open-source autopilot Pixhawk. Therefore designed control methodologies can be implement on the real system as desired. The Pixhawk has MicroSD card for high-rate data logging to analyzing the flight performance. It has a small dimensions to use on the micro UAV's.

**Figure 3.5** : Pixhawk autopilot [8].

**Figure 3.6** : IRIS+, T-motor MN 2213 motors and 9.5 × 4.5 propellers (clockwise and counterclockwise).

In the Pixhawk, there is 3DR Robotics firmware but Arducopter firmware is used for experiments. Compiled codes can be uploaded with Mission Planner Software into the processor with Micro USB cable easily. The Mission Planner gives some other facilities which are sensor & ESC calibrations, planning the route as a autonomously way-point navigation, taking the logged flight data from onboard Micro-SD card and flight data related to attitude or altitude on real time.

### 3.2.2 System state estimation

State estimation of the IRIS+ is divided by into two section. First, motor and propeller thrust and drag coefficients are needed for calculation of thrusts and drag of motors. Moments of inertias ($I_{xx}$, $I_{yy}$ and $I_{zz}$) are the other parameters for calculations.

#### 3.2.2.1 Motor and propeller state estimation.

IRIS+ has four 9.5 × 4.5 propellers that diameter is 9.5 inc and four MN 2213 950 kV series motors of T-motor Company as illustrated in Figure 3.6. Two motors and two propellers rotate clockwise(CW) others rotate counterclockwise (CCW).

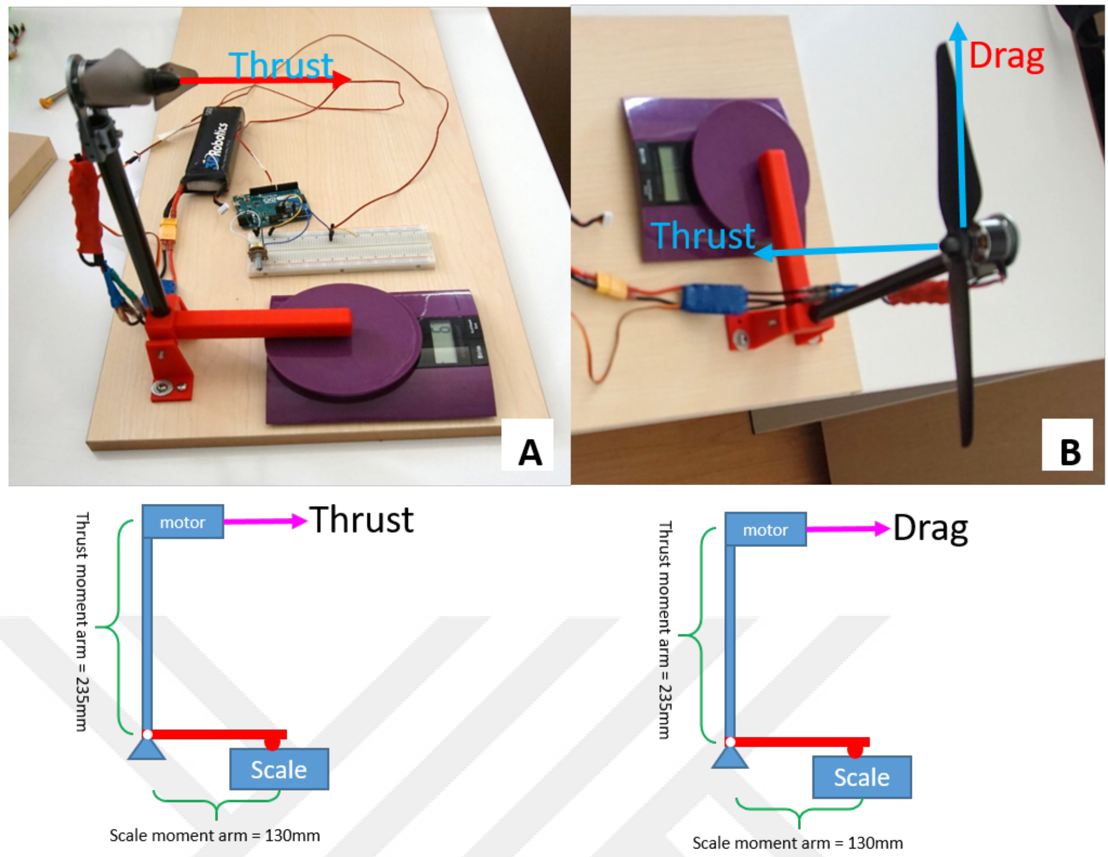

**Figure 3.7** : Thrust and drag measurements.

Measurement of the trust and drag are generated by the propeller. Thrust force is perpendicular to the rotation direction of the motor. The drag force generates moments around the rotation axis as seen in Figure 3.7. Basic formulation is in Eq. 3.2 from [31].

$$\begin{aligned} T \times 235mm &= S \times 130mm \\ D \times 235mm &= S \times 130mm \end{aligned} \tag{3.2}$$

S is refer to scale reaction force.

There was some bottleneck when experimental works was going. Upstream flow is in front of the propeller that air flow encounters it. Downstream flow is back of the propeller that air flow passes it. First experiment, there was a blockage for downstream flow that acting as a ground effect that reducing the thrust nearly fifty percent and generated irritating sounds. Second experiment, the blockage was removed but when the gathering data in front of the propeller, thrust value was changing so upstream flow affects the thrust relatively. This aerodynamic experienced was gained during the motor and propeller coefficients were experimented.

**Table 3.1** : RPM-Thrust, Drag, K_t, K_d

| PWM | RPM | Thrust | Drag | K_{t} | K_{d} |
|---|---|---|---|---|---|
| 1000 | 1354 | 0.000 | 0.000 | 0.000 | 0.000 |
| 1100 | 2009 | 0.315 | 0.025 | 0.068 | 0.023 |
| 1200 | 2665 | 1.153 | 0.091 | 0.141 | 0.046 |
| 1300 | 3320 | 2.088 | 0.168 | 0.164 | 0.055 |
| 1400 | 3975 | 2.978 | 0.229 | 0.163 | 0.052 |
| 1500 | 4631 | 3.830 | 0.284 | **0.155** | **0.048** |
| 1600 | 5286 | 4.850 | 0.376 | 0.150 | 0.048 |
| 1700 | 5941 | 6.157 | 0.503 | 0.151 | 0.051 |
| 1800 | 6596 | 7.440 | 0.589 | 0.148 | 0.049 |
| 1900 | 7252 | 7.895 | 0.635 | 0.130 | 0.043 |
| 2000 | 7907 | 7.866 | 0.614 | 0.109 | 0.035 |

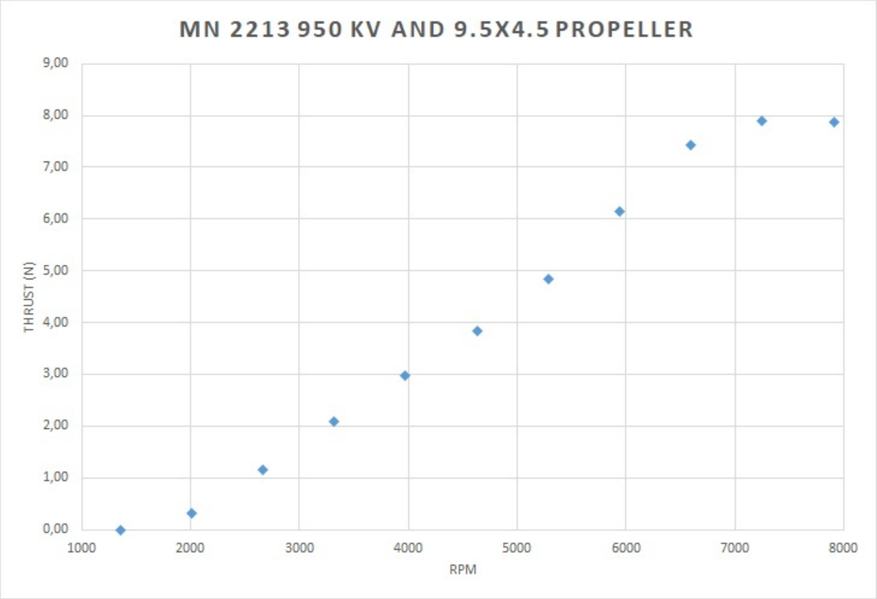

**Figure 3.8** : RPM-Thrust Graphic.

Experiment data consists of PWM, RPM, RPS and thrust (N) or drag (N) values. Trust and drag experiments were done and the results can be seen in Table 3.1.

$K_t$ and $K_d$ are calculated with the Eq. 3.1 as in Table 3.1.

#### 3.2.2.2 Moments of inertia

The moment of inertia of the IRIS+ quadrotor is important for designing the controller. $I_{xx}$, $I_{yy}$ and $I_{zz}$ are calculated analytically and obtained experimentally. With assumption of symmetric geometry of the quadrotor, $I_{xy}$, $I_{xz}$, $I_{yx}$, $I_{yz}$, $I_{zx}$ and $I_{zy}$ are zero.

The analytical method needs weights and lengths of the quadrotor parts as seen in Figure 3.10 and Table 3.2. This properties are measured with steel rule and scale

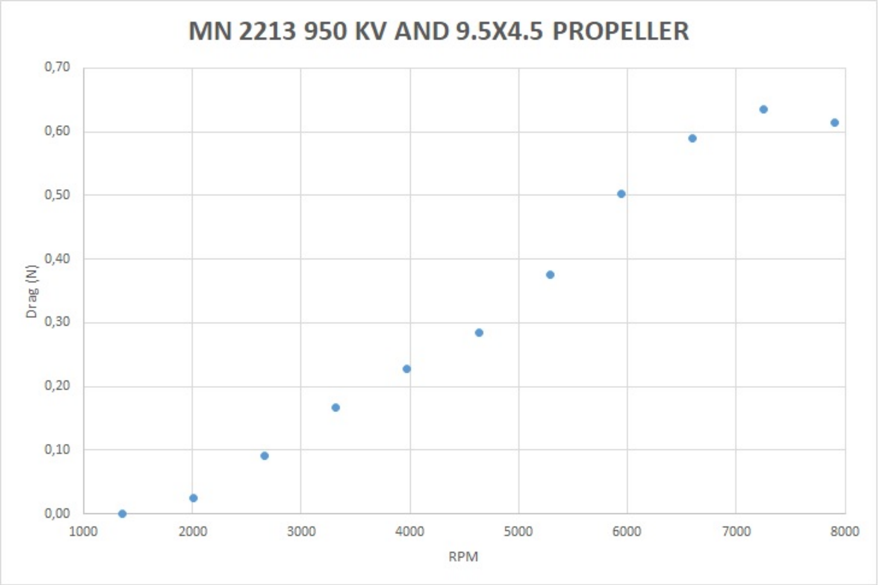

**Figure 3.9** : RPM-Drag Graphic.

having accuracy of 1 gram. Length of the arms is 9 inc ($9 \times 25.4 = 228.6mm$). The angle of the front and back arms are 120° and 140°. Analytically calculations are used from [11];

- $I_{xx} = 0.0238kg.m^2$
- $I_{yy} = 0.00882kg.m^2$
- $I_{zz} = 0.0303kg.m^2$

The obtaining the moments of inertias with experimental is a methodology that chronometer, ruler, scale and some rope are the basic needs. To preparing for the experimental set up, the quadrotor lash down along the desired moment of inertia axis. The ropes have to be lashed down perpendicular to the flat and parallel each others. To find $I_{xx}$ and $I_{yy}$ two rope is enough but for $I_{zz}$ three rope is adequate. Firstly, quadrotor is turned small angle enough to initiate oscillation on the center of mass along the desired axis. After oscillation and chronometer are started same time, one coming and one going is counted one turn. Whenever it is felt enough then the period of the oscillation calculate with time($t$) and turn counter($N$), $T = t/N$. The Eq. 3.3 is used to calculate moment of inertia. $D$ is refer to distance between ropes. $L$ is refer to lenght of ropes. $g$ refer to gravity. $m$ is mass of the quadrotor.

$$I = \frac{m \times g \times D^2 \times T^2}{16 \times \pi^2 \times L} \tag{3.3}$$

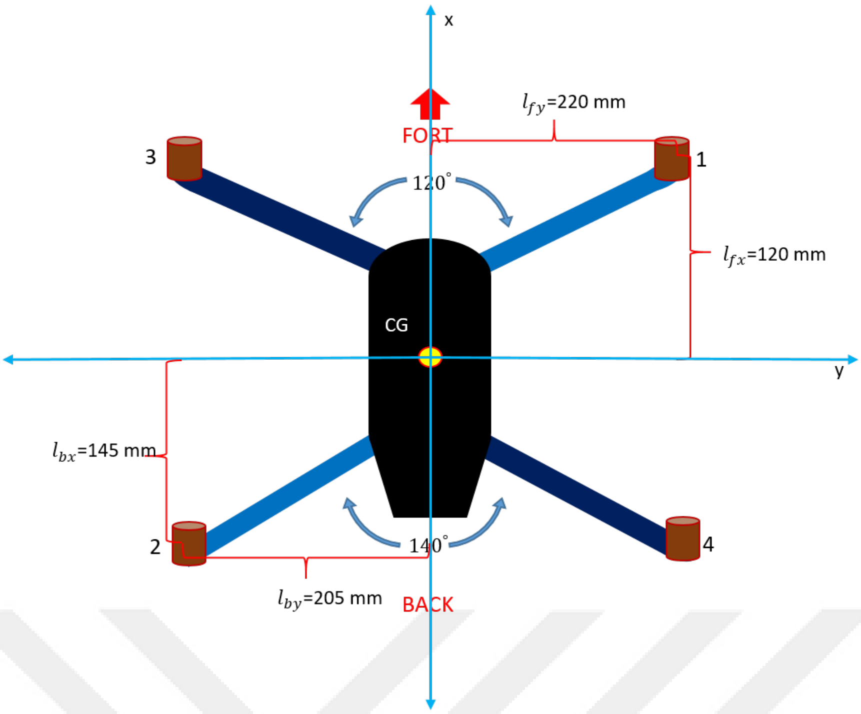

**Figure 3.10 :** Moment of arms.

**Table 3.2 :** Measurement of Quadrotor Physical Specifications.

| Quadrotor Physical Properties | | |
|---|---|---|
| Back Moment Arm Length along x axis | $L_{bx}$ | 145 mm |
| Back Moment Arm Length along y axis | $L_{by}$ | 205 mm |
| Front Moment Arm Length along x axis | $L_{fx}$ | 120 mm |
| Front Moment Arm Length along x axis | $L_{fy}$ | 220 mm |
| Mass of Ready to Fly | $m$ | 1.384 kg |

$I_{xx}$ is in Figure 3.11 and Eq. 3.4 with $N = 50$, $t = 64.52sec$, and $T = 1.2904sec$;

$$\begin{aligned} I_{xx} &= \frac{1.384 \times 9.8065 \times 0.37^2 \times 1.4929^2}{16 \times \pi^2 \times 1.2} \\ I_{xx} &= 0.021838 \quad kgm^2 \end{aligned} \tag{3.4}$$

$I_{yy}$ is in Figure 3.12 and Eq. 3.5 with $N = 50$, $t = 64.52$ sec, and $T = 1.2904$ sec;

$$\begin{aligned} I_{yy} &= \frac{1.384 \times 9.8065 \times 0.3^2 \times 1.2904^2}{16 \times \pi^2 \times 1.17} \\ I_{yy} &= 0.01100 \quad kgm^2 \end{aligned} \tag{3.5}$$

$$\begin{aligned} I_{zz_{ws}} &= \frac{1.384 \times 9.8065 \times 0.42^2 \times 1.56^2}{16 \times \pi^2 \times 1.2} \\ I_{zz_{ws}} &= 0.03074 \quad kgm^2 \\ I_{zz_{support}} &= m_{cb} \times (D_{out}^2 - D_{in}^2) = 0.04 \times (0.46^2 - 0.39^2) = 5.9510^{-4} \quad kgm^2 \\ I_{zz} &= I_{zz_{ws}} - I_{zz_{support}} = 0.030145 \quad kgm^2 \end{aligned} \tag{3.6}$$

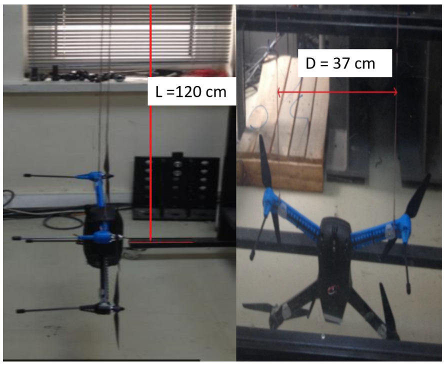

**Figure 3.11** : Experimental setup to $I_{xx}$ of IRIS(+).

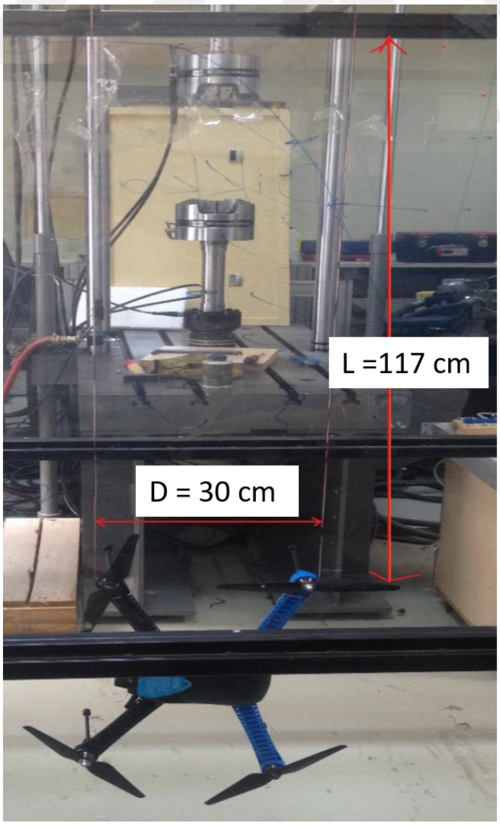

**Figure 3.12** : Experimental setup to $I_{yy}$ of IRIS(+).

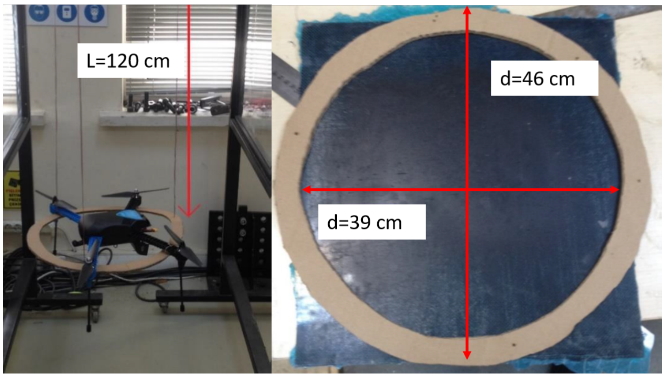

**Figure 3.13** : Experimental setup to $I_{zz}$ of IRIS(+).

**Table 3.3** : IRIS+ Momments of Inertia

| | Experimental | Analytical | Unit |
|---|---|---|---|
| $I_{xx}$ | 0.0218 | 0.0238 | $kg \times m^2$ |
| $I_{yy}$ | 0.0110 | 0.0088 | $kg \times m^2$ |
| $I_{zz}$ | 0.0301 | 0.0303 | $kg \times m^2$ |

$I_{zz}$ is in Figure 3.13 and Eq. 3.6 with $N = 50$, $t = 78sec$, and $T = 1.56sec$. In Figure 3.13, the support part is used for connection of the third rope. After the experiment, th moment of support is extracted from total moment of inertia. $D_{out}$ and $D_{in}$ are refer to outer and inner circle of the support part.When the Experimental and analytical results are compared, $I_{xx}$, $I_{yy}$ and $I_{zz}$ values are enough to close. So experimental results were decided to use on the simulation models.

### 3.3 3DOF Multirotor Test Platform

Validation of the simulation outputs is important before ready to flight. So experimental works are needed to verify how the control signal is reliability and stability. Nonlinear PID and model depended nonlinear feedback linearized controller performance have to be tested on the testbench. Therefore, a new testbench is designed to implement control methods on the quadrotor vehicle with three degree of freedom (3DOF). 3DOF is refer to attitude motions thats are pitch, roll and yaw.

#### 3.3.1 Development of test platform

Designing of the testbed is a process consist of predesign, detailed design, production and assembly of the all the parts to be ready for experimental applications.

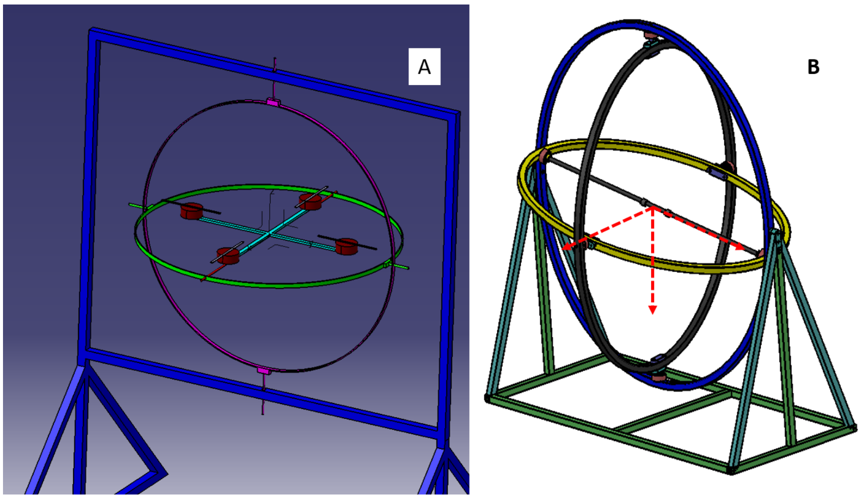

**Figure 3.14** : Preliminary (A) and detailed (B) design of the Gyroscopic HIL testbed.

#### 3.3.1.1 Design

Designing the testbench process consist of predesign and detailed design. During the predesign, literature researching took long time to be informed about existing testbench for quadrotors and satellite attitude determination and control (ADC) testbeds to design best and appropriate one for this work. ADC capability with 3DOF, quick production and usage of cheap materials are most important parameters of the predesign process. Furthermore, the testbed makes changing the different quadrotor or else prototypes possible whenever wanted. There are many kind of testbench but 3DOF testbed having gyroscope working principle as main idea as in literature [32] and [33] were draw my attention. Then predesign drawing was generated in computer aided drafting (CAD) software as illustrated in Figure 3.14.

The next step is detailed design process. In first step, Rotating encoders were wanted to put on each of the degree of freedom but this idea was brought about production difficulties. Then as much as simple and effective testbed was drawn as CAD files as seen in Figured 3.14

#### 3.3.1.2 Production

Main frame, outer circle, inner circle and beams materials were chosen as iron because of cheap, reachable and easily shaping and welding factors. Computer numerical control and iron bending machines were used at production process. Last step of producing was printing all the part with white color. After the assembly hole parts,

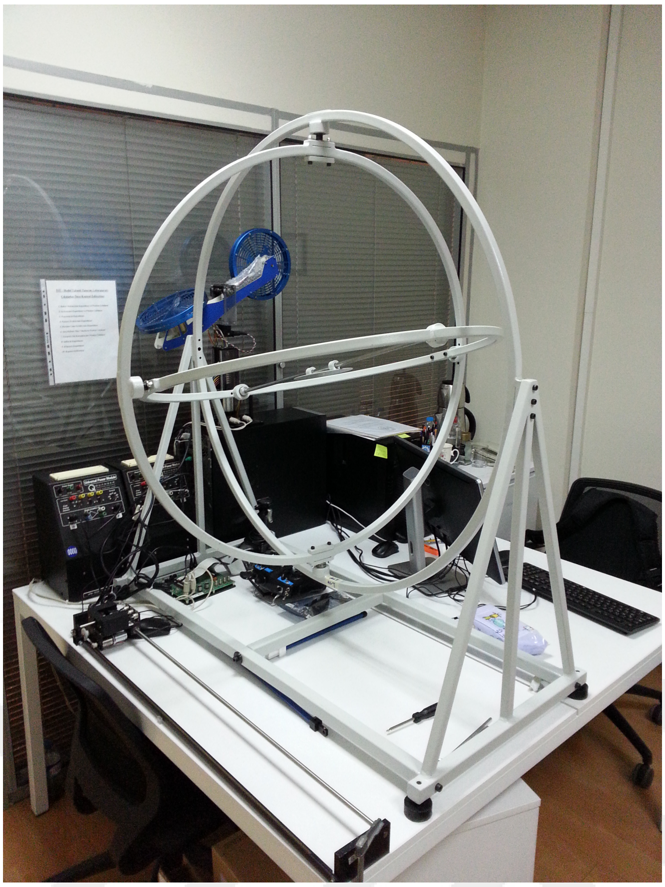

**Figure 3.15** : The Gyroscopic testbed

testbench took place of experimental studies in Model Based Design Laboratory (MBDL) as seen in Figure 3.15.

#### 3.3.1.3 Assembly of quadrotor on the test Platform

Mounting the quadrotor model on the test platform is needed one part design and production. This parts should be at the center of mass of the quadrotor and the testbench and made from light material because of little affects on the quadrotor motion. The support part connecting the quadrotor on testbed was designed as CAD part then produced on the 3D (three dimensional) printer as seen in Figure 3.16.

### 3.3.2 Mathematical modeling of the test platform

The test platform affects the quadrotor motion respectively. So moment of inertia is important factor on the total system. There are two circle and beam on the platform. First circle is outer one that is related with yaw motion. Inner circle is related with pitch motion. The beam was mounted for connection the quadrotor to platform for roll motion. The circle moment of inertia is $I_1 = I_2 = \frac{1}{2}mr^2$ and other axis is $I_3 = mr^2$. $\tau_1$

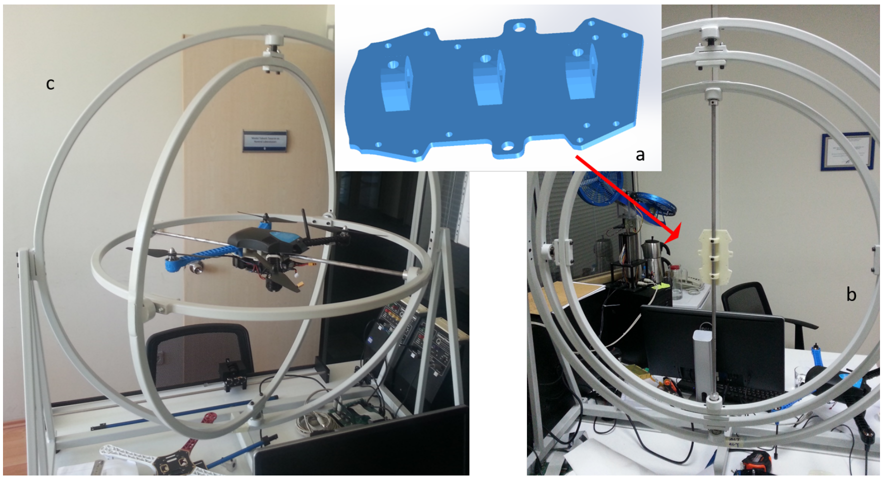

**Figure 3.16** : The support part production, a CAD design, b mounting on testbed, c quadrotor assembly with support on the testbed.

is roll torque, $\tau_2$ is pitch torque, $\tau_3$ is yaw torque. A is refer to outer circle, B is refer to inner circle and beam, and C is refer to quadrotor as seen in Figure 3.17.

$$L_A = \begin{bmatrix} I_{Axx} & 0 & 0 \\ 0 & I_{Ayy} & 0 \\ 0 & 0 & I_{Azz} \end{bmatrix}, L_B = \begin{bmatrix} I_{Bxx} & 0 & 0 \\ 0 & I_{Byy} & 0 \\ 0 & 0 & I_{Bzz} \end{bmatrix}, L_C = \begin{bmatrix} I_{Cxx} & 0 & 0 \\ 0 & I_{Cyy} & 0 \\ 0 & 0 & I_{Czz} \end{bmatrix} \tag{3.7}$$

Calculation of the A and B moments of inertia matrix element;

$$\begin{aligned} I_{Axx} &= I_{Ayy} = \frac{1}{2} m_A r_A^2 \\ I_{Azz} &= m_A r_A^2 \\ I_{Bxx} &= \frac{1}{2} m_B r_B^2 \\ I_{Byy} &= \frac{1}{2} m_B r_B^2 + \frac{1}{2} m_{beam} (2r_B)^2 \\ I_{Bzz} &= m_B r_B^2 + \frac{1}{2} m_{beam} (2r_B)^2 \end{aligned} \tag{3.8}$$

C part moments of inertia was measured with experimentally as shown in table 3.3. $\omega_{AN}$, $\omega_{AN}$, and $\omega_{AN}$ refer to be expressed respectively, angular velocity of A to earth fixed Newtonian frame (N), angular velocity of B to A, and angular velocity of C to B in equation 3.9. a, b, and c are each parts body reference frames. $\omega_1$, $\omega_2$, and $\omega_3$ are angular velocities of each parts.

$$\omega_{AN} = \omega_3 a_3, \qquad \omega_{BA} = \omega_2 a_2, \qquad \omega_{CB} = \omega_1 a_1 \tag{3.9}$$

$$N = R_{AN} a_3, \quad a = R_{BA} b_2 \quad b = R_{CB} c_1 \tag{3.10}$$

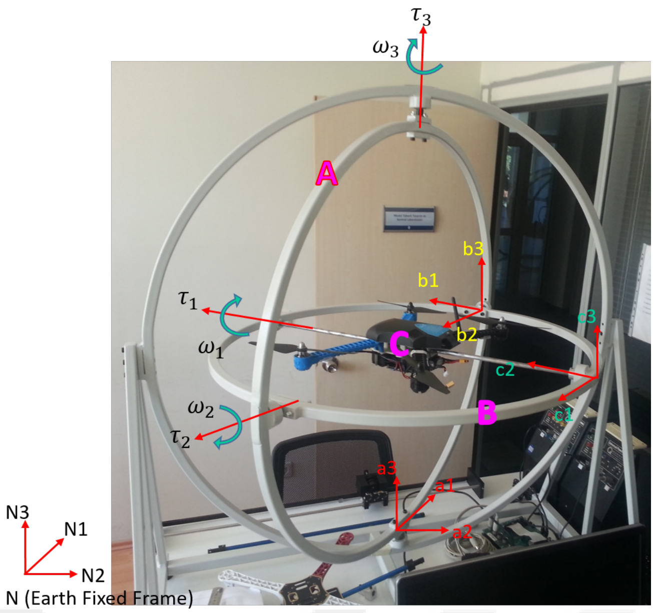

**Figure 3.17** : Testbed is composed of **A** outer circle, **B** inner circle and beam, **C** quadrotor vehicle.

Transformation matrices to other frames are $R_{AN}$, $R_{BA}$, and $R_{CB}$.

$$R_{AN} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_{BA} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \quad R_{CB} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \tag{3.11}$$

The angular velocity ($\omega_{AN}$) of A part relative to the Newtonian frame;

$$\omega_{AN} = \omega_3 a_3 \tag{3.12}$$

The angular velocity ($\omega_{BN}$) of B part relative to the Newtonian frame;

$$\omega_{BN} = \omega_{BA} + \omega_{AN} = \omega_2 b_2 + \omega_3 a_3 \tag{3.13}$$

The angular velocity ($\omega_{CN}$) of C part relative to the Newtonian frame;

$$\omega_{CN} = \omega_x c_1 + \omega_y c_2 + \omega_z c_3 = \dot{\psi} a_3 + \dot{\theta} b_2 + \dot{\phi} c_1 \tag{3.14}$$

$$\begin{gathered} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \Rightarrow b_2 = c_2 c\phi + c_3(-s\phi) \\ \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \Rightarrow a_3 = -b_1 s\theta + b_3 c\theta \\ a_3 = -c_1 s\theta + (c_2 s\phi + c_3 c\phi) c\theta \end{gathered} \tag{3.15}$$

The angular velocities of A and B parts with respect to A, and C part with respect to C as matrix form;

$$\begin{bmatrix}\omega_{Ax}\\ \omega_{Ay}\\ \omega_{Az}\end{bmatrix} = \begin{bmatrix}0\\ 0\\ \dot{\psi}\end{bmatrix}$$
$$\begin{bmatrix}\omega_{Bx}\\ \omega_{By}\\ \omega_{Bz}\end{bmatrix} = \begin{bmatrix}0\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} \tag{3.16}$$
$$\begin{bmatrix}\omega_{Cx}\\ \omega_{Cy}\\ \omega_{Cz}\end{bmatrix} = \begin{bmatrix}1 & 0 & -s\theta\\ 0 & c\phi & s\phi c\theta\\ 0 & -s\phi & c\phi c\theta\end{bmatrix} \begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}$$

Lagrange Kinetic Energy;

$$T = \frac{1}{2}(\omega_{AN}^{A})^{T} I_A \omega_{AN}^{A} + \frac{1}{2}(\omega_{BN}^{A})^{T} I_{B/A} \omega_{BN}^{A} + \frac{1}{2}(\omega_{CN}^{C})^{T} I_C \omega_{CN}^{C} \tag{3.17}$$

$$\omega_{AN}^{A} = \begin{bmatrix}0\\ 0\\ \dot{\psi}\end{bmatrix} \quad \omega_{BN}^{A} = \begin{bmatrix}0\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} \quad \omega_{CN}^{C} = \begin{bmatrix}\dot{\phi} - \dot{\psi} s\theta\\ \dot{\theta} c\phi + \dot{\psi} s\phi c\theta\\ -\dot{\theta} s\phi + \dot{\psi} c\phi c\theta\end{bmatrix} \tag{3.18}$$

The B part moment of inertia transformation $I_{B/A}$ to A frame is;

$$\begin{aligned} I_{B/A} &= R_{BA} I_B R_{BA}^{T}\\ &= \begin{bmatrix}c\theta & 0 & s\theta\\ 0 & 1 & 0\\ -s\theta & 0 & c\theta\end{bmatrix} \begin{bmatrix}I_{Bxx}c\theta & 0 & -I_{Byy}s\theta\\ 0 & I_{Byy} & 0\\ I_{Bzz}s\theta & 0 & I_{Bzz}c\theta\end{bmatrix}\\ &= \begin{bmatrix}I_{Bxx}c^2\theta + I_{Bzz}s^2\theta & 0 & (I_{Bzz} - IByy)s\theta c\theta\\ 0 & I_{Byy} & 0\\ (I_{Bzz} - I_{Bxx})s\theta c\theta & 0 & I_{Bxx}s^2\theta + I_{Bzz}c^2\theta\end{bmatrix} \end{aligned} \tag{3.19}$$

The total kinectic energy of the system is $L =$ Kinetic Energy of A Body + Kinetic Energy of B Body + Kinetic Energy of C Body.

$$\begin{aligned} L = \frac{1}{2} I_{Azz}\dot{\psi}^2 &+ \frac{1}{2}\left[I_{Byy}\dot{\theta}^2 + (I_{Bxx}s^2\theta + I_{Bzz}c^2\theta)\dot{\psi}^2\right]\\ &+ \frac{1}{2}\left[\dot{\phi} - \dot{\psi}s\theta \quad \dot{\theta}c\phi + \dot{\psi}s\phi c\theta \quad -\dot{\theta}s\phi + \dot{\psi}c\phi c\theta\right]\\ &\begin{bmatrix}I_{Cxx} & 0 & 0\\ 0 & I_{Cyy} & 0\\ 0 & 0 & I_{Czz}\end{bmatrix} \begin{bmatrix}\dot{\phi} - \dot{\psi}s\theta\\ \dot{\theta}c\phi + \dot{\psi}s\phi c\theta\\ -\dot{\theta}s\phi + \dot{\psi}c\phi c\theta\end{bmatrix} \end{aligned} \tag{3.20}$$

Lagrangian $L(\theta_i, \dot{\theta}_i)$;

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} = T_i \qquad i = 1,2,3,... \tag{3.21}$$

$i = 1$ is Lagrangian of Roll action T1;

$$T_1 = \frac{d}{dt}\frac{\partial L}{\partial \dot{\phi}} - \frac{\partial L}{\partial \phi} \tag{3.22}$$

$$\begin{aligned} T_1 = {} & \ddot{\phi} I_{Cxx} - \ddot{\psi} s\theta I_{Cxx} - \dot{\theta}\dot{\psi} c\theta I_{Cxx} - (I_{Czz} - I_{Cyy})(s\phi c\phi)(\dot{\psi}^2 c^2\theta - \dot{\theta}^2) \\ & + (I_{Czz} c\theta - I_{Cxx} s\theta)(\dot{\theta}\dot{\psi}(c^2\phi - s^2\phi)) \end{aligned} \tag{3.23}$$

$i = 2$ is Lagrangian of pitch action T2;

$$T_2 = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} \tag{3.24}$$

$$\begin{aligned} T_2 = {} & \ddot{\phi} c\phi s\phi c\theta (I_{Cyy} - I_{Czz}) + \ddot{\theta}(I_{Cyy} c^2\phi + I_{Czz} s^2\phi + I_{Byy}) \\ & + (I_{Cyy} - I_{Czz})\left[\dot{\psi}(c\theta\dot{\phi}(c^2\phi - s^2\phi) - \dot{\theta} c\phi s\phi s\theta) - 2\dot{\theta}\dot{\phi} s\phi c\phi + \dot{\theta}\dot{\psi} c\phi s\phi s\theta\right] \\ & + I_{Cxx}\dot{\phi}\dot{\psi} c\theta + \dot{\psi}^2 s\theta c\theta (I_{Bzz} - I_{Bxx} - I_{Cxx}) + (I_{Cyy} s^2\phi + I_{Czz} c^2\phi)(\dot{\psi}^2 c\theta s\theta) \end{aligned} \tag{3.25}$$

$i = 3$ is Lagrangian of yaw action T3;

$$T_3 = \frac{d}{dt}\frac{\partial L}{\partial \dot{\psi}} - \frac{\partial L}{\partial \psi} \tag{3.26}$$

$$\begin{aligned} T_3 = {} & -\ddot{\phi} I_{Cxx} s\theta + \ddot{\theta} c\phi s\phi c\theta (I_{Cyy} + I_{Czz}) \\ & + \ddot{\psi}(I_{Cxx} s^{\theta} I_{Azz} + I_{Bxx} s^2\theta + I_{Bzz} c^2\theta + I_{Cyy} s^2\phi c^2\theta + I_{Czz} c^2\phi c^2\theta) \\ & + (I_{Cyy} + I_{Czz})(\dot{\theta}(\dot{\phi} c^2\phi c\theta - \dot{\phi} s^2\phi c\theta - \dot{\theta} c\phi s\phi s\theta)) + (I_C yy - I_C zz)(\dot{\psi}\dot{\phi} s2\theta c^2\theta) \\ & - (\dot{\theta} s2\theta)(I_{Cyy} s^2\phi + I_{Czz} c^2\phi) + \dot{\phi}\dot{\theta} s2\theta (I_{Cxx} + I_{Bxx} + I_{Bzz}) - I_{Cxx}\dot{\phi}\dot{\theta} c\theta \end{aligned} \tag{3.27}$$

# 4. DESIGNING FLIGHT CONTROLLER FOR SIMULATION AND IMPLEMENTATION

Both quadrotor vehicle (IRIS+) and HIL testbed are dynamically modeled up to this section. The experimental identification of the quadrotor is done to run simulations and implement on the HIL testbench via apply different type of control methodologies. Main aim of this chapter is comparison PID and Feedback Linearization methods in both simulation and real time response of the quadrotor motions in the HIL testbed. This chapter compose of two main part. First is related with Matlab Simulink environment to run the nonlinear system models on two control methods. Second is about to implementation of the designed PID and Feedback Linearized controller real time C++ codes on the Pixhawk autopilot on the testbed and gathering the logged real flight data with manual tunning gains of the controllers.

There are two types of control which are Open Loop and Closed-loop control. Main idea of the open loop control is controlling the system or plant disregarding the system outputs in contrast to closed-loop. Closed-loop control recycles the output of system to optimize the control signal to reach desired conditions as illustrated in Figure 4.1. The important part of the closed-loop system is sensor, a electronic device to logging data related to aim, to be informed about the system outputs.

The main control arhitecture of this work is based on angel control via measured or simulated feedback signals. For attitude control of the quadrotor, HIL testbed was designed and simulations are run due to this perspective.

In this work, Pixhawk, controller of the system as hardware on board different types of sensors, controls the quadrotor system via designed control algorithm and sensors data to simulate HIL and validation. The Pixhawk runs the NuttX Shell (NSH) real time operating system with Arupilot open-source algorithms as software. At second part of this chapter, open source of the autopilot is changed as desired to implement designed controller methods. On the HIL testbed, controller parameters are manually set to reach optimal outputs.

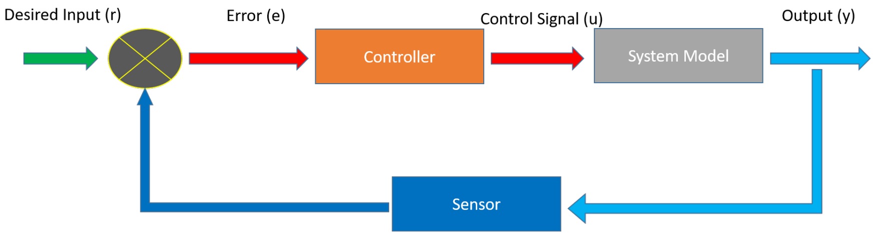

**Figure 4.1** : Closed-Loop control schema.

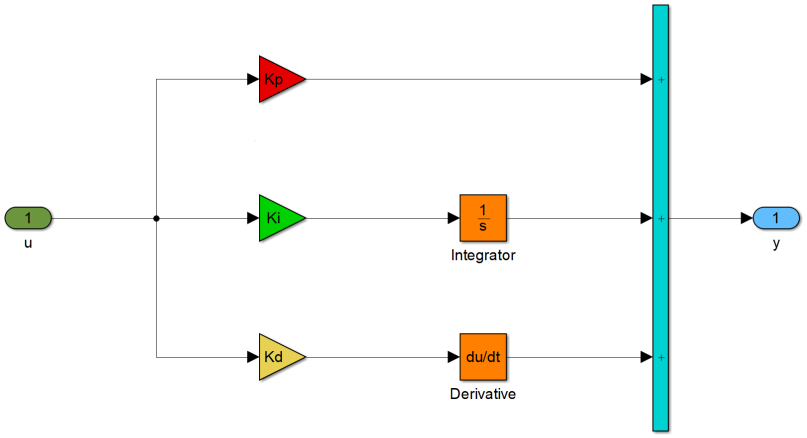

**Figure 4.2** : General architecture of PID controller schema.

## 4.1 PID Controller

PID (Proportional-Integral-Derivative) controller is currently most used closed-loop (feedback) controller in industry applications. Besides, simple structure and easily tunning the parameters and enabling efficiently and relatively robust control capabilities with regard to Modern Control Methods are attracting ways to be most in usage. It is currently been used to design autopilot for UAVs as a starting point for private companies and researchers.

Main structure of PID controller is shown as illustrated in Figure 4.2 and formulated in Eq. 4.1. P component $K_p$ is mostly relevant with the dominant response of the system. Increasing the $K_p$ coefficient generally cause to decrease rise times and setling time, but also larger overshoots and increase osciallatory or unstable behavior. Derivative component $K_d$ responds to the rate of change of the error signal and is mostly relevant with shaping the damping behavior of the closed-loop system. Increasing the $K_d$ coefficient, generally cause to decrease overshoot and a better damped behavior, but also to increase steady-state errors. Integral component $K_i$ is typically optimize the steady-state response of the system and shape its dynamic behavior. Increasing the

$K_i$, cause to reduction of the steady-state error (often elimination) but also increase oscillations. PID components have couple relationship in each-other. Therefore PID tuning is a highly coupled and iterative procedure [34].

$$u(t) = K_p e(t) + Ki \int_t^0 e(\tau) d\tau + K_d \frac{d}{dt} e(t) \tag{4.1}$$

## 4.2 Feedback Linearization

Control of nonlinear system has different type of methods. Feedback Linearization is one of the solution which main principle based on using system real outputs to design a controller to eliminate nonlinear terms in the system. In other words, designing controller via feedbacks to force the system to act as desired with nonlinear pole-placement.

A nonlinear system described by in Eq. 4.2;

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= h(x) \end{aligned} \tag{4.2}$$

There are two variations of the transformation to linearization. In the first, $z = \Phi(x)$, for which $\Phi^{-1}(z)$ exist with $\frac{d\Phi(x)}{dx}$ have to be continuous and a transformation $\Psi$ with $V = \Psi(x,u)$, also invertible for control signal $u = \Phi^{-1}(x,v)$, such that [35]

$$\begin{aligned} \dot{z} &= \frac{d\Phi(x)}{dx} f(x,u)|_{x \to \Phi^{-1}(z), \quad u \to \Psi^{-1}(\Phi^{-1}(z),v)} \\ &= Az + Bv \end{aligned} \tag{4.3}$$

Then the system is called feedback linearizable, or specially input-state feedback linearizable. When such a transformation can be found, we can design $v(k)$ and then generate $u = \Psi^{-1}(x,v)$. $v$ is refer to virtual input to generate control signal $u$.

In this work to control the system ;

$$\begin{aligned} x_1 &= \theta \\ \dot{x_1} &= x_2 = \dot{\theta} \\ \dot{x_2} &= \ddot{\theta} = v \end{aligned} \tag{4.4}$$

by designing of virtual imput $v$,

$$\begin{aligned} v = \ddot{\theta} &= k_1\theta + k_2\dot{\theta} \\ &= -2\lambda_1\dot{\theta} - \lambda_1^2\theta \end{aligned} \tag{4.5}$$

where the constant $\lambda_1$ is chosen as same root at left side of the pole-zero map to pole placement of the system. In Chapter 3, The Lagrangian Eqs. 3.23, 3.25, and 3.27 are generated to feedback linearized controller (FLC).

For roll, pitch and yaw action virtual inputs are;

$$\begin{aligned} v_\phi &= \ddot{\phi} = -2\lambda_\phi\dot{\phi} - \lambda_\phi^2\phi \\ v_\theta &= \ddot{\theta} = -2\lambda_\theta\dot{\theta} - \lambda_\theta^2\theta \\ v_\psi &= \ddot{\psi} = -2\lambda_\psi\dot{\psi} - \lambda_\psi^2\psi \end{aligned} \tag{4.6}$$

Then the control signals are transformed into;

For Roll Motion;

$$\begin{aligned} u_\phi &= T_1 \\ &= v_\phi I_{Cxx} - v_\psi s\theta I_{Cxx} - \dot{\theta}\dot{\psi}c\theta I_{Cxx} - (I_{Czz} - I_{Cyy})(s\phi c\phi)(\dot{\psi}^2 c^2\theta - \dot{\theta}^2) \\ &\quad + (I_{Czz}c\theta - I_{Cxx}s\theta)(\dot{\theta}\dot{\psi}(c^2\phi - s^2\phi)) \end{aligned} \tag{4.7}$$

For Pitch Motion;

$$\begin{aligned} u_\theta &= T_2 \\ &= v_\phi c\phi s\phi c\theta(I_{Cyy} - I_{Czz}) + v_\theta(I_{Cyy}c^2\phi + I_{Czz}s^2\phi + I_{Byy}) \\ &\quad + (I_{Cyy} - I_{Czz})\left[\dot{\psi}(c\theta\dot{\phi}(c^2\phi - s^2\phi) - \dot{\theta}c\phi s\phi s\theta) - 2\dot{\theta}\dot{\phi}s\phi c\phi + \dot{\theta}\dot{\psi}c\phi s\phi s\theta\right] \\ &\quad + I_{Cxx}\dot{\phi}\dot{\psi}c\theta + \dot{\psi}^2 s\theta c\theta(I_{Bzz} - I_{Bxx} - I_{Cxx}) + (I_{Cyy}s^2\phi + I_{Czz}c^2\phi)(\dot{\psi}^2 c\theta s\theta) \end{aligned} \tag{4.8}$$

For Yaw Motion;

$$\begin{aligned} u_\psi &= T_3 \\ &= -v_\phi I_{Cxx}s\theta + v_\theta c\phi s\phi c\theta(I_{Cyy} + I_{Czz}) \\ &\quad + v_\psi(I_{Cxx}s^\theta I_{Azz} + I_{Bxx}s^2\theta + I_{Bzz}c^2\theta + I_{Cyy}s^2\phi c^2\theta + I_{Czz}c^2\phi c^2\theta) \\ &\quad + (I_{Cyy} + I_{Czz})(\dot{\theta}(\dot{\phi}c^2\phi c\theta - \dot{\phi}s^2\phi c\theta - \dot{\theta}c\phi s\phi s\theta)) + (I_Cyy - I_Czz)(\dot{\psi}\dot{\phi}s2\theta c^2\theta) \\ &\quad - (\dot{\theta}s2\theta)(I_{Cyy}s^2\phi + I_{Czz}c^2\phi) + \dot{\phi}\dot{\theta}s2\theta(I_{Cxx} + I_{Bxx} + I_{Bzz}) - I_{Cxx}\dot{\phi}\dot{\theta}c\theta \end{aligned} \tag{4.9}$$

The Eqs. 4.7, 4.8, and 4.9 are both simulated in Matlab and implemented on the Pixhawk autopilot as seen in Figure4.3.

## 4.3 Simulation Results

Control architecture of quadrotor vehicle in simulation environment is illustrated in Figure 4.3. There is a input to attitude controller Block which includes both desired $\phi$, $\theta$, $\psi$ angles and $Z$ position and outputs of feedback which are measured angles $\phi$, $\theta$, $\psi$, and measured $X$, $Y$ and $Z$ positions. In Figure 4.3, A is refer to simulations of full nonlinear model of quadrotor (IRIS+) with PID and FLC methods. B is refer to simulation of full nonlinear of HIL testbed model with FLC method.

### 4.3.1 PID controller simulation result

In this work, Matlab Control System Toolbox is used for $K_p$, $K_i$, and $K_d$ parameter of PID tuning for simulation of nonlinear model of quadrotor. During tuning each coefficient, others parameters values are get zero values to eliminate their effect on the tunned parameter. PID coefficients of quadrotor in simulation environment is shown on the Table 4.1.

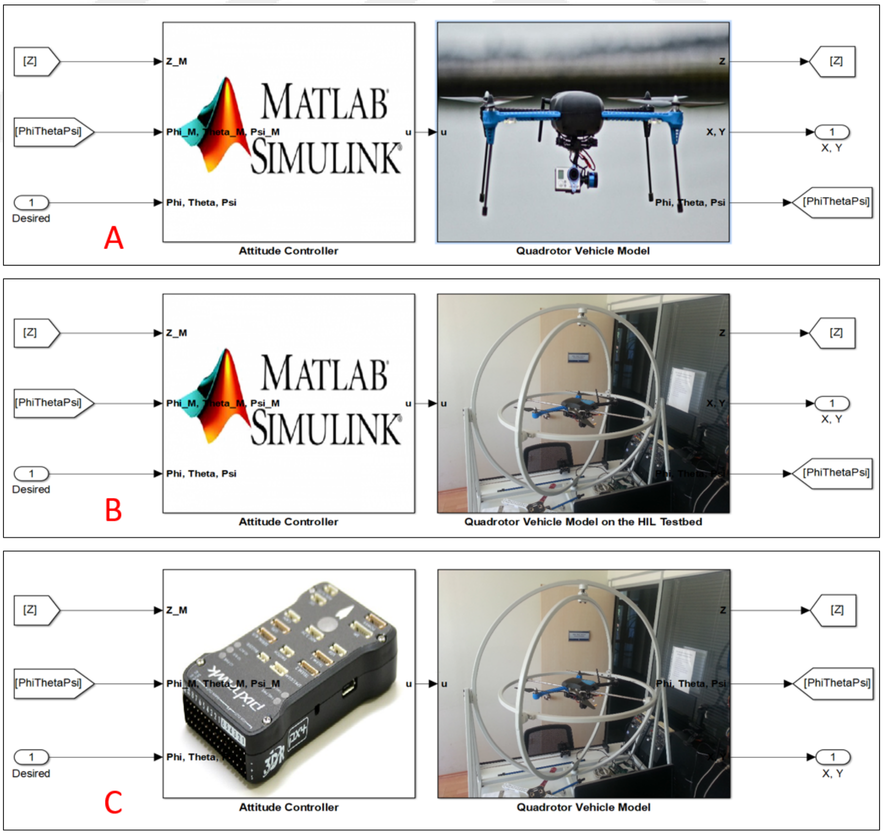

**Figure 4.3** : 3 Kind of control architecture: A. Simulation of nonlinear quadrotor model, B. Simulation of quadrotor with nonlinear testbed model, C. Real time control of quadrotor on the testbed.

**Table 4.1** : PID coefficient of quadrotor in simulation environment.

| | **Roll Input** $U_2$ | **Pitch Input** $U_3$ | **Yaw Input** $U_4$ | **Z Input** $U_1$ |
|---|---|---|---|---|
| **K_p** | 0.27 | 0.169 | 1.106 | 239 |
| **K_i** | 0.039 | 0.026 | 0.107 | 20.14 |
| **K_d** | 0.43 | 0.245 | 2.286 | 100.25 |
| **Filter coefficient (N)** | 5.19 | 3.76 | 3.76 | 10 |

Inputs of the simulation are set 20° for $\phi$ and $\theta$ angles but $\psi$ is set 10° degree as seen in Figure 4.4. Input of Z position is $5m$. Blue line of the graphics refer to desired or inputs values and red lines refer to measured or outputs of the simulation. Both of the graphics have smaller than one second of rise time and 30% percent overshoot and 4 second settling time and have ignorable steady state errors. As seen in table 4.1, PID coefficients of Z position is bigger than other parameters coefficients relatively and is affected the system with saturation understood by solid line. The nonlinearities of the system causes termination of simulation in short time. In shortly, the results shows that controlling of the quadrotor via PID controller is possible.

### 4.3.2 Feedback linearized nonlinear controller simulation results

As mentioned above, there are two simulation via FLC. First simulation is about controlling the nonlinear quadrotor model via FLC. Inputs of the simulation are set 20° for $\phi$ and $\theta$ angles but $\psi$ is set 10° degree as seen in Figure 4.5. Input of Z position is $5m$. $\lambda_\phi = 8.3$, $\lambda_\theta U 8.4$, $\lambda_\psi = 8.3$, and $\lambda_Z = 9$ are tuned as same method in PID tuning. Blue line of the graphics refer to desired or inputs values and red lines refer to measured or outputs of the simulation. Both of the graphics have smaller than one second of rise time and 30% percent overshoot and 3.5 second settling time and have ignorable steady state errors. But Z position have negative steady-state error. In shortly, the results shows that controlling of the quadrotor via FLC controller is possible.

Secondly, nonlinear HIL testbed with quadrotor vehicle model is controlled via FLC. During the simulation seven different $\lambda_\phi$, $\lambda_\theta$, and $\lambda_\psi$ are tried as seen in table 4.2. Inputs of the simulation are set 20° for $\phi$ and $\theta$ angles but $\psi$ is set 10° degree as seen in Figures 4.6, 4.7, and 4.8.

**Table 4.2** : Seven type of $\lambda_\phi$, $\lambda_\theta$, and $\lambda_\psi$ in HIL simulation.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $\lambda_\phi$ | 5 | 8 | 11 | **13** | 15 | 17 | 20 |
| $\lambda_\theta$ | 1.5 | 2.5 | 3 | **3.5** | 4 | 5 | 7 |
| $\lambda_\psi$ | 6 | 7.5 | 9 | **10** | 11 | 12.5 | 14 |

In Figures 4.6 and 4.7, time responses of the pitch and roll angles $\theta = 20^\circ$, $\phi = 20^\circ$ are set with $\lambda_\psi = 0^\circ$ via seven different $\lambda_\theta$ and $\lambda_\phi$ values. Small values of $\lambda_\theta$ and $\lambda_\phi$ behave as slowly response with no overshoot and ignorable steady state error in contrast to bigger values of $\lambda_\theta$ and $\lambda_\phi$. Finally, $\lambda_\theta$ and $\lambda_\phi$ are set to 3.5 and 13 value to minimum overshoot, steady-state error and and settling time.

In Figure 4.8, time response of the pitch angle $\psi = 10^\circ$ is shown with zero values of $\phi$ and $\theta$ via seven different $\lambda_\psi$ values. Small values of $\lambda_\psi$ behaves as slowly response with no overshoot and ignorable steady state error in contrast to bigger values of $\lambda_\psi$. Finally, $\lambda_\psi$ is set to 10 value to minimum overshoot, steady-state error and and settling time.

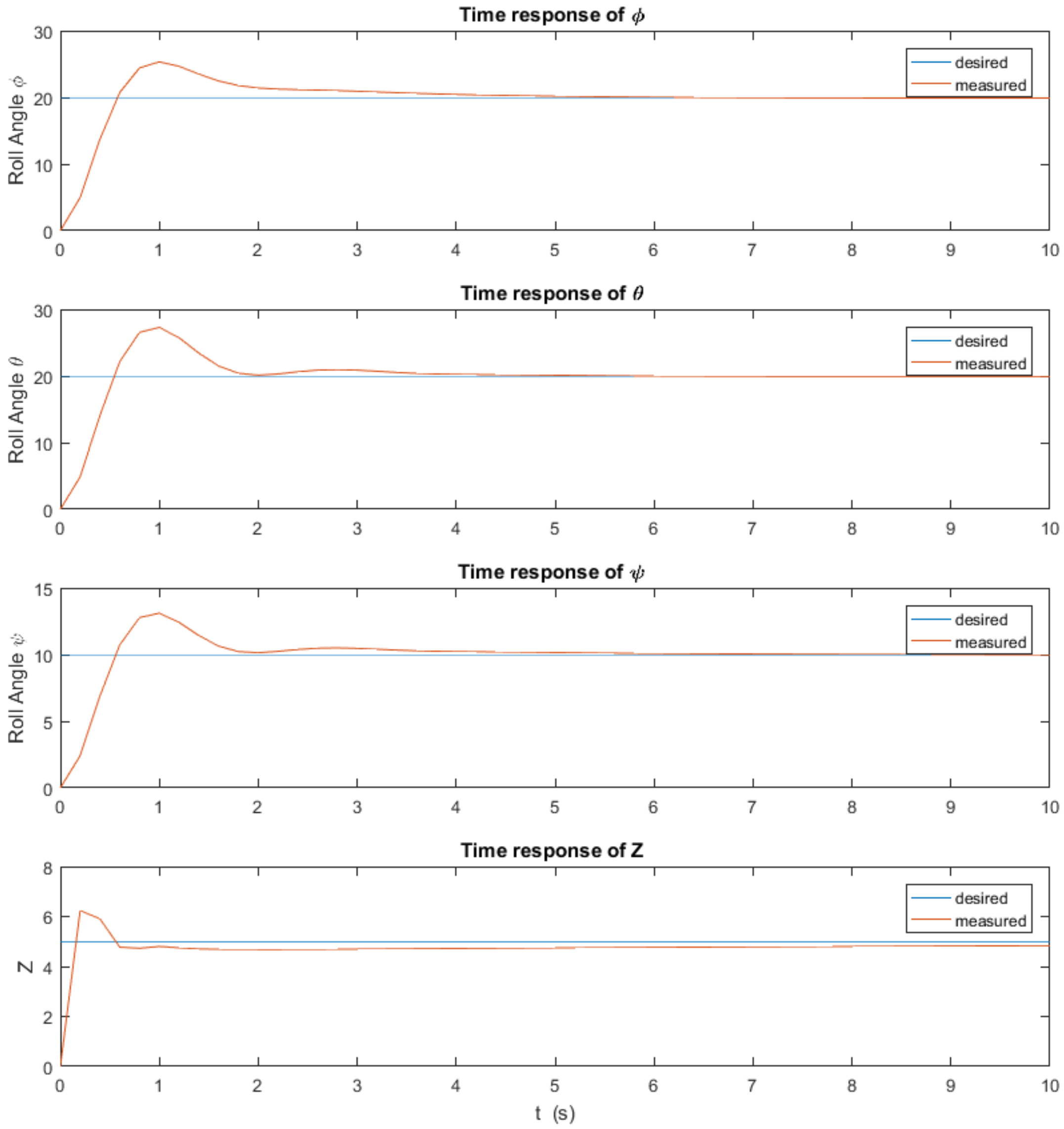

**Figure 4.4** : Simulation results of PID controller on the nonlinear quadrotor model.

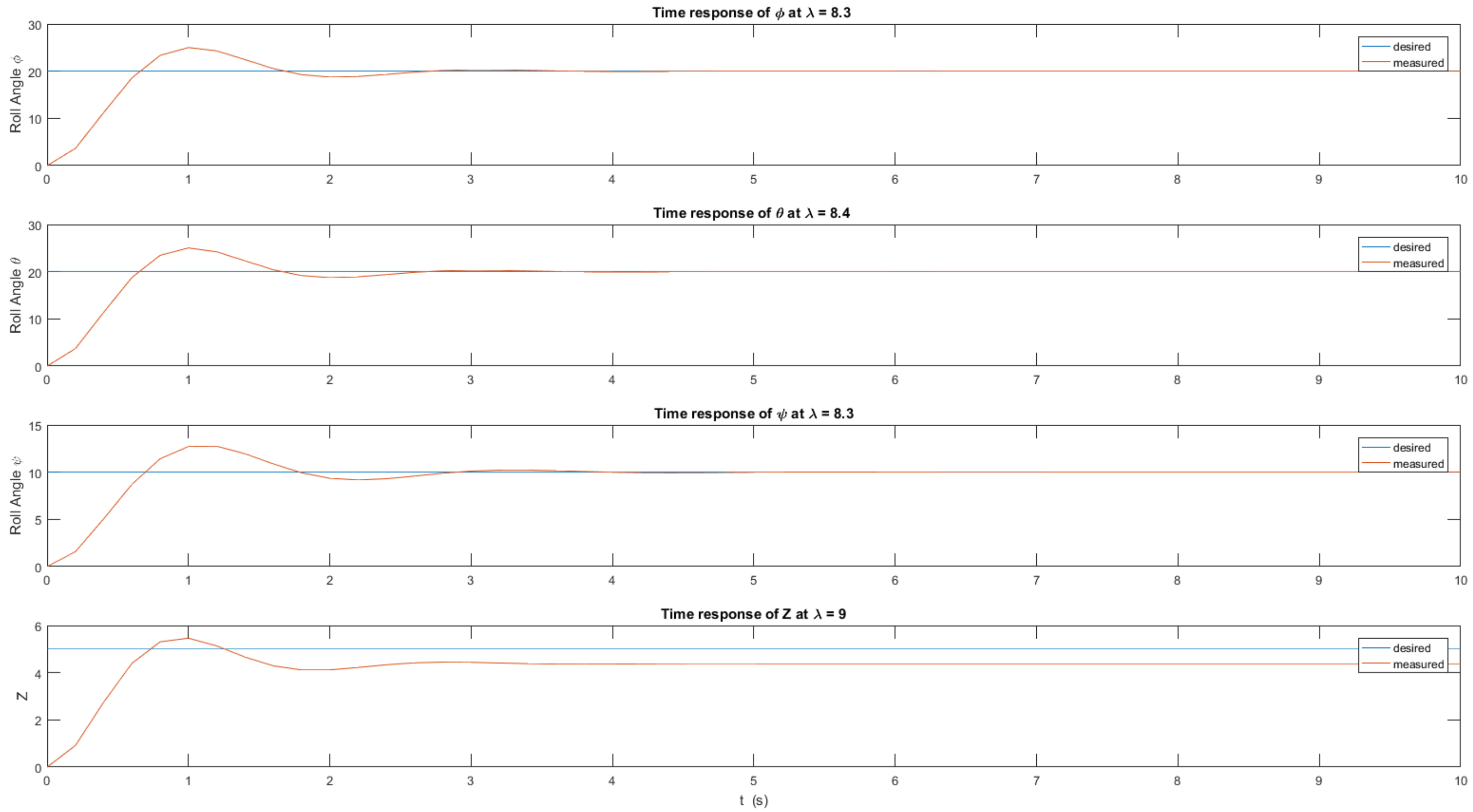

**Figure 4.5** : Simulation results of Feedback nonlinear controller on the nonlinear quadrotor model.

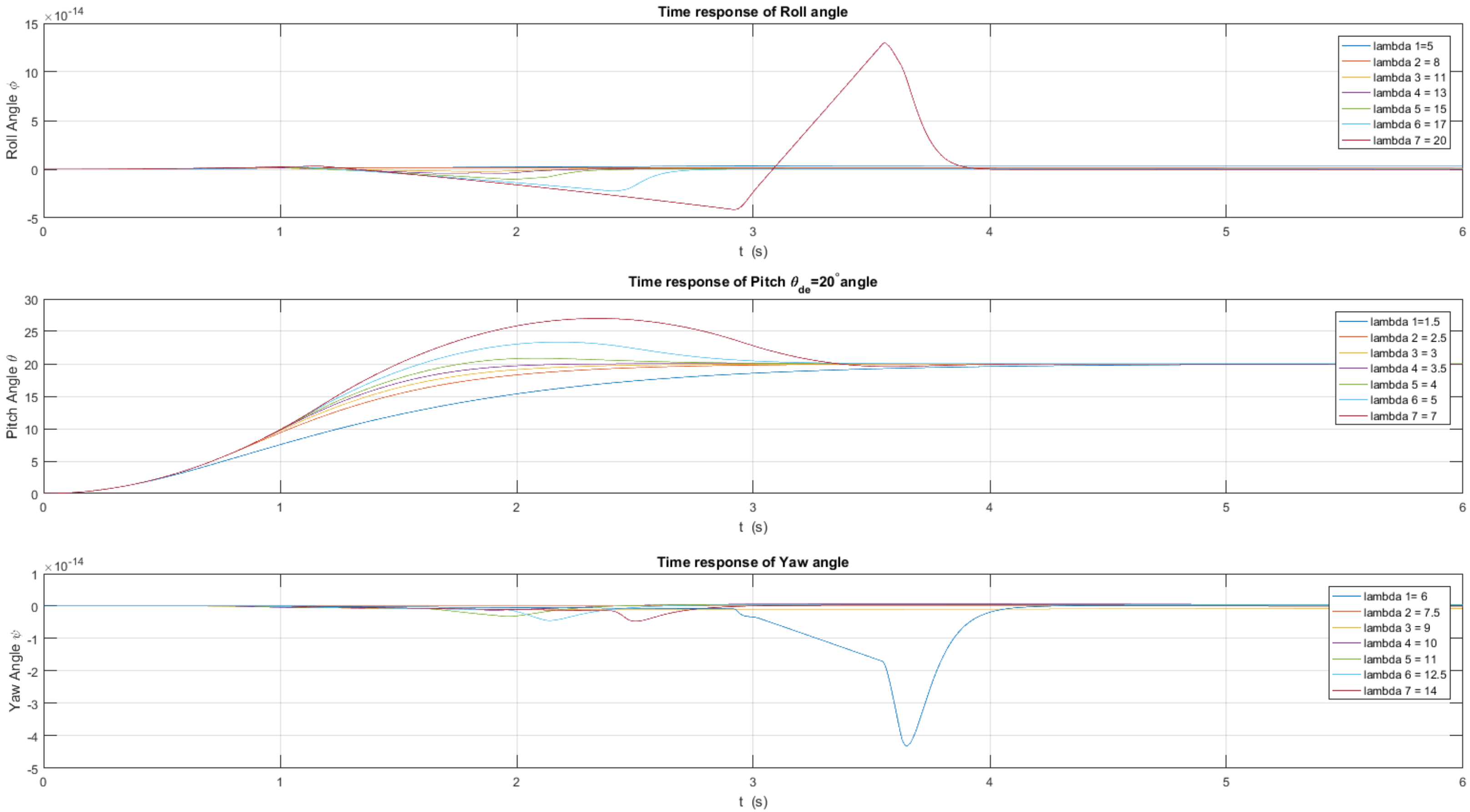

**Figure 4.6** : Simulation results of Feedback nonlinear controller on the HIL testbed $\theta = 20°$.

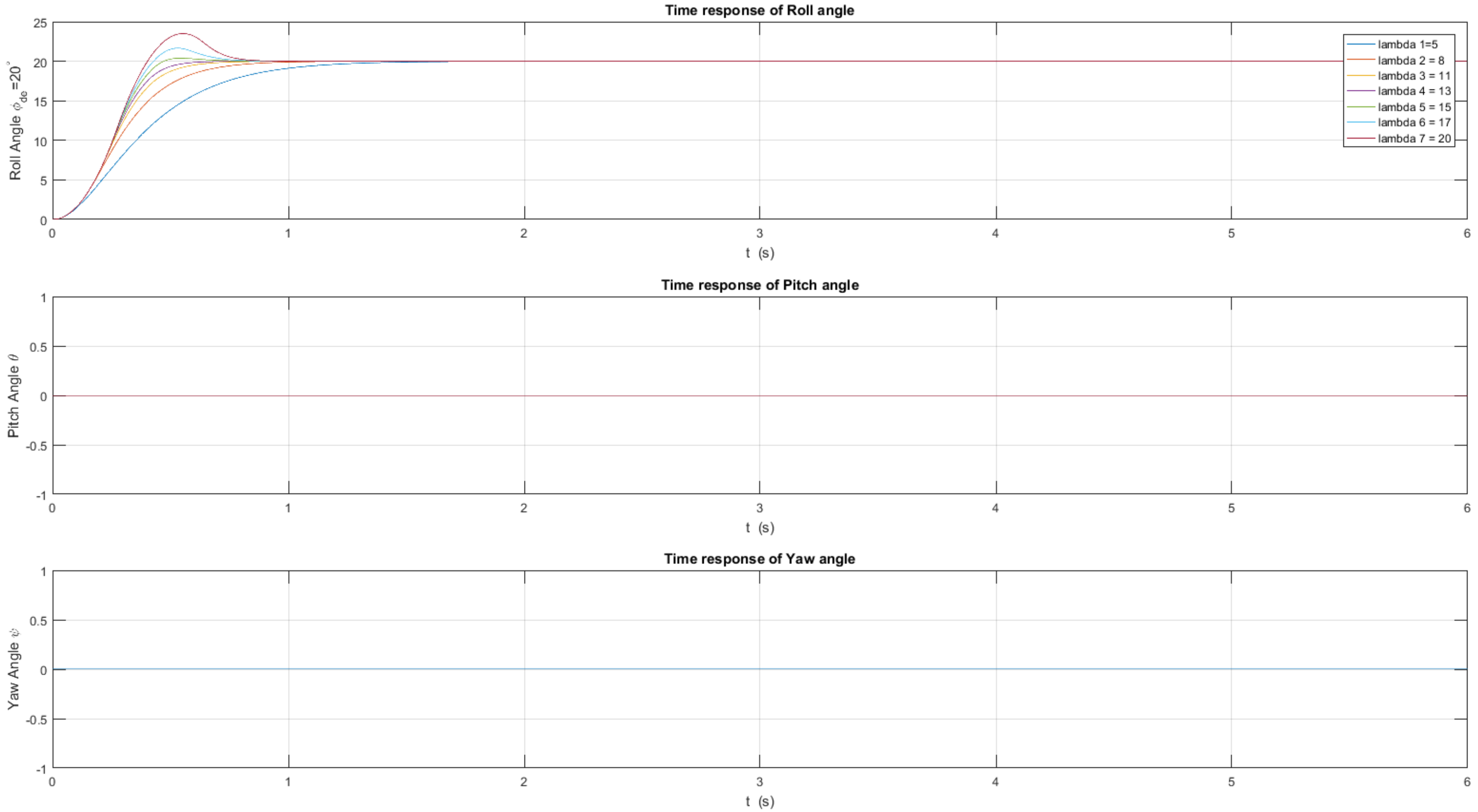

**Figure 4.7** : Simulation results of Feedback nonlinear controller on the HIL testbed $\phi = 20°$.

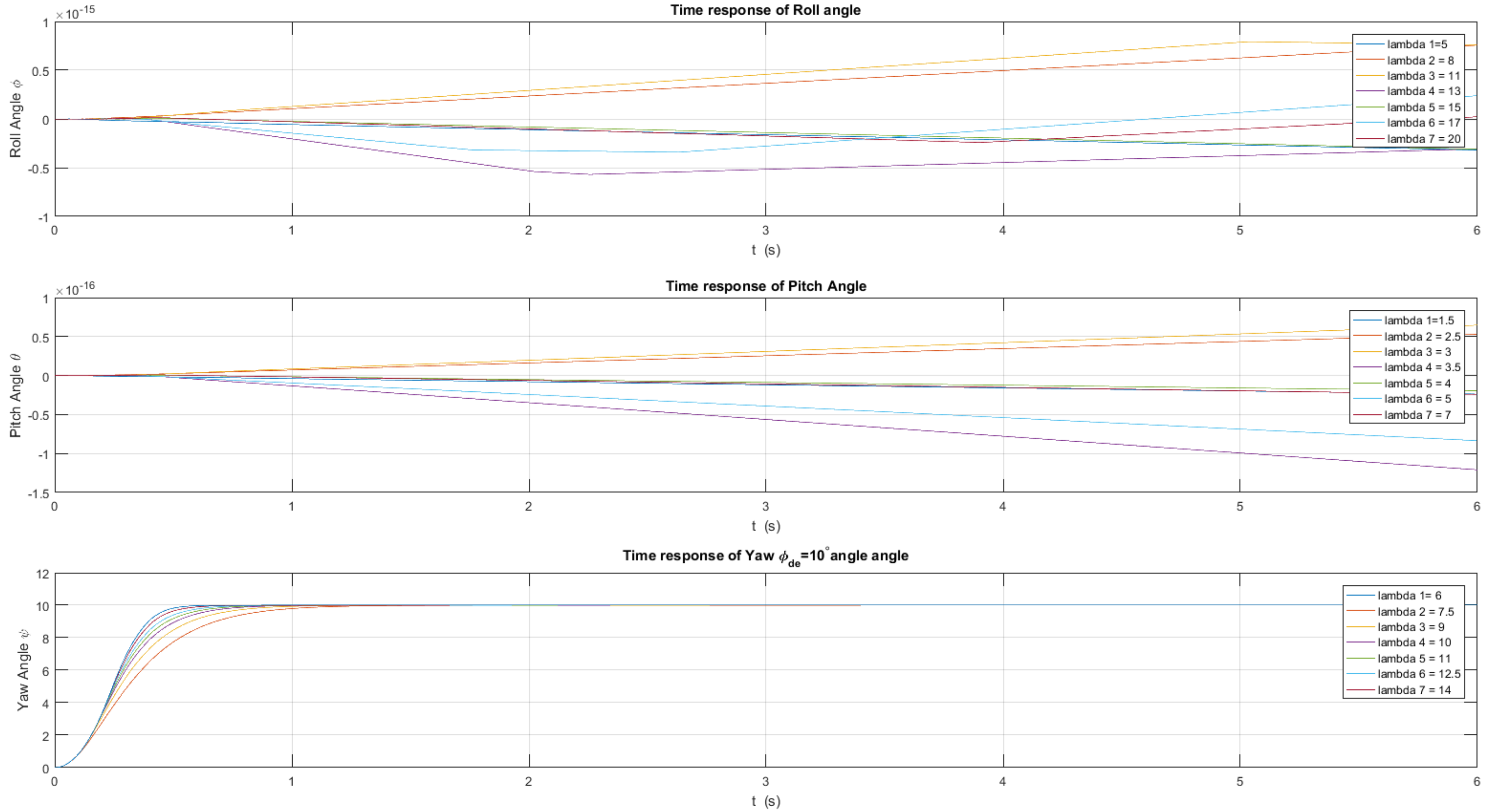

**Figure 4.8** : Simulation results of Feedback nonlinear controller on the HIL testbed $\psi = 10°$.

**Table 4.3** : PID Coefficients of Controller at $1^{st}$ Experiment

| | **Roll Rate** | **Pitch Rate** | **Yaw Rate** |
|---|---|---|---|
| **K_p** | 0.3 | 3 | 2 |
| **K_i** | 0.09 | 0.09 | 0.018 |
| **K_d** | 0.0036 | 0.0036 | 0 |

**Table 4.4** : PID Coefficients of Controller at $2^{nd}$ Experiment

| | **Roll Rate** | **Pitch Rate** | **Yaw Rate** |
|---|---|---|---|
| **K_p** | 0.4 | 4 | 0.35 |
| **K_i** | 0.09 | 0.09 | 0.018 |
| **K_d** | 0.036 | 0.0036 | 0 |

## 4.4 HIL Implementation Results On The Testbench

HIL test platform is designed to research the response of 3DOF actions of quadrotor vehicle. The testbed advantage and disadvantages are seen easily during the real time experiments. The main advantages of the system as seen are easily tuning the coefficients at 3DOF especially pitch and roll axes without time consuming under safety working condition. The negative side is huge amount of the moment fo inertia affects pitch and yaw motion seriously observed easily during the experiments. This section presents real time validation and implementation data of quadrotor vehicle on the HIL testbench via PID and FLC controller results.

### 4.4.1 PID controller real time results

Two experiments are done on the HIL testbed with PID controller via different coefficients as seen in Tables 4.3 and 4.3. First experiment duration is 65 seconds and other is 35 seconds . The response of the roll angle is generally has some overshoots but rise and settling time is so small. About roll motions, there are ignorable overshoots and no time delay, so its so fast and robust. The pitch motions have no overshoots but some time delays and oscillations related with moments of inertia. The yaw motions has high time delays and overshoots generated from high total moment of inertia from outer and inner circle of the testbench.

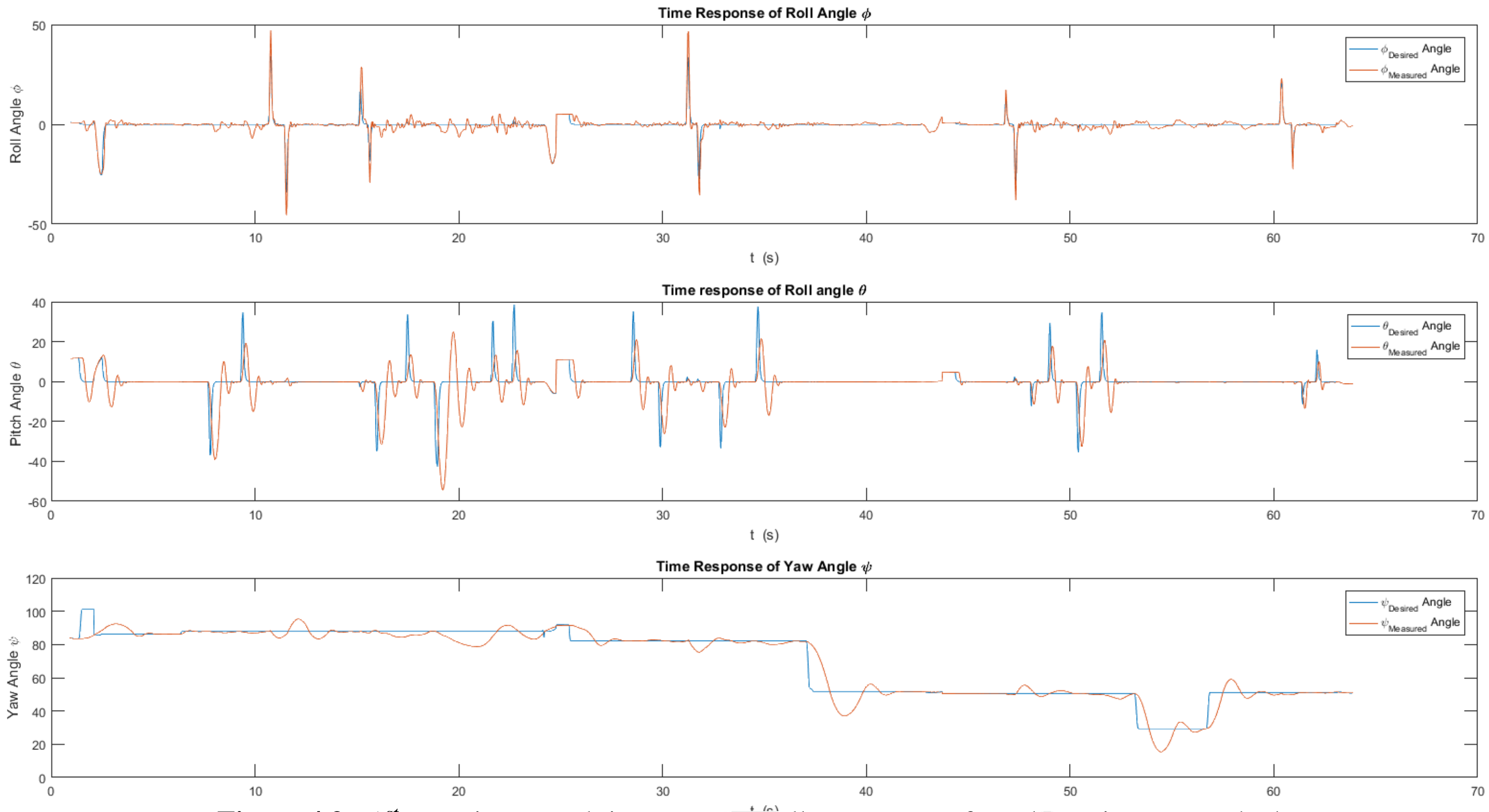

**Figure 4.9** : 1$^{st}$ experiment real time PID controller response of quadrotor in HIL Testbed.

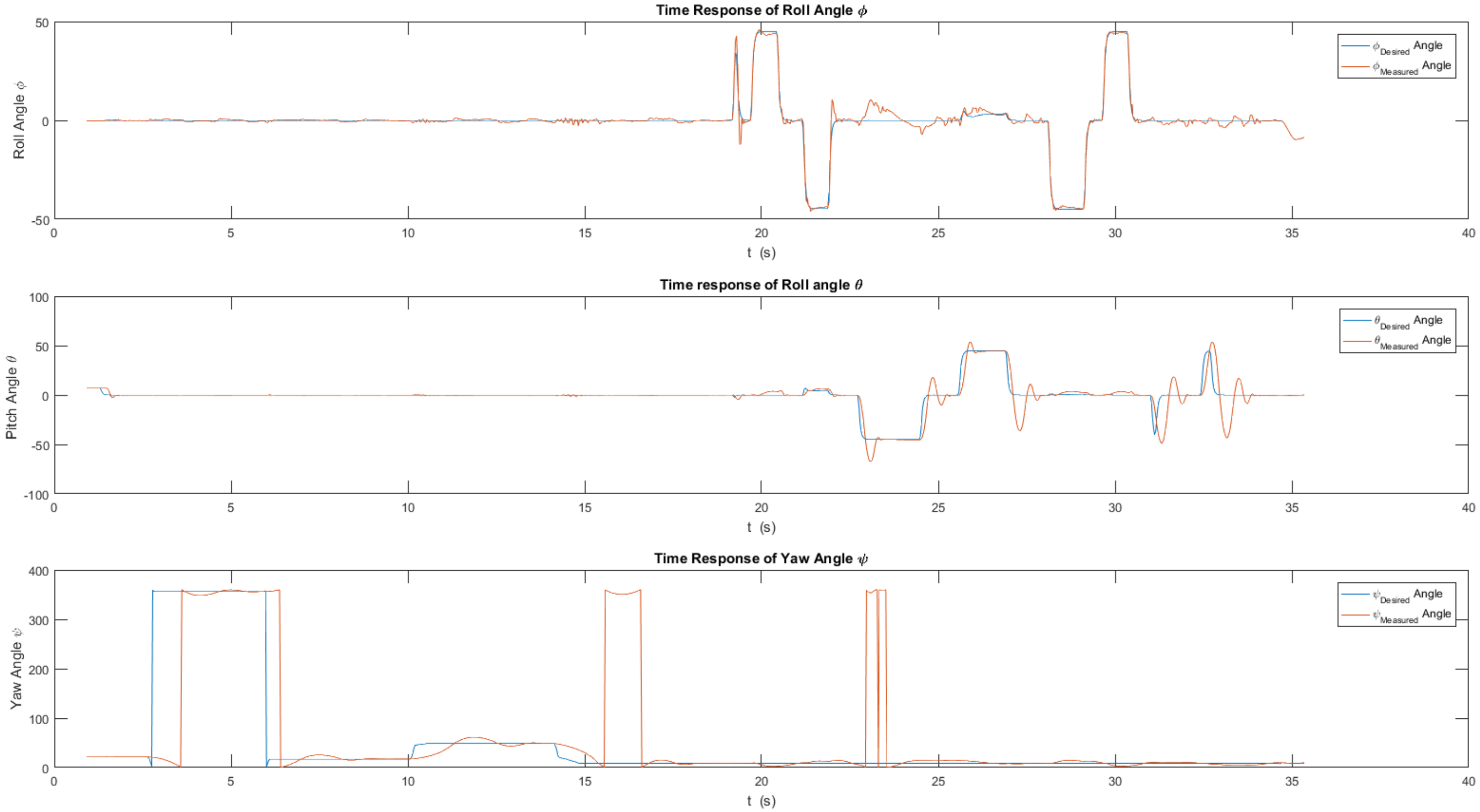

**Figure 4.10** : $2^{nd}$ experiment real time PID controller response of quadrotor in HIL Testbed.

### 4.4.2 Feedback linearized nonlinear controller real time results

Experimental result of quadrotor vehicle on HIL test bench via FLC is shown as graphics in Figure 4.11. The experiment duration is 33 seconds. The response of the roll angle is generally has ignorable overshoots but rise and settling time is so small with no time delay, so it is fast and robust. The pitch motions have little overshoots but no time delays and oscillations. The yaw motions has high time delays and some overshoots generated from high total moment of inertia from outer and inner circle of the testbench.

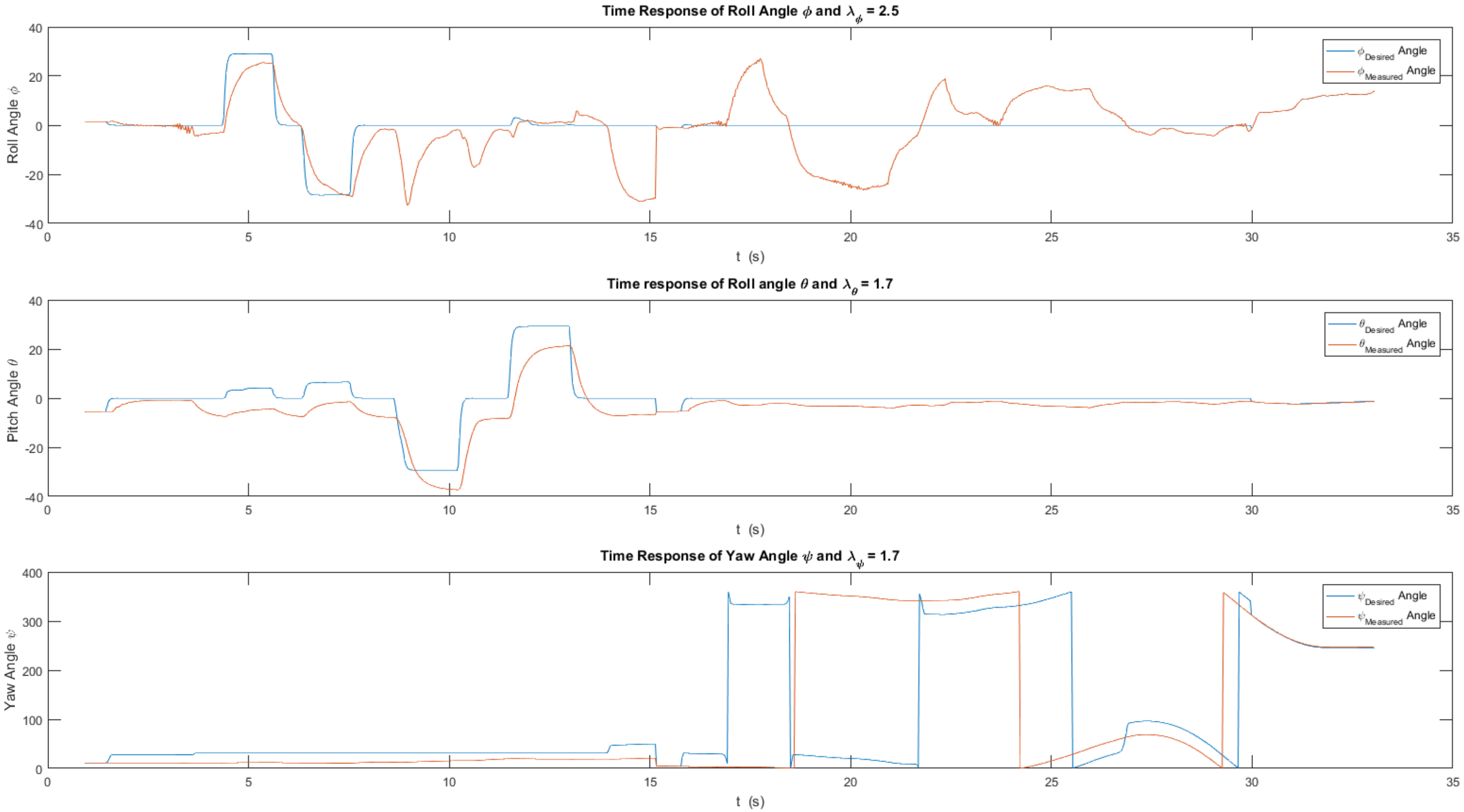

**Figure 4.11** : Real time Feedback linearized controller response of quadrotor in HIL Testbed

# REFERENCES

[1] **Prach, A.**, **Kayacan, E. and Bernstein, D.S.** (2016). An experimental evaluation of the forward propagating Riccati equation to nonlinear control of the Quanser 3 DOF Hover testbed, *American Control Conference (ACC), 2016*, IEEE, pp.3710–3715.

[2] **Nicol, C.**, **Macnab, C. and Ramirez-Serrano, A.** (2011). Robust adaptive control of a quadrotor helicopter, *Mechatronics*, *21*(6), 927–938.

[3] **Yu, Y. and Ding, X.** (2012). A quadrotor test bench for six degree of freedom flight, *Journal of Intelligent & Robotic Systems*, 1–16.

[4] **Bouabdallah, S.**, **Murrieri, P. and Siegwart, R.** (2004). Design and control of an indoor micro quadrotor, *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, IEEE, pp.4393–4398.

[5] **An, H.**, **Li, J.**, **Wang, J.**, **Wang, J. and Ma, H.** (2013). Second-order geometric sliding mode attitude observer with application to quadrotor on a test bench, *Mathematical Problems in Engineering*, *2013*.

[6] **Alexis, K.**, **Nikolakopoulos, G. and Tzes, A.** (2010). Design and experimental verification of a constrained finite time optimal control scheme for the attitude control of a quadrotor helicopter subject to wind gusts, *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, pp.1636–1641.

[7] DRONSGLOBE–Drone Buying Guides Reviews, `http://dronesglobe.com`.

[8] PX4Autopiot, `https://pixhawk.org/`.

[9] **Adigbli, P.**, **Gr, C.**, **Mouret, J.B. and Doncieux, S.** (2007). Nonlinear attitude and position control of a micro quadrotor using sliding mode and backstepping techniques.

[10] **De Lellis Costa de Oliveira, M.**, (2011), Modeling, Identification and Control of a Quadrotor Aircraft.

[11] **Fum, W.Z.** (2015). Implementation of Simulink controller design on Iris+ quadrotor, *Ph.D. thesis*, Monterey, California: Naval Postgraduate School.

[12] **Bouabdallah, S.** (2007). Design and control of quadrotors with application to autonomous flying.

[13] **Min, S.Y.**, **Lee, C.H.**, **Seung, M.H.**, **Kim, Y.S.**, **Hur, C.M. and Kim, S.J.** (2015). Experimental Study of a Quadrotor Cyclocopter, *Journal of the American Helicopter Society*, *60*(3), 1–10.

[14] **Bouabdallah, S. and Siegwart, R.** (2007). Full control of a quadrotor, *Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on*, Ieee, pp.153–158.

[15] **Pounds, P.**, **Mahony, R. and Corke, P.** (2010). Modelling and control of a large quadrotor robot, *Control Engineering Practice*, *18*(7), 691–699.

[16] **Xu, R. and Ozguner, U.** (2006). Sliding mode control of a quadrotor helicopter, *Decision and Control, 2006 45th IEEE Conference on*, IEEE, pp.4957–4962.

[17] **Lebedev, A.**, (2013), Design and Implementation of a 6DOF Control System for an Autonomous Quadrocopter.

[18] **Altug, E.**, **Ostrowski, J.P. and Mahony, R.** (2002). Control of a quadrotor helicopter using visual feedback, *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, IEEE, pp.72–77.

[19] **Bresciani, T.** (2008). Modelling, identification and control of a quadrotor helicopter, *MSc Theses*.

[20] **Reyhanoglu, M.**, **Damen, R. and MacKunis, W.** (2016). Observer-based sliding mode control of a 3-DOF hover system, *Control, Automation, Robotics and Vision (ICARCV), 2016 14th International Conference on*, IEEE, pp.1–6.

[21] **Kırlı, A.** (2010). İnsansız Dört Rotorlu Hava Araçları İçin Değişken Serbestlik Dereceli Yere Sabit Deney Düzeneği Ve Denetleyici Tasarımı, *Ph.D. thesis*, Fen Bilimleri Enstitüsü.

[22] **Mian, A.A. and Daobo, W.** (2008). Modeling and backstepping-based nonlinear control strategy for a 6 DOF quadrotor helicopter, *Chinese Journal of Aeronautics*, *21*(3), 261–268.

[23] **Stebler, S.**, **MacKunis, W. and Reyhanoglu, M.** (2016). Nonlinear output feedback tracking control of a quadrotor UAV in the presence of uncertainty, *Control, Automation, Robotics and Vision (ICARCV), 2016 14th International Conference on*, IEEE, pp.1–6.

[24] **Leishman, G.J.** (2006). *Principles of helicopter aerodynamics with CD extra*, Cambridge university press.

[25] **Naidoo, Y.**, **Stopforth, R. and Bright, G.** (2011). Quad-rotor unmanned aerial vehicle helicopter modelling & control, *International Journal of Advanced Robotic Systems*, *8*(4), 45.

[26] **Castillo, P.**, **Lozano, R. and Dzul, A.E.** (2005). *Modelling and control of mini-flying machines*, Physica-Verlag.

[27] **MathWorks, I.** *MATLAB: the language of technical computing. Desktop tools and development environment, version 2016*.

[28] Unified Engineering, http://web.mit.edu/16.unified.

[29] The Complete Commercial Drone Platform, https://3dr.com/.

[30] ArduPilot Open Source Autopilot, http://ArduPilot.org/.

[31] **Brandt, J.B.**, **Deters, R.W.**, **Ananda, G.K. and Selig, M.S.** UIUC Propeller Database UIUC Propeller Data Site.

[32] QUARTZ, https://qz.com/.

[33] **Briod, A.**, **Kornatowski, P.**, **Zufferey, J. and Floreano, D.** (2014). A Collision-resilient Flying Robot, *Journal of Field Robotics*, *31*(4), 496–509, http:https://doi.org/10.1002/rob.21495.

[34] PID control, Kostas Alexis, http://www.kostasalexis.com/pid-control.html.

[35] **Westphal, L.C.** (2012). *Handbook of Control Systems Engineering*, volume635, Springer Science & Business Media.

**CURRICULUM VITAE**

**Name Surname: Muhsin HANÇER**

**Place and Date of Birth: Tarsus 1986**

**EDUCATION:**

- **B.Sc.:** 2012, Istanbul Technical University, Faculty of Aeronautic and Astronautic, Astronautical Engineering