

# Interfacing Matlab/Simulink with V-REP for A Controller Design for Quadrotor

Javid KHALILOV<sup>1</sup>, Ali Türker KUTAY<sup>2</sup>

<sup>1,2</sup>Middle East Technical University, Ankara, Turkey

---

**Abstract:** The primary factor restricts of the new control systems developments for air vehicles and the implementation of new sensors is being physically unable to quick and cost-effective testing. Flight testing is costly and requires a long preparation process. The aim of this article is improve the simulator infrastructure which easily implementable to the AscTec Hummingbird Quadrotor that located in the University's lab and have interaction with the 3D physical environment. Firstly, quadrotor's mathematical model has been developed and this model is implemented on the Matlab/Simulink environment. Afterwards, a basic PID controller is developed. The quadrotor's physical model, the electric motor, the rotor and the IMU model have been modelled on Virtual Robotics Experimentation Platform (V-REP). Matlab / Simulink controller is synchronized with the V-REP and is used for the quadrotor's physical environment tests. The verification of the simulation is planning made by real experimental flight data of AscTec Hummingbird.

**Keywords:** Quadrotor, Matlab/Simulink, V-REP, PID Controller.

---

## I. INTRODUCTION

The quadrotor is a Vertical Takeoff and Landing (VTOL) rotorcraft which is propelled by four fixed pitch rotor. Because of mechanical simplicity and high mobility research related to the quadrotor topic is widespread. The use of area of quadrotors include military applications such as border patrolling, cartography, cost guards and also civilian applications such as mapping, search and rescue missions, agricultural applications, power and nuclear plants inspection and etc.

Recent years there is a wide range of academic quadrotor and quadrotor testbed studies have been made. Some of them are STARMAC II in Stanford University [1], MARK II X4 Flyer in Australia National University [2], swarm of nano quadrotors in UPenn GRASP Lab [3], quadrotor project in ETH Zurich [4], variable pitch quadrotor in MIT [5], and etc. On the other hand the commercial products also had widespread in this period, such as Microdrones MD4-200 [6], Draganfly [7], Silverlit X-UFO [8], Quattrokooper [9], Asctec Hummingbird [10], and etc.

A quadrotor system is a simple structure but design and implementation of control algorithms is not always an easy task. Especially flight tests in the early stages of algorithm developments can be dangerous and costly. Besides the lack technical hardware such as indoor localization systems (Vicon cameras), 3D scanner and visual sensors, and etc. is reducing development ability. Another challenge the studies related to swarm flight algorithms requires number of quadrotors and technological infrastructure. To avoid these types of problems simulator platform tests and implementations have been spread. Most of these simulator studies are generally developed in Matlab/Simulink platform. In his thesis Martines modelled the quadrotor in Matlab/Simulink platform and his model based around Draganfly XPro quadrotor [11]. The non-linear model of quadrotor was derived using classical mechanics and multiple control techniques were employed in Matlab/Simulink in Marcelo's thesis study [12]. Karwoski developed a model in Simulink based on AscTec Hummingbird quadrotor for design and analysis of variable control systems [13].

The Matlab/Simulink environment is widespread and easy use for developing, debugging and testing complex control algorithms in a fast prototyping and testing. On the other hand Matlab lacks an easy to use 3D physical simulation engine for robotic applications. In this point Virtual Robot Experimentation Platform (V-REP) is one of the alternatives for 3D modeling and physical environment interaction testing. In Matteucci's study simulation environment created for an

autonomous all-terrain mobile robot in V-REP and validation experiments presented in which the behavior of the real system is compared with the corresponding simulations [14]. There are also quadrotor modeling and control algorithm implementation studies in V-REP. One of them is PyQuadSim study in which quadrotor is controlled via joystick or gamepad, and the controller is written in Python and modeling is done in V-REP [15]. In Riccardo’s study the interface of Matlab/Simulink with V-REP using ROS as communication middleware is done and the default V-REP quadrotor model control is done using visual servoing [16]. The configuration of V-REP and ROS to work in parallel, using ROS packages for pose estimation based on vision and for the design and use fuzzy logic control system in quadrotor is done in Mendez’s study [17]. The experimental flights were set by a real Ar.Drone parrot power edition quadrotor and the results support virtual flight test data.

In our study goal is to design Matlab/Simulink model and controller for simulation. To test model and implement controller algorithms and observe 3D physical interaction Matlab/Simulink is interfaced with V-REP directly.

## II. QUADROTOR SETUP

The quadrotor that will be used in this study is an AscTec Hummingbird which is equipped with the AscTec Autopilot. The quadrotor, shown in Fig. 1 is small, lightweight and an ideal platform for testing flights in indoor environments. The main frame of quadrotor is made out of sandwich material consisting of carbon fiber and balsa wood. Due to lightweight and small size control bandwidth does not restrict aggressive maneuvers significantly. The system has a pressure sensor, three axis compass, GPS unit, an acceleration sensor, and three gyroscopes (one for each axis). AscTec Autopilot main board consist of two ARM 7 60MHz, 32 Bit microcontrollers – a low level processor (LLP) and a high level processor (HLP). LLP collects IMU (accelerometer + gyroscopes) data, sends commands to the motor controller and is responsible from attitude control of quadrotor. The HLP controls the GPS and free for user defined control algorithm designs. The control algorithms can be designed in Matlab/Simulink environment, the control system is then translated to the C code by the Real Time Workshop Embedded coder. The AscTec HL-SDK provides all necessary tools to program custom C code and to flash and debug the code on the processor. The communication of AscTec Hummingbird with ground is provided by a wireless Xbee model via serial port. The technical details of the quadrotor are summarized in TABLE I [18].

TABLE I: Technical Details of AscTec Hummingbird Quadrotor

Model	AscTec Hummingbird
Manufacturer	Ascending Technologies GmbH
Takeoff weight (2100 mAh TP battery)	480 g
Distance between motor axes	34 cm
Flight control and IMU board	AscTec Autopilot
Propeller	8” flexible standard propeller grey
Motors	AscTec X-BL 52s
Motor controller	AscTec X-BLDC
Thrust per Motor (standard propeller)	0,05 – 3,5 N
Radio control system	Futaba Fasst 2,4 GHz
Telemetry System	Xbee 2,4 GHz
Moment of inertia	$I_{xx} = I_{yy} = 5,6 \cdot 10^{-3} \text{ kg} \cdot \text{m}$ ; $I_{zz} = 8,1 \cdot 10^{-3} \text{ kg} \cdot \text{m}$

The propellers used in quadrotor are fixed pitch propellers, so the control of quadrotor is done by changing the rotation speed of propellers. The front and rear propeller are rotating clockwise and the left and right propellers are rotating counter clockwise. Roll motion is controlled by manipulating the rotation speed of right and left propellers, pitch motion is controlled by varying the rotation speed of front and rear propellers. Yaw motion is controlled by speeding up the propellers rotating in one direction and slowing down ones spinning in other direction. So the thrust does not change, but change of angular momentum causes rotation in yaw axis.



Fig.1. AscTec Hummingbird quadrotor

### III. MATHEMATICAL MODEL AND PID CONTROLLER

The quadrotor that we use in our study is fixed pitch propeller quadrotor. The angle of attack of fixed pitch propellers are constant and thrust force produced by propeller is adjusted by changing the rotational speed of corresponding rotor. The relation between thrust and rotational speed is as follows,

$$T = c_T \omega^2$$

where  $c_T$  is motor-prop specified thrust coefficient. The relation between propeller torque and rotational speed is as follows,

$$Q = c_Q \omega^2$$

The control of quadrotor model is done by adjusting rotational speed of each propeller. The matrix describing the thrusts and torques shown below:

$$\begin{bmatrix} \Sigma T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & dc_T & 0 & -dc_T \\ -dc_T & 0 & dc_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

where  $d$  is the distance between the propeller rotation axes. Deriving state equations

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = (I_B)^{-1} \left[ M - \Omega I_B \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right] \quad \text{where } I_B \text{ is inertia matrix and, } \Omega = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}$$

The next state allows us to determine the rate of change of the Euler angles in the inertial frame

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & t(\theta) s(\phi) & t(\theta) c(\phi) \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) / c(\theta) & c(\phi) / c(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

In next equations we discuss the velocity state equation

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \left(\frac{1}{m}\right) F + g_b - \Omega \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \text{where } g_b = R_{bi} g_i$$

$$R_{bi} = \begin{bmatrix} c(\theta) c(\psi) & c(\theta) s(\psi) & -s(\theta) \\ (-c(\phi) s(\psi) + s(\phi) s(\theta) c(\psi)) & (c(\phi) c(\psi) + s(\phi) s(\theta) s(\psi)) & s(\phi) c(\theta) \\ (s(\phi) s(\psi) + c(\phi) s(\theta) c(\psi)) & (-s(\phi) c(\psi) + c(\phi) s(\theta) s(\psi)) & c(\phi) c(\theta) \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_{ib} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

This state equation determines the velocity of quadrotor in (x,y,z) directions of the inertial frame. Using this mathematical model PID controller is designed for the altitude and attitude control of the quadrotor. The hovering height and roll, pitch, and yaw angles have been controlled. The general controller scheme for roll control and hover control can be seen in Fig. 2. The pitch and yaw controller scheme are similar to roll controller.

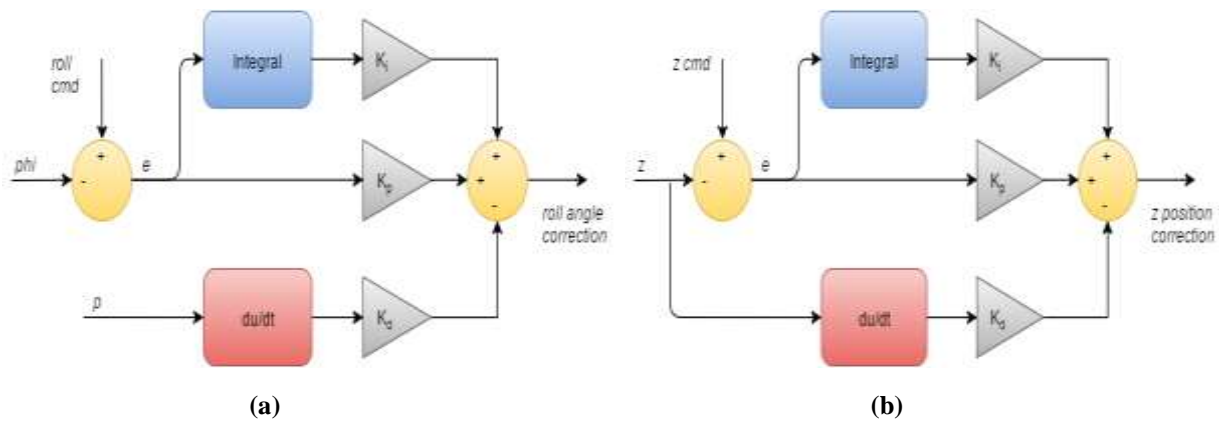


Fig.2. (a) Roll angle PID controller, (b) Altitude PID controller

#### IV. V-REP MODEL AND MATLAB/SIMULINK INTERFACING

In our study we have used the Virtual Robot Experimentation Platform (V-REP), a physical simulator which provides an easy and intuitive environment to create your own virtual platform and to include some popular robots, objects, structures, actuators and sensors. There are wide variety of sensors including accelerometer, gyroscope, GPS unit, load sensors, 2D and 3D vision sensors, and etc. An embedded script is based on LUA language, which associate communication between sensors, robots and external controller. The external controller can be designed using remote api functions for C/C++, Python, Java, Octave and Matlab. In our case the controller will be implemented via Matlab/Simulink.

##### A. Propeller Model:

The propeller of AscTec Hummingbird quadrotor has been modeled in V-REP environment. The experimental data have been used in modeling study. The experimental values are obtained from Derya's experimental studies [19]. The Fig.3 shows the experimental thrust and torque measurements of propeller with varying rotation speed (RPM).

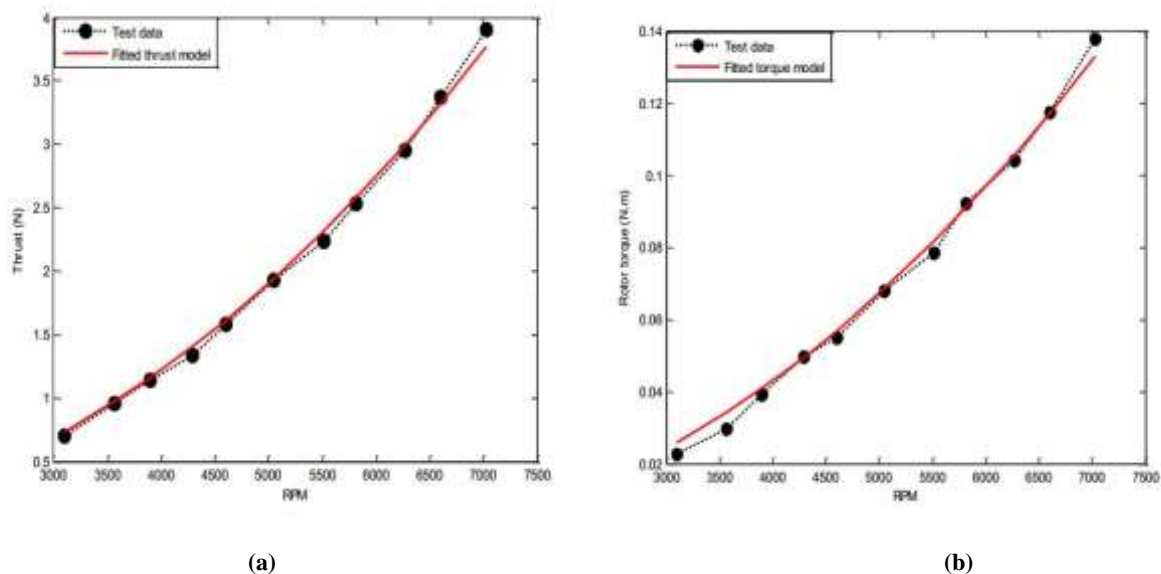


Fig.3. (a) Thrust change w.r.t. the angular velocity of the rotor using BET in hovering flight, (b) Rotor torque change w.r.t. the angular velocity of the rotor using BET in hovering flight

The thrust coefficient ( $N/RPM^2$ ) and rotor torque coefficient ( $Nm/RPM^2$ ) parameter are found by using the method of least squares.

$$c_T = 7.6184 \times 10^{-8} N/RPM^2$$

$$c_Q = 2.6839 \times 10^{-9} Nm/RPM^2$$

The estimated parameter then has been implemented in V-REP environment. The CAD model of propeller is imported to the platform and appropriate LUA embedded script has been written which gets the rotation speed and particle velocity from main script. Particle velocity is modeled for ground effect studies in future studies. In Fig.4 there is one example work of propeller at 3900 RPM which attached to the cube and force and torque sensors have been mounted.

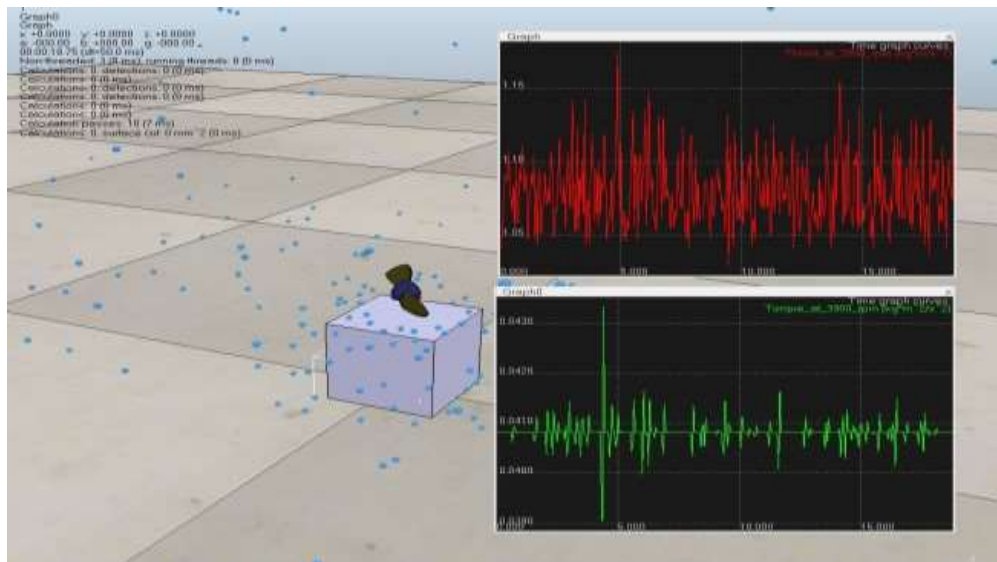


Fig.4. Propeller model in V-REP

### B. Quadrotor Model in V-REP:

V-REP provides a very powerful tool to create and modify mechanical model. It is comfortable that you assemble all the parts in your CAD program first and export the model into a single stl file. It is easy to import this stl file V-REP and divide the model into smaller parts and assign them own mechanical properties. The quadrotor model stl file for the AscTec Hummingbird quadrotor is downloaded from the AscTec wiki [20]. The stl file imported into the V-REP and a simplified dynamic model created for simulation. An appropriate mass and inertia value has been set. Then developed propeller model has been included to the model. This process is done by copy and paste between propellers designed scene and quadrotor model scene. In Fig.5 general view of quadrotor model can be seen.

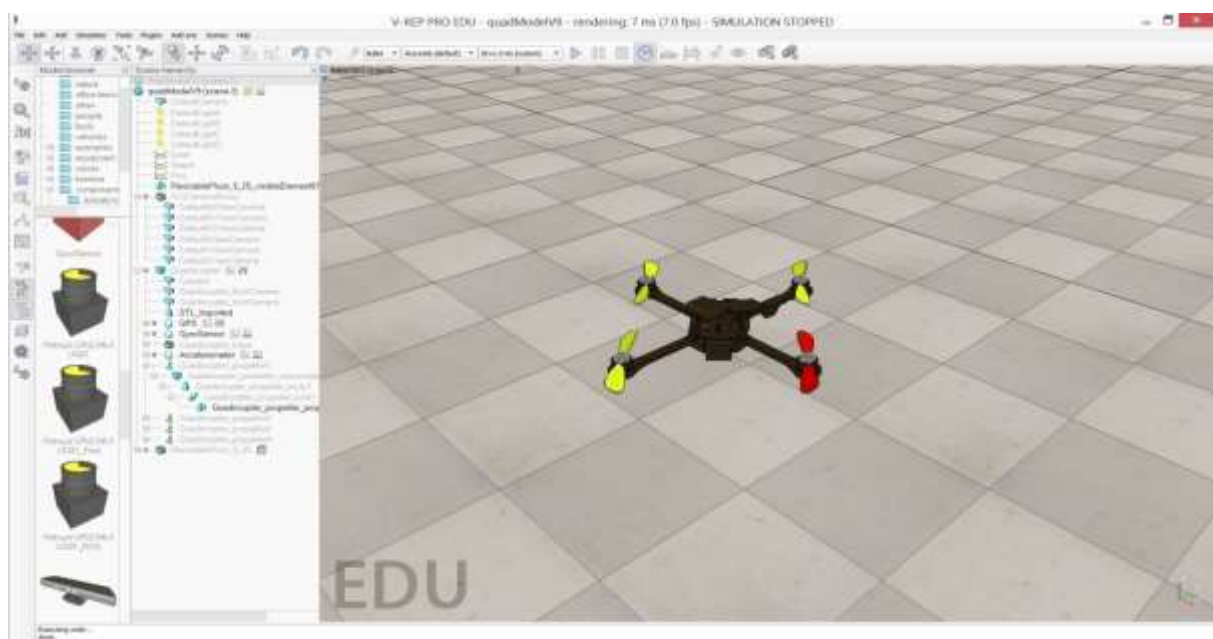


Fig.5. General AscTec Hummingbird Quadrotor View in V-REP



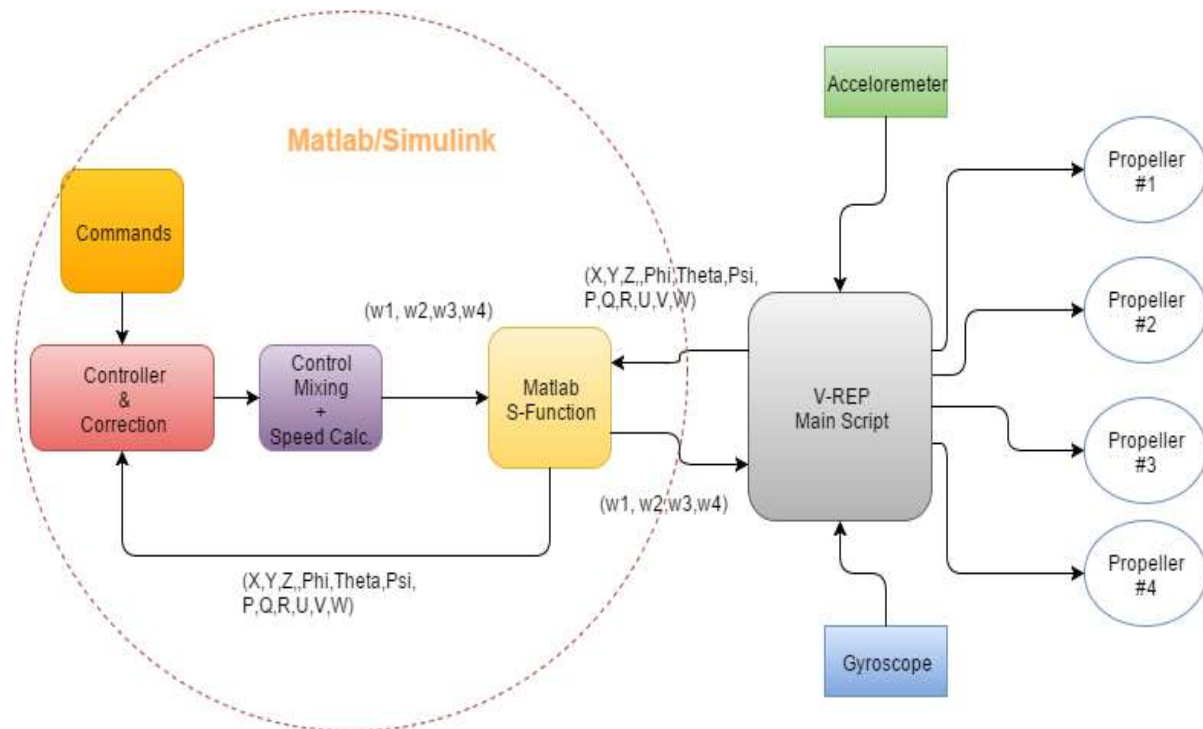
Then sensor components like accelerometer, gyroscope and GPS unit has been added to simulation platform. In this point of study the accelerometer and gyroscope sensors have been used for calculation of roll and pitch angle. The sensors have their own embedded LUA scripts and using these scripts the data has been in read in main script. To obtain angular position both accelerometer and gyroscope data can be used. Gyroscope can do this by integrating angular velocity over time, with accelerometer it can be done by determining the position of gravity vector by using atan2 function. But in gyroscope due to integration over time, the measurement has tendency to drift and not returning to zero when it comes its original position, so gyroscope data is reliable in short term. On the other side, accelerometer all the small forces will disturb our measurement completely, so accelerometer data is reliable in long term. Using complementary filter will solve this problem because using this filter on the short term data from the gyroscope will be used and on the long term data from accelerometer will be used. The complementary filter looks like as below:

$$angle = 0.98 * (angle + gyroData * dt) + 0.02 * (accData)$$

Using formulation above the appropriate code has been developed in main script to calculate roll and pitch angle.

**C. Matlab/Simulink and V-REP Interfacing:**

V-REP offers a remote API allowing to control a simulation from an external application. The V-REP remote API is composed by approximately one hundred functions that can be called from Matlab program. In our application the command and controller part is developed in Simulink and actuation and physical interaction part created in V-REP. The general scheme of developed simulator can be seen in Fig.6.



**Fig.6. General operation diagram of simulator**

In Simulink part controller gets commands and state data, makes corrections and sends this data to control mixing and motor speed calculation block. Then speed data is sent to Matlab S-Function where all communication process is done. In Matlab S-Function the remote API functions and Matlab functions are used. The remote API functions are interacting with V-REP via socket communication in a way that reduces lag and network load to a great extent. The synchronization with V-REP is also done in this part of simulation. Matlab S-Function sends motor speeds to V-REP and gets state data from V-REP and sends these data to controller and correction block in Simulink. In V-REP side main LUA script gets motor speeds from Matlab/Simulink and sends these values to appropriate propeller model child LUA scripts. Meanwhile, main LUA script read accelerometer and gyroscope data from child scripts, implement complementary filter to these data and calculate roll and pitch angles. On the other hand, also gets absolute positions (x,y,z) and absolute velocities (u,v,w) of quadrotor, combining them with angle (φ,θ,ψ) and angular velocity (p,q,r) data sends them to Matlab/Simulink application. The visualization of processed data is done both in Matlab/Simulink and V-REP application.

### V. SIMULATION RESULTS

Afterwards interfacing the Matlab/Simulink with V-REP controller tuning and flight simulations trials have done. Processing with hand tuning and referring to Karwoski's studies [13] PID gains have been determined and can be seen in TABLE II.

TABLE II: Tuned PID Gains

Gains	Hover	Roll Angle	Pitch
$K_p$	16	25	25
$K_i$	1	0	0
$K_d$	10	6	6

After tuning PID gains some flight conditions have been tested in simulator. One of them is to hover from 1 m to 1.5 m and 10 degree roll angle doublet input in 1 sec. The motor command and state results can be seen in Fig.7 and Fig.8.

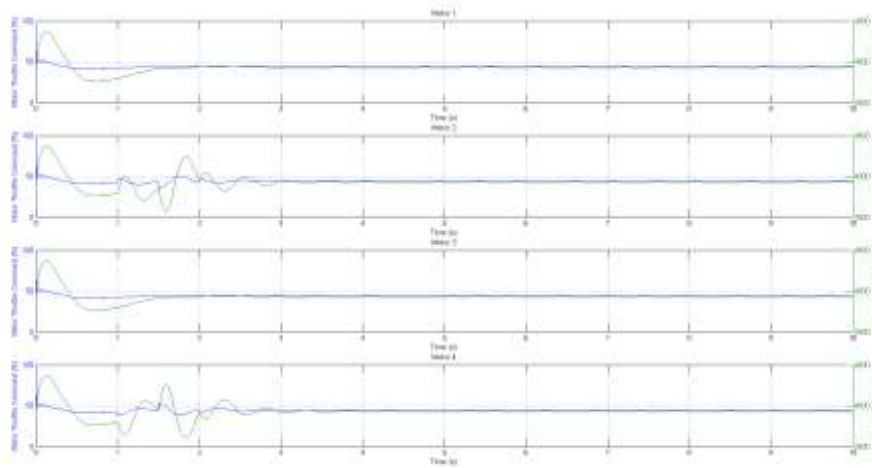


Fig.7. Motor throttle command and rpm values

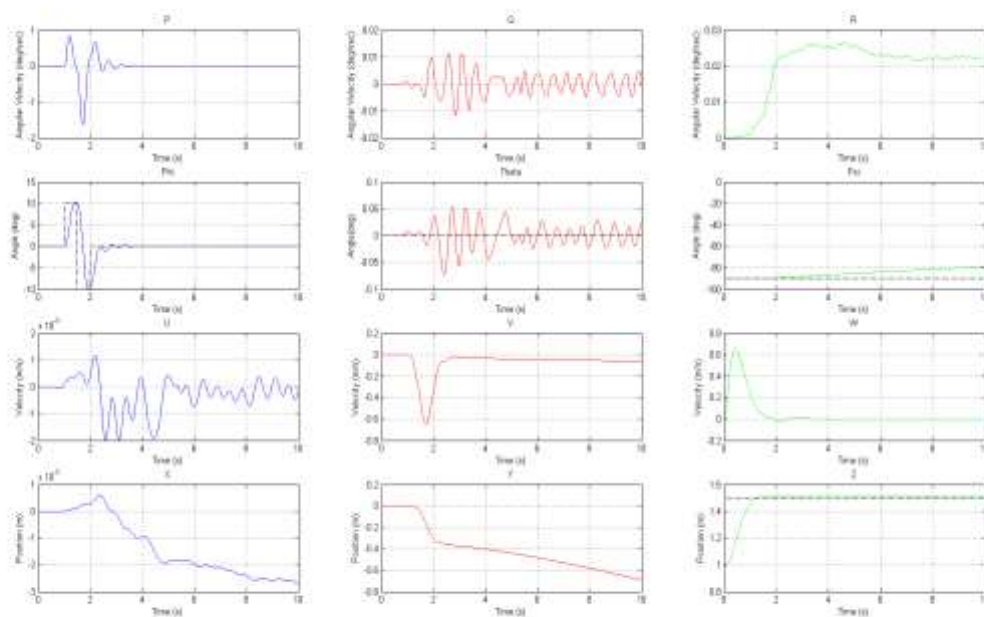


Fig.8. State values of quadrotor during simulation

## VI. CONCLUSION AND FUTURE WORK

In this paper the easy use 3D simulator is designed for the test of variety control algorithms in AscTec Hummingbird quadrotor. First the dynamic model of the quadrotor is derived and PID controller is designed in Matlab/Simulink platform. Afterwards propeller model is created in V-REP relying on real test data of propeller. The 3D model of quadrotor is also designed and accelerometer, gyroscope sensors, and GPS unit is added to quadrotor model in V-REP. Then the controller in Matlab/Simulink is interfaced with quadrotor model in V-REP and three dimensional physical environment interacted tests become easier in this combined simulator. The flight test has been made in simulator. As a next step first author is planning to make experimental flights with real AscTec Hummingbird quadrotor and compare the results with simulator, afterwards adding 3D vision sensor for localization and mapping algorithm tests.

## REFERENCES

- [1] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. Jang and C. J. Tomlin, "The Stanford testbed of autonomous rotorcraft for multi agent control(STARMAC)," Digital Avionics Systems Conference, 2004. DASC 04. The 23<sup>rd</sup>, Vol.2, 2004.
- [2] P. Pounds, R. Mahony and P. Corke, "Modeling and Control of a Quad-Rotor Robot", Australian National University, Canberra, Australia, 2006.
- [3] University of Pennsylvania, GRASP Lab., <http://www.gizmag.com/grasp-nano-quadrotor-robots-swarm/21302/>
- [4] Flying Machine Arena, Institute for Dynamic Systems and Control, ETH Zurich, <http://flyingmachinearena.org/>
- [5] M. J. Cutler, "Design and Control of an Autonomous Variable-Pitch Quadrotor Helicopter", M.Sc. thesis, Massachusetts Institute of Technology, Massachusetts, USA, 2012.
- [6] Microdrones MD4-200, <http://www.microdrones.com/md4-200.html>
- [7] Draganfly, <http://www.draganfly.com/industrial/products.php>
- [8] Silverlit X-UFO, <http://www.silverlit-flyingclub.com/xufo-body.htm>
- [9] Quattrokoetter, <http://www.lcc-shop.de>
- [10] Ascending Technologies, <http://www.asctec.de/uav-uas-drohnen-flugsysteme/asctec-hummingbird/>
- [11] V. M. Martinez, "Modelling of the Flight Dynamics of a Quadrotor Helicopter", M.Sc. thesis, Cranfield University, Cranfield, UK, 2007.
- [12] M. Oliveira, "Modeling, Identification and Control of a Quadrotor Aircraft", M.Sc. thesis, Czech Technical University in Prague, Prague, Czech Republic, 2011.
- [13] K. Karwowski, "Quadcopter Control Design and Flight Operation", NASA USRP – Internship Final Report.
- [14] G. Bardaro, A. Cucci, L. Bascetta and M. Matteucci, "A simulation based architecture for the development of an autonomous all terrain vehicle", 4th International Conference, SIMPAR 2014, Bergamo, Italy, October 20-23, 2014. Vol.8810, pp. 74-85, 2014.
- [15] PyQuadSim: An open-source Python Quadcopter Simulator for Linux, Windows, and Mac OS X, <http://home.wlu.edu/~levys/software/pyquadsim/>
- [16] R. Spica, G. Claudio, F. Spindler and P.R. Giordano, "Interfacing Matlab/Simulink with V-REP for an Easy Development of Sensor-Based Control Algorithms for Robotic Platforms", University of Rennes.
- [17] M. Mendez, S. Kannan and H. Voobs, "Setting Up a Testbed for UAV Vision Based Control Using V-REP & ROS: A Case Study on Aerial Visual Inspection", 2014 International Conference on Unmanned Aircraft Systems (ICUAS) May 27-30, 2014. Orlando, FL, USA
- [18] M. Achtelik, "Nonlinear and Adaptive Control of a Quadcopter", M.Sc. thesis, Technische Universität München, München, Germany, 2010.
- [19] D. Kaya, "Modeling and Experimental Identification of Quadrotor Aerodynamics", M.Sc. thesis, Middle East Technical University, Ankara, Turkey, 2014.
- [20] AscTec Hummingbird CAD Model, <http://wiki.asctec.de/display/AR/CAD+Models>.