2011 8th International Multi-Conference on Systems, Signals & Devices

Autonomous aerial vehicles, guidance, control and signal processing platform

Mohammad A. Al-Jarrah

Professor, American University of Sharjah

Sonny Adiansyah; Zaid Khalil Marji, Md Sheruzzaman Chowdhury

Graduate Students, American University of Sharjah

Abstract

This paper presents the work done in the field of autonomous unmanned aerial vehicle test bed UAV development at American University of Sharjah. The key areas for establishing the system includes flight control system development for accomplishing complete autonomous flight, design and evaluation of avionics unit and air-data system, and image acquisition and target geolocation. The development include the low level autopilots, the guidance algorithm for way point navigation, INS-GPS sensor fusion, signal processing, ground-station, and gimbaled camera payload integration. The test bed was prototyped using hardware in the loop simulation using mpc555 embedded system and simulink tools. The paper summarizes flight test results for auto-takeoff and landing, trajectory tracking, waypoint navigation, and target tracking. Simulation results for control and navigation of UAV are presented and discussed. Experimental results from real flight tests validate the control, navigation and target tracking algorithms.

978-1-4577-0411-6/11/$26.00

1. Introduction

Autonomy of UAVs has been a growing area of research in the aerospace technology for a few decades now. Before the emergence of microcontrollers and embedded software, the autopilot of UAVs used to be built with analog circuits that formed several PID loops to provide low level stability controller. Nowadays, microcontrollers with embedded software are used to develop not only stability augmentation autopilots but also high level guidance and navigation autopilots. Commercial, off-the-shelf (COTS) enabled research and development programs in most educational institutions. Brigham Young University developed their micro autonomous aircraft over several years [1]. Their research effort resulted in commercializing the Kestrel autopilot system. Georgia Institute of Technology has established their UAV research platform using Almost Ready To Fly (RTF) size 60 of the shelf model aircraft [2]. Other examples include Stanford Dragonfly project [3], Massachusetts Institute of Technology (MIT), Pennsylvania State University, University of Colorado, and many others [4].

There are several commercial autopilots in the market widely used for developing such systems such as Piccolo produced by Cloud Cap Technology [5], MicroPilot MP 2028 and more recent 2128, Kestrel by Procerus Technologies [5], etc. These systems have their advantages and disadvantages. However, these autopilots are limited in terms of their use for research and development since access to the source software is normally not available and if purchased are too expensive and if bought are not documented well to be used for implementing new navigation algorithms or trajectory tracking, or target following. Therefore, to contribute to the research and development in this filed one needs to develop the autopilots, and the flight control station. Most of these autopilots are capable of waypints navigation, auto-takeoff and landing, altitude hold, air speed hold, and Multi-UAV Support.

Table 1: Comparison of physical specifications of autopilots

	Size (cm)	Weight (g) w/o radio	Power Consumption	DC In (V)	CPU	Memory (k)	Price (k USD)
Kestrel 2.2	5.08x3.5x1.2	16.7	500mA (3.3or 5V)	6-16.5	Rabbit 29 MHz	512	5
MP 2028	10x4x1.5	28	140mA 6.5 V	4.2-26	3MIPS	-	5.5
MP 2128	10x4x1.5	28	140mA 6.5 V	4.2-26	150 mips RISC	NA	7.0
Piccolo SL	11.9x5.7x1.8	45	4W	4.8-24	40MHz	448	7.0
Piccolo Plus	14.2 x 4.6 x 6.2	140	4W	8-20	40 MHz		

This paper presents the development of a complete low cost autonomous aerial vehicle. The article summarizes development of the model, Hardware In the Loop Simulation (HILS) platform for rapid prototyping of the various autopilots and navigation algorithms, and system and day time camera payload integration. In addition, summary of rigorous flight tests to verify the functionality and the robustness of the whole system is reported.

Integral part of this system is the payload which consists of a gimbaled camera and a video link with a target following algorithm. The pan and tilt camera is gyro-stabilized to perform target tracking and high value targets surveillance functions.

1.1 Overview of AUS fixed wing activities

A lot of research work has been done in the area of UAV at AUS. The first research was completed by a Hadi [4] a graduate student. He designed, built and tested a low cost avionics unit consisting of an embedded system integrated with sensors and actuators using 2 hcs12 microcontrollers. Hassan [6] a graduate student completed his thesis in 2006 in which he developed the Hardware in the Loop Simulation (HILS) and path planning system using the same platform developed by Hadi [4]. This setup was developed further to simulate several UAV test-beds utilized in this research activities to enable the rapid prototyping of the flight control embedded software for performing tasks such as waypoint navigation in three dimensions, trajectory tracking, no-fly zone avoidance, in addition to the autopilot low level PID control loops.

Abu-Hashim [7] developed fault tolerant algorithm for GPS-INS integration using loosely coupled and tightly coupled INS-GPS extended Kalman Filter to estimate the position and pose of the aircraft. This work was continued by Sahawneh [8] to produce a real time implementation and actual production of the GPS-INS algorithm using analog devices accelerometers, rate gyros, and Honeywell digital compass. The phyCORE –MPC555c microcontroller (72×57)mm, a product of Freescale Semiconductors, Inc., is chosen as the navigation computer. The Navigation system is equipped with an IMU (developed, designed and calibrated in-house), digital compass (HRM3000) and an Inertial Navigation aided with GPS (MIDG II). The real-time INS/GPS algorithm is developed on a desktop and cross-compiled to generate code that can be used in the target navigation computer, hence this process is accomplished in two stages:

- Software build-up: development, simulation and performance evaluation using the Real Time Workshop embedded with Simulink and Matlab.
- Real-time execution: compile, download, run and debug the generated code on the target navigation computer hardware using Embedded Target for Freescale MPC5xx in conjunction with Real-Time Workshop and Simulink.

Al-Radaideh [9] implemented trajectory tracking using Mehdi path planning algorithm [6] and autopilot design was developed using successive loop closure [1] and implemented a hardware in the loops simulation using of the flight control algorithm using MPC555 embedded system.

For the past year, after the establishment of aeronautics Lab, there are several research and development work carried out in specific areas like avionics development, hardware in the loop simulations, camera integration and target tracking algorithm development, and navigation and path planning. In addition, a team is also working on designing and building a new fuel cell powered aircraft capable of 6 hours endurance flight. This aircraft is developed in-house and will provide a reliable platform for furthering the research in this field.

Mazari aircraft system architecture is shown in figure 1. The system consists of the mission planning layer. In this layer, the mission is defined in terms of the function of the flight. At this level, the mission is designed to use the payload either for monitoring, target tracking, or other functions designed by the user. At this level, the output of the mission layer are the number of waypoints and the task associated with each waypoint like loiter, visit the waypoint, or avid the waypoint. The second layer is the path planning layer. This layer is responsible for coming up with the flight plan in terms of the waypoints and their tasks and their order. The next layer is to generate a flyable trajectory taking into account the aircraft constraints and the aircraft state at each waypoints. The next layer is the trajectory tracker with the outcome of this layers the set points for the autopilots and the state transitions between various tasks or autopilots to accomplish the mission.

Mazari aircraft is equipped with a Gimbaled Camera payload. The gimbals mechanism has Azimuth and Elevation degrees of freedom. Target-tracking, target-following, and target-geolocation were implemented. The autopilot controls the gimbals in order to maintain a target at the center of video input, and command the aircraft to loiter around a fixed target, or follow a moving target. These features are implemented by using an off-the-shelf commercial system, namely "Kestrel AP". And this autopilot has associated software, called "On-point targeting" [10]. This off-the-shelf commercial hardware and software is used to benchmark in-house development of a reliable open source vision system. The current on-point system has been used and studied and its inner-workings, limitations, and shortcoming are understood.

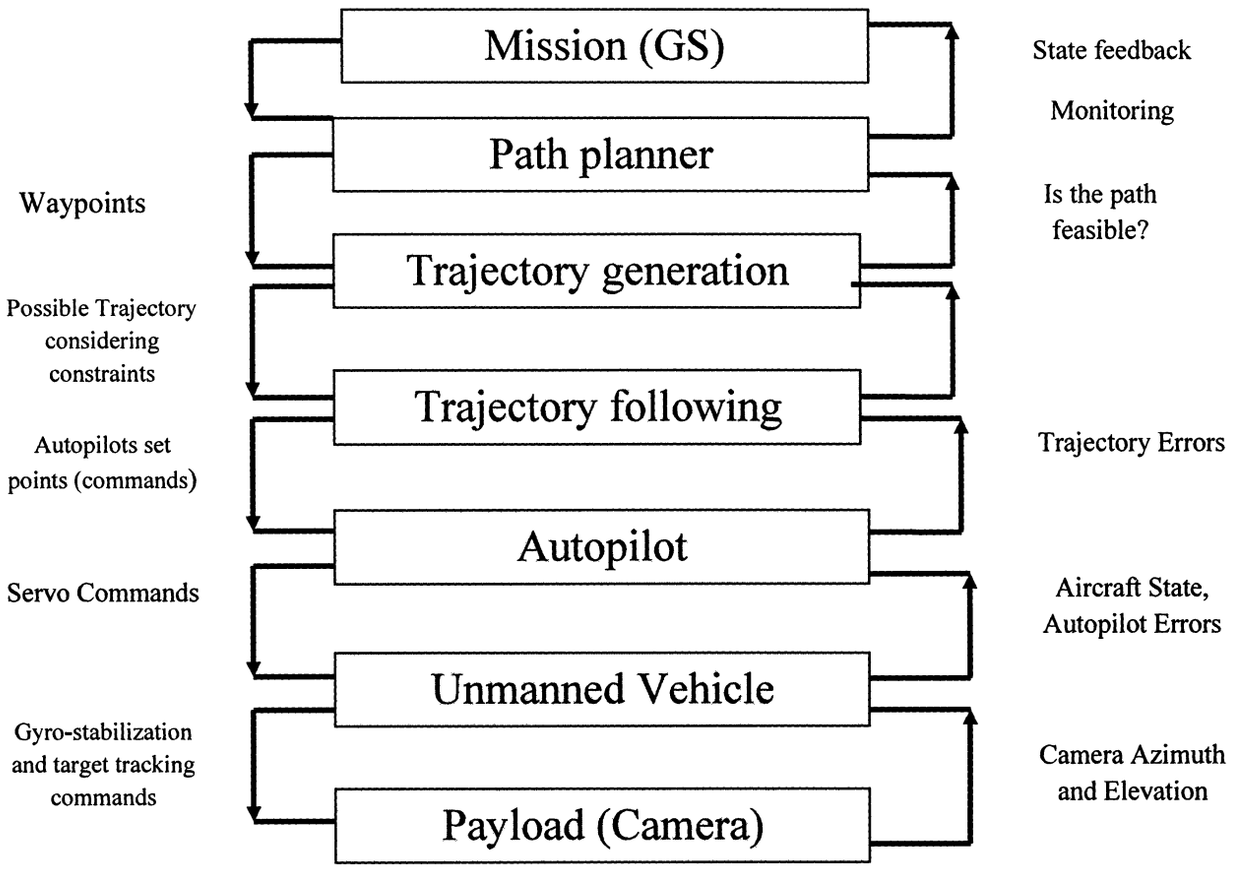

Figure 1: The UAV architecture

2. Mazari UAV Model Development

The nonlinear equations of motion of a rigid aircraft are obtained using linear aerodynamics forces and moments models. The model is based on first order Taylor series analysis of the nonlinear aerodynamics forces and moments. The objectives of the system identification are to identify the terms of the Taylor series expansion of these forces. The propulsion model of the aircraft was evaluated experimentally as shown in figures 2 and 3 [9].

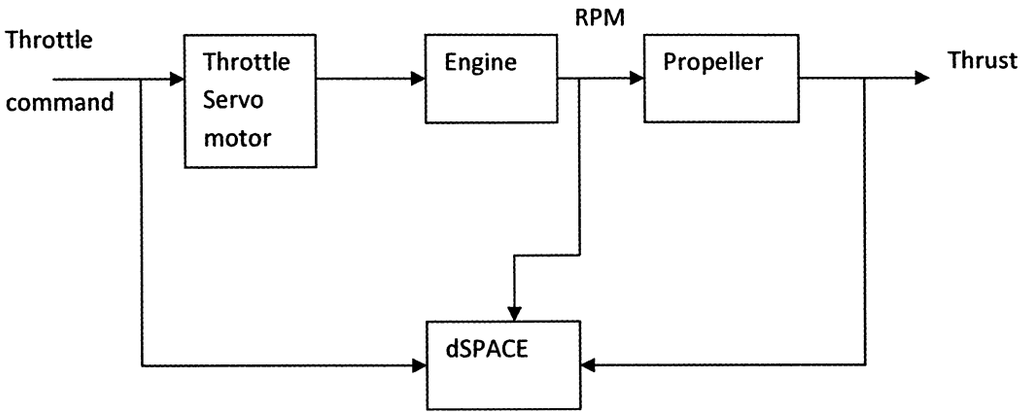

Figure 2: dSPACE interface for the Engine identification

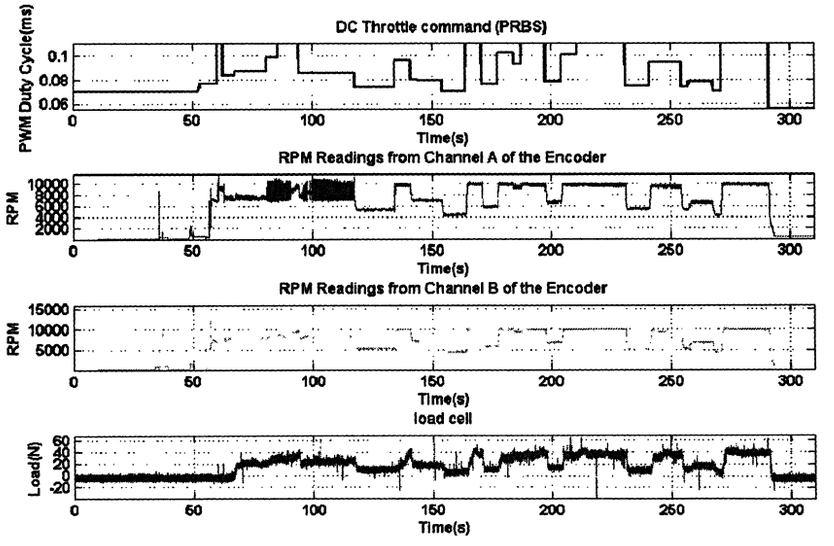

Figure 3: Propulsion system identification input-output data collected

After acquiring the experimental data, Matlab identification toolbox was used to identify two transfer functions one for the engine and one for the propeller using ARX parametric model.

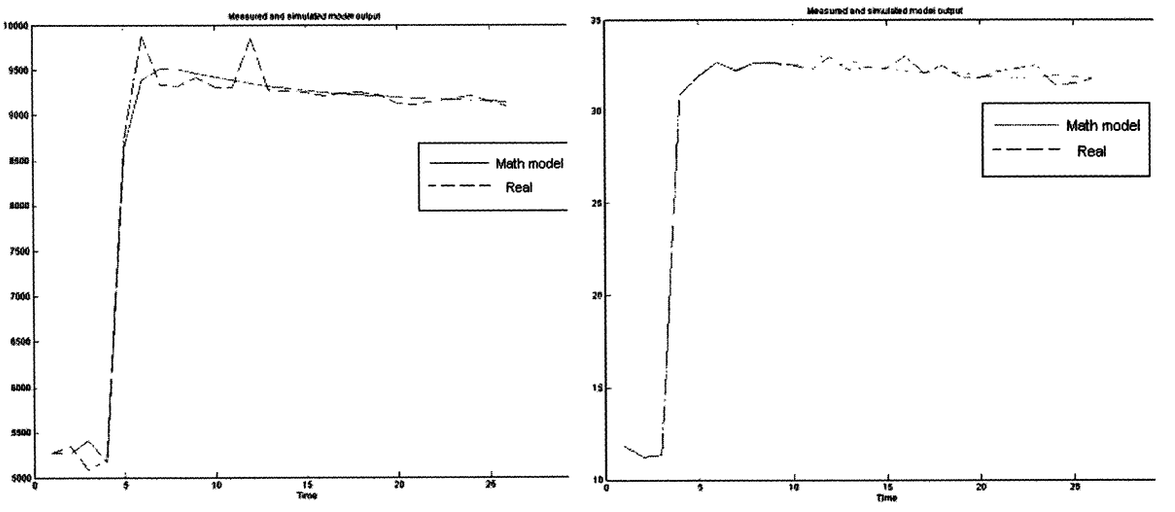

Figure 4: Engine response. Left figure is RPM response while right figure is thrust in Newton.

Inertia property of the aircraft body platform is evaluated using pendulous test setup as shown in figure 5 according to the procedures referenced in [4]. Environment model will determined the property of air, as well as gravity force that will interact with the motions of the aircraft. All of this can be seen in figure 6.

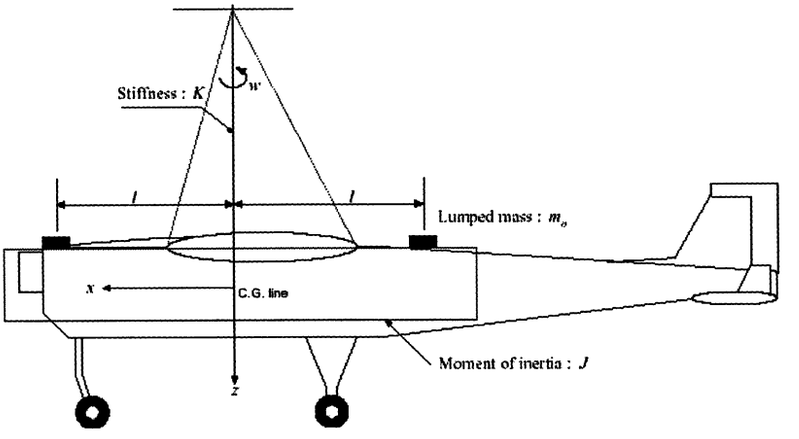

Figure 5: Modeling as a simple torsional pendulum [4]

The experimental aircraft reported in this article is the well known SIG Kadet Senior which was chosen for its excellent performance, low speed, and platform stability allowing stable imagery process with excellent performance. SIG Kadet Senior aircraft geometry and mass properties, as well as the cruise trim conditions can be seen on table 2 below. Inertia properties estimated are based on the test setup shown in figure 5.

Table 2: Aircraft Parameters

Parameter	Value	Dimension
weight	6	kg
Height	100	m
rho	1.117746	kg/m3
Cruise Speed	15	m/s
Temperature	40	deg C
CGx	0.44	m
Cgy	0.02	m
S (wing area)	0.7326	m2

Parameter	Value	Dimension
CGz	0.01	m
Ixx	1.9942	kgm2
Iyy	2.3982	kgm2
Izz	2.75311	kgm2
Jxz	0.0132	kgm2
MAC	0.37	m
b (wingspan)	1.98	m
Xcg from nose	0.46	m

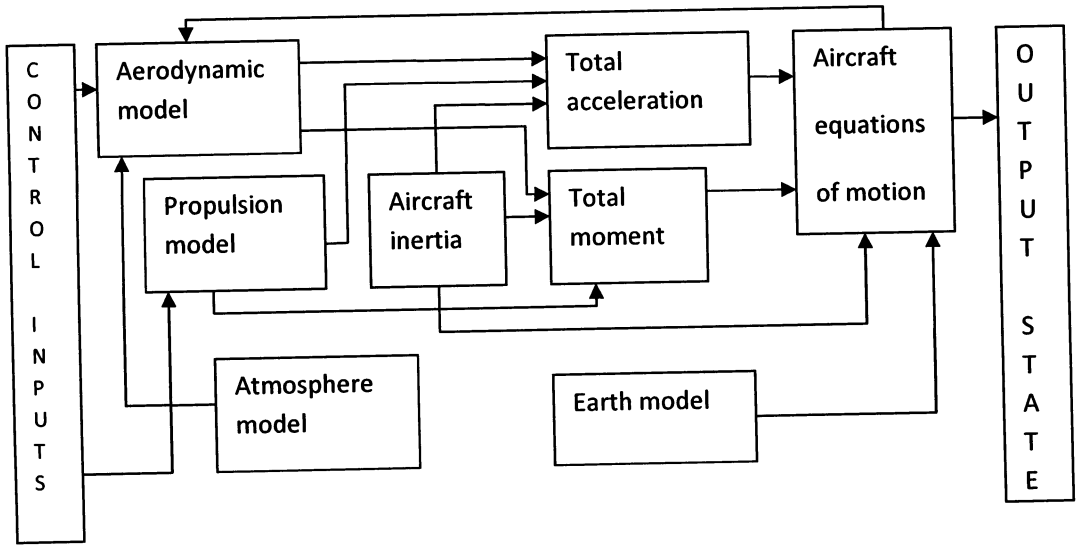

Figure 6: Non linear 6-DOF model

Aerosim blockset [11] is used in the modelling of nonlinear dynamics of the aircraft. Aerodynamic model is developed using stability derivatives calculated based on geometry and shape of the body of the aircraft [12]. This is the example of the calculation of stability coefficient for the aircraft, for lateral dynamics.

$$C_{y\beta} = -\mu \frac{S_e}{S} C_{L\delta e} \left(1 + \frac{d\sigma}{d\beta}\right)$$

$$C_{yp} = C_L \frac{AR + \cos swp}{AR + 4\cos swp} \tan swp$$

$$C_{yr} = -2\left(\frac{l_v}{b}\right)\left(C_{y\beta}\right)_{tail}$$

$$C_{y\delta a} = 0$$

$$C_{y\delta r} = \frac{S_v}{S}\tau(C_{L\alpha v})$$

Autopilots are designed using successive loop closure based on linearized aircraft model. Based on the autopilot function, a trim condition is selected and linearization procedure is followed to generate the corresponding linear model. Small disturbance assumption also used, so that the small disturbance velocities, such as w and v, can be converted in terms of α and β. The linear equations will be converted into state space equation for both longitudinal and lateral-directional dynamics. Each element of the state space representation of the aircraft is filled with nondimensional parameter of the stability derivatives that was defined [12]. Example of nondimensional parameter shown below:

$$Y_\beta = \frac{QSC_{y\beta}}{m};\ Y_p = \frac{QSbC_{yp}}{2mu_0};\ Y_r = \frac{QSbC_{yr}}{2mu_0};\ Y_{\delta a} = \frac{QSC_{y\delta a}}{m};\ Y_{\delta r} = \frac{QSC_{y\delta r}}{m}$$

This coefficient will form State Space representation of matrices :

$$\begin{bmatrix}\Delta\dot{\beta}\\ \Delta\dot{p}\\ \Delta\dot{r}\\ \Delta\dot{\varphi}\end{bmatrix} = \begin{bmatrix}\frac{Y_\beta}{u_0} & \frac{Y_p}{u_0} & -\left(1-\frac{Y_r}{u_0}\right) & \frac{g\cos\theta_0}{u_0}\\ L_\beta & L_p & L_r & 0\\ N_\beta & N_p & N_r & 0\\ 0 & 1 & 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\beta\\ \Delta p\\ \Delta r\\ \Delta\varphi\end{bmatrix} + \begin{bmatrix}0 & \frac{Y_{\delta r}}{u_0}\\ L_{\delta a} & L_{\delta r}\\ N_{\delta a} & N_{\delta r}\\ 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\delta a\\ \Delta\delta r\end{bmatrix}$$

For longitudinal motion, the state space representation can be seen :

$$\begin{bmatrix}\Delta\dot{u}\\ \Delta\dot{w}\\ \Delta q\\ \Delta\dot{\theta}\end{bmatrix} = \begin{bmatrix}X_u & X_w & 0 & -g\\ Z_u & Z_w & u_0 & 0\\ M_u + M_{\dot{w}}Z_w & M_w + M_{\dot{w}}Z_w & M_q + M_{\dot{w}}u_0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\Delta u\\ \Delta w\\ \Delta q\\ \Delta\theta\end{bmatrix} + \begin{bmatrix}X_{\delta e} & X_{\delta t}\\ Z_{\delta e} & Z_{\delta t}\\ M_{\delta e} + M_{\dot{w}}Z_{w\delta e} & M_{\delta t} + M_{\dot{w}}Z_{w\delta t}\\ 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\delta e\\ \Delta\delta t\end{bmatrix}$$

System identification was developed using minimum least square method where the nonlinear equations of Forces and Moment was obtained from measurement of the states and estimation of some angle parameter [2].

Identification of the stability derivatives used to estimate the forces X,Y,Z were obtained from measured acceleration, including thrust effect for x direction, and moment calculation was derived from six degree for freedom equation of motions with all the parameter are measured

such as angular rates p,q,r, inertia, and angular accelerations. Example of this procedure can be seen in equations below. (X).

$$C_X = \frac{m}{qS} a_m^x - \frac{T}{qS}$$

$$I_y \dot{q} + (I_x - I_z)pr + J_{xz}(p^2 - r^2) = M$$

$$C_m = \frac{M}{qSb}$$

This Forces and moment coefficients was also describes as an expansion of linear terms of the aircraft states and control input (z) :

$$C_X = C_{xo} + C_{x\alpha}\alpha + C_{xq}\frac{\bar{c}}{2V_T}q + C_{x\delta e}\delta e + C_{x\delta t}\delta t$$

$$C_m = C_{mo} + C_{m\alpha}\alpha + C_{mq}\frac{\bar{c}}{2V_T}q + C_{m\delta e}\delta e + C_{m\delta t}\delta t$$

Both term can be formed as :

$$x\theta^T = z$$

Where θ are unknown coefficient parameter and these will be solved in minimum least square terms

$$\hat{\theta} = (X^T X)^{-1} Xz$$

This algorithm was tested in the Aerosim blockset nonlinear simulation which gives the result as tabulated in table 3.

Table 3: Linear model versus SID parameter

STATES	EXPLANATION	System ID SIM	LINEAR MODEL SIM
Cybeta	sideslip side force derivative	-0.1911	-0.1596
Cyp	roll rate side force derivatives	-0.0021	-0.1589
Cyr	yaw rate side force derivatives	0.0644	0.0712
CyAileron	aileron deflection side force derivative	7.8061e-4	0
CyRudder	rudder deflection side force derivative	0.0544	0.1097
Cz0	trim z-axis force coefficient	0.027	0.0304
Czalpha	alpha z-axis force derivative	-5.7161	-5.7270
Czq	pitch rate z-axis force derivative	-3.3978	-3.3978
CzElevator	elevator deflection z-axis force derivative	-0.3762	-0.38154

Cl0	trim roll moment coefficient	0	0
Clbet	sideslip roll moment derivative	-0.2109	-0.2113
ClP	roll rate roll moment derivatives	-0.9484	-0.95
ClR	yaw rate roll moment derivatives	0.1373	0.1375
ClAileron	aileron deflection roll moment derivative	0.150	0.151
ClRudder	rudder deflection roll moment derivative	0.0038	0.0039
Cm0	trim pitch moment coefficient	0.001	0
Cmalpha	alpha pitch moment derivative	-0.7492	-0.8911
CmQ	pitch rate pitch moment derivative	-3.4570	-4.0960
CmElevator	elevator deflrction pitch moment derivative	-0.74311	-0.8834
Cn0	trim yaw moment coefficient	-9.43E-06	0
Cnbeta	sideslip yaw moment derivative	0.3180	0.3329
CnP	roll rate yaw moment derivatives	-0.0674	-0.0675
CnR	yaw rate yaw moment derivatives	-0.2508	-0.2515
CnAileron	aileron deflection yaw moment derivative	-0.0469	-0.0470
CnRudder	rudder deflection yaw moment derivative	-0.2284	-0.2288

Using the state space representations, the dynamics modes of the aircraft in longitudinal and lateral-directional results are shown below:

Table 4: Longitudinal mode

MODE	EIGENVALUE	DAMPING RATIO	FREQUENCY (rad/s)
Phugoid	-0.01130+ 0.561i	0.0202	0.561
Short Period	-5.38 + 6.54i	0.635	8.47

Table 5: Lateral mode

MODE	EIGENVALUE	DAMPING RATIO	FREQUENCY (rad/s)
Spiral	-0.00183	-1.00	0.00183
Dutch Roll	-1.65 + 7.86i	0.206	4.98.036
Roll Subsidence	-2.28	1.00	2.28

This UAV has stable flight characteristics in both longitudinal and lateral dynamics. This section discusses the PID control structure developed to improve stability and control the aircraft. To fly the aircraft autonomously, the autopilot must be capable of navigating waypoints.

3. Mazari UAV Flight Controls Developments:

3.1. Autopilots

Autopilots consist of stability augmentation system, attitude hold controller, altitude hold, speed hold, and trajectory follower. Stability augmentation system is needed to improve the handling quality of the aircraft, such as roll subsidence and short period longitudinal mode. Attitude hold controller is used to control the heading, altitude, and airspeed of the aircraft. For manual control, it is required that the autopilot also accepts pitch and roll angle commands. To accomplish this, a controller constructed of nested PID loops has been developed. The aileron, elevator, and throttle commands are controlled via inner PID loops that dampened the high angular rates of the aircraft. The altitude and heading are controlled with outer loops, which produce commanded values for the inner loops. These nested loops are called Successive Loop Closure (SLC) design. Figure 7 and 8 are the representations of this SLC algorithm for longitudinal and lateral directional control. After inner loop controller has been designed, the trajectory tracking controller can be implemented. The autopilot control is divided into two controllers: the lateral controller and the longitudinal controller

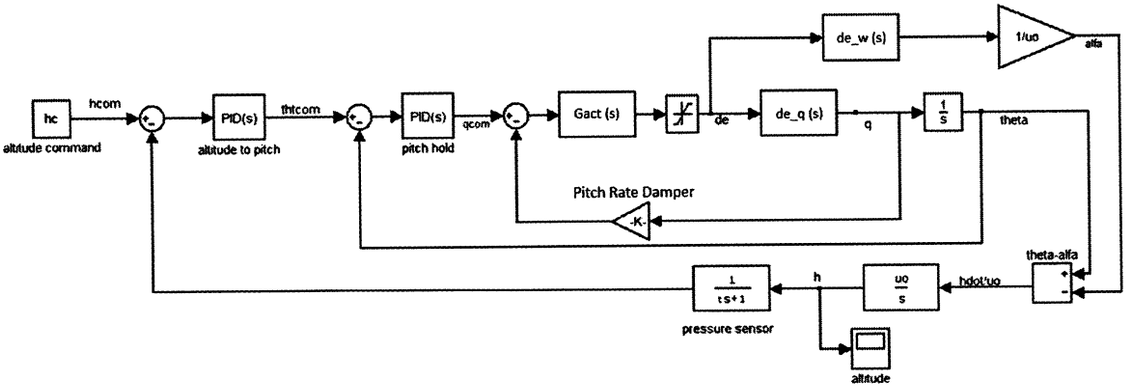

Figure 7: Pitch altitude hold longitudinal SLC control loop

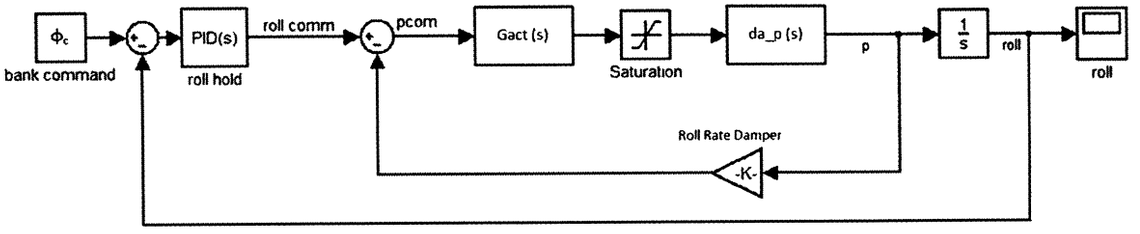

Figure 8: Bank angle hold lateral SLC control loop

3.2. Hardware In the Loop Simulation (HILS) setup and results

Hardware in the loop simulation is used to cut the cost of developing the flight control system and rapid development of the flight control system software. Testing of Unmanned Aerial System is very expensive and involves risk of many crashes prior to the successful development of the flight control system. Flight test requires extensive preparations and high risk of failure, especially during the test of autopilot controller response. Real time simulation provides more safe approach in testing this condition, and there are some categories of simulations that can be developed for these purposes. Rapid prototyping involves real model or plant that is tested using simulated control. AUS has developed HILS setup for the purpose of autopilot controller development and testing. This includes the development of nonlinear 6-DOF aircraft model using aerosim blockset with the stability derivatives derived from the flight test identification results and autopilot control algorithms using successive loop closure. HILS setup is also used to develop the guidance laws for trajectory tracking and waypoints navigation based on the use of vector field path following algorithm [13]. The aircraft control law that are used for this purpose involving longitudinal and lateral control that has been developed before. The main objective of this simulation is to test the vector field path following with different aircraft platform and controller gains.

AUS HILS setup involving two major parts: Simulated aircraft model, which has been developed using aerosim blockset, and the dSPACE real time system (figure 9). This system has compatibility with simulink, in which the aircraft model was built in. The autopilot system, implemented o MPC 555 embedded controller, is interfaced with the dSPACE platform representing the virtual UAV to get the aircraft sensors data. The Autopilot interfaced with a transceiver to communicate with the ground station. MPC555 microcontroller used as the flight control computer, in which the autopilot control laws and gains are embedded. The flight control computer communicates with the ground station to receive the flight path and transmits the aircraft state states and autopilot configurations thru wireless modem.

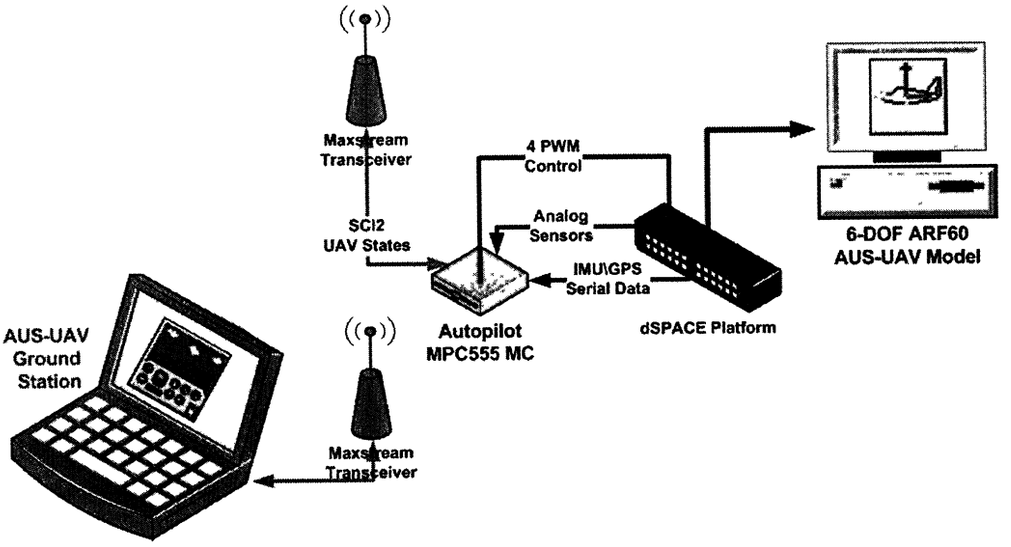

Figure 9: AUS HILS setup [6, 9]

Figure 10 shows the attitude angels read from the dSpace system representing Euler angels of the aircraft, speed and altitude as a response to the elevator input. Figure 11 is the command from controller to the aircraft simulator. HILS result with implementation of vector field path following seen from figures 10-13. There is four set of manoeuvre that has to be accomplished to obtain hourglass trajectory, as it can be seen in roll angle response of the aircraft. These manoeuvres consist of two right banking and two left banking manoeuvre. There is slight change in elevator and throttle command to maintain speed and velocities during this manoeuvre, so that altitude and airspeed was maintained when the aircraft following trajectories. The low frequency oscillation of the pitch attitude hold system was due to the altitude hold controller to maintain the height. The choice of vector field gains affects trajectory-tracking performance results. This will results in steady state error path during some of the flight path.

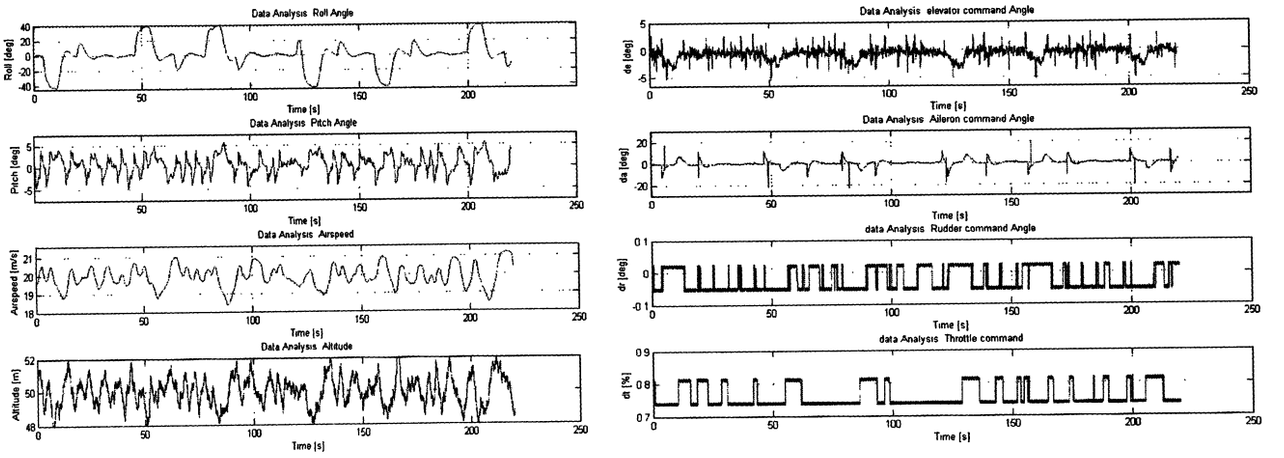

Figure 10: Euler Angles, altitude, and airspeed from HILS

Figure 11 Flight control command to HILS

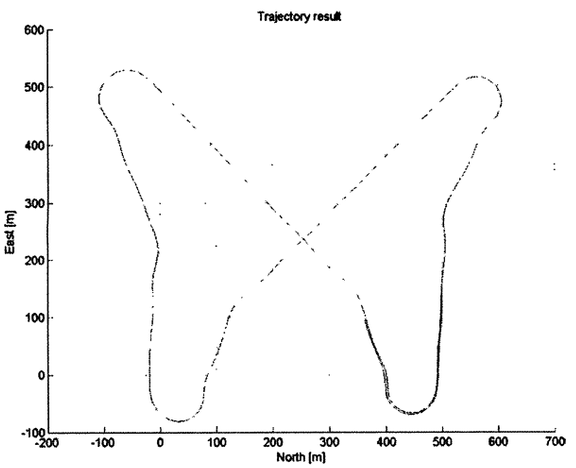

Figure 12: 2D Path Tracking in HILS

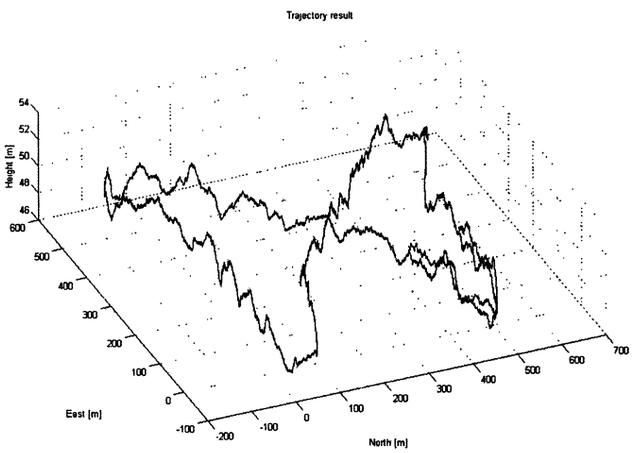

Figure 13 3D Path tracking in HILS

4. Mazari UAV Camera Payload

4.1. Vision System formulation

Target geolocation is an essential mechanism that allows the system to perform target-tracking and target-following. The mathematical formulation for target geolocation forward kinematics as well as inverse kinematics is formulated and simulated in Matlab. The forward kinematics predicates the knowledge of the position of the target, as well as the task space variables. The inverse kinematics predicates the knowledge of the position of the target, and calculates the joint-space variables (i.e. Azimuth and Elevation). The purpose of the "Target Geolocation" model is to map images from the camera to actual location in the real-world [10].

The coordinate frames associated with the camera. The camera coordinate frame represented by {Xc, Yc,Zc} has origin at the camera center and its elements have units of meters with Xc pointing up in the image, Yc pointing right in the image plane, and Zc directed along the optical axis. The frame {Xim, Yim,Zim = Zc −f} is centered at the image plane and has units of meters. The frame (Xip, Yip) is centered in the upper left hand corner of the image and has units of pixels. The transformation from pixel to world coordinates is carried out as follows [10]:

$$\bar{q} = C\, p_{obj}^{c}$$

$$p = \left[CT_g^c T_b^g T_v^b T_I^v\right]^{-1} \lambda q$$

$$\lambda = \begin{bmatrix} L & 0 & 0 & 0 \\ 0 & L & 0 & 0 \\ 0 & 0 & L & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$p_{cc}^{I} = \left[CT_g^c T_b^g T_v^b T_I^v\right]^{-1} \lambda \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Where C is camera calibration matrix:

$$C = \begin{bmatrix} 0 & f_x & O_x & 0 \\ -f_y & 0 & O_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where fx and fy are the scaled focal lengths and 0x and 0y are the pixel coordinates of the center of the image. T_i^j is the transformation matrix from frame i to frame j and (I,j)= b (body), c (camera), g (gimbal), I (inertial), v (vehicle) frames. Finally, p_{cc}^{I} is the focal center of the lens in the Inertial frame.

Using the flat earth model, the "z-component" of p must be zero. To satisfy this condition, L has a solution as follows: $L = \frac{p_{cc}^{I}(z)}{p_{cc}^{I}(z) - p_{obj}^{I}(z)}$

4.2. Simulation of Camera

The Geolocation algorithm assumes a pinhole camera model, flat-earth model of the terrain, and the existence of bias, in addition to white Gaussian noise on all the inputs that are measured using sensors. A model of the terrain is necessary to be able to map a 2D-image, onto a 3D-

world. The flat-earth model is the most common model used in UAV's, although be it the least accurate. The flat-earth model fails to capture the details of the terrain such as mountains and valleys. On the other hand, a round-earth model is unnecessary, since the UAV is capturing only a small area on earth's surface which if we ignore the details of the terrain is indeed flat. The vision system architecture implemented is shown in figure 14.

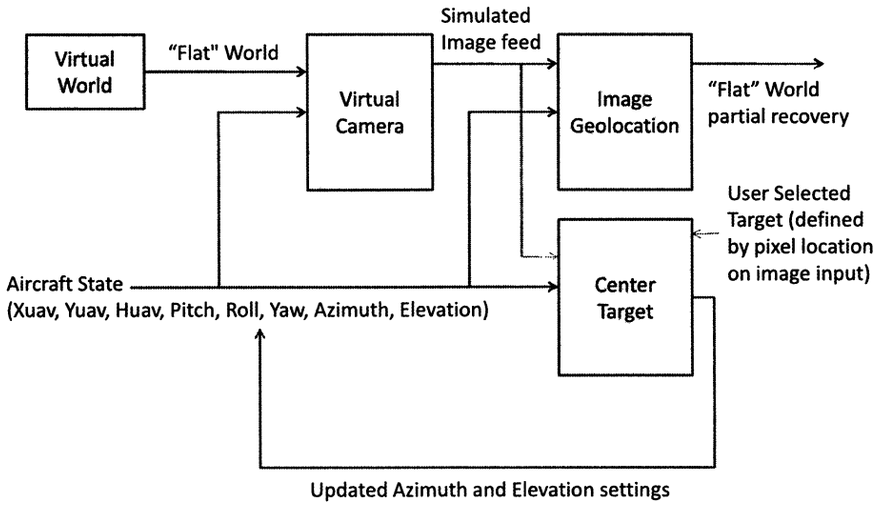

Figure 14: Vision system development architecture

The simulation assumes a flat-earth model of the world, with an aircraft state defined by position, orientation, and Gimbal orientation. First, it starts by defining a "virtual 2D-world environment". This world is defined by an image, and inside the code, we can define the world-coordinate that the lower-left corner of the image corresponds to, and the world-coordinate that the upper-right corner of the image corresponds to. Two functions were developed to calculate the transformation from pixel to world coordinates, and world to pixel coordinates conversion. The convention used is that the letter "p" denotes world-coordinates, and the letter "q" denotes pixel coordinates. Both functions takes information about the position and orientation of the aircraft, as well as the current commanded position for the Gimbal as input, and uses this information to calculate the world-coordinate given a pixel coordinate or vice versa. Note that the function returns only the x and y coordinates, because the z-coordinate is assumed to be zero. "L" is the image depth, which is defined as z-component of the object in the camera-frame. Some parameters are assumed to be constants; those include the focal length of the camera, the principal point of the image.