NUS
National University of Singapore

FACULTY OF ENGINEERING

Autonomous Landing Unmanned Aerial Vehicle

Fong Jun Wen

Department of
Mechanical Engineering

In partial fulfilment of the
requirements for the Degree of
Bachelor of Engineering
National University of Singapore

Session 2008/2009

Summary

This thesis presents the system architecture for landing an Unmanned Aerial Vehicle (UAV) from a hovering position without the intervention of a human operator. Through the use of feedback information from a height sensor, the UAV is commanded to perform controlled descent with the desired landing parameters by implementation of the flight control laws.

The plant model of the system was determined in order to simulate the system using Matlab, Simulink. Through the use of simulations, the variables of the controllers are varied to determine the most appropriate gains that will result in the most preferred landing profile.

In this project, the vertical component which controls the climb and descends of the platform was isolated from the roll, pitch and yaw through the use of a jig. Therefore, this current project only commands the height of the platform but can be fully expanded to command the roll, pitch and yaw with addition sensors such as rate gyro.

Acknowledgement

The author would like to express his appreciation to the project supervisor, Associate Professor Gerard Leng Siew Bing, for the opportunity to work on this project, as well as for his patient guidance in the various aspects of the project.

The author would also like to extend his sincere appreciation to the following people for their assistance during the course of this project:

1. Mr Muhamad Azfar bin Ramli, Graduate student of the COSY lab, for his kind assistance to the various problems that was encountered during the course of this project.
2. Ms Amy Chee, Ms Priscilla Lee, Mr Cheng Kok Seng and Mr Ahmad Bin Kasa, Staff of the Dynamics & Vibration lab, for their help and support with necessary equipments.
3. Mr Ronald Tan Han Rong for his support and assistance during the construction phase of the coander effect flying saucer.

Contents

List of Figures

List of Tables

List of Symbols

h	Altitude, displacement from ground
$\dot{h}$	Altitude rate change
e	Error
K_p or K_c	Proportional gain
K_i	Integral gain
K_d	Derivative gain
τ_I	Integral controller scaler
τ_D	Derivative controller scaler
P_u	Ultimate period
G_c	Controller transfer function
G_{OL}	Open loop transfer function
G_{CL}	Closed loop transfer function
t_r	Rise time
t_p	Peak time
t_s	Settling time
M_p	Maximum overshoot

1. Introduction

1.1. Objective

This project aims to develop an autonomous landing system that will enable a UAV to land autonomously without the interference of a human operator. The scope of this project was limited to the altitude control only, without the intervention of the Roll, Pitch and Yaw motion. However, the developed concepts and control laws can also be further extended to encompass Roll, Pitch and Yaw control with additional sensors such as the rate gyro or the tilt sensor.

Some other objectives are the selection and testing of a suitable altitude sensor, the implementation of the flight control laws using the Proportional-Integral-Derivative (PID) controller, the tuning of the PID gains using various methods and lastly a flight demonstration to validate the autonomous landing system.

1.2. Historical background

The concept of UAV began during the American Civil War, when the North and the South attempted to bomb each other’s ammunition depot by launching balloons carrying explosive device which would be released at a controlled timing. However, the actual beginning started during World War II, when a company Chance Vought Aircraft had proposed building missiles with landing gear, in order to save cost.

Recently, UAV such as the Global Hawk and the Predator, have achieved considerable popularity, when it was employed to provide aerial surveillance as well as attack missions in Afghanistan. There are many other useful applications of UAV such as homeland security, crop dusting and traffic monitoring.

UAV can be classified under two distinct categories, Fixed-wing and Rotary. Some examples of Rotary UAV include helicopter, Micro Air Vehicle and Organic Air Vehicle, whereas the Global Hawk and Predator represents Fixed-wing aircraft.

According to ReinHardt [2], the next generation of UAVs will be smaller in size, more affordable, easier to train and more precise than the existing UAVs. Also, UAVs are expected to be capable of detecting nuclear, biological & chemical weapons, looking into double canopy jungles and provide low-cost, reliable communications and data relay across the battlefield. In urban built-up areas, where airspace is often limited, Vertical Take-Off and Landing (VTOL) UAV is often employed.

The remote piloting of a VTOL UAV is a very challenging task which requires great operator skill and attention [4]. Furthermore, most maneuvers would require the pilot to maintain full visual contact with the UAV at all times, especially during the landing phase. In addition, some other factors that might affect the Remote control (RC) performance are poor positional accuracy, poor altitude accuracy and pilot fatigue [4].

2. Theory

2.1. Helicopter Aerodynamics

The four working states of a rotor in axial flight are described in Appendix II. In particular, the vortex ring state should be avoided during descent as it might result in highly unsteady flow with nonlinearity as affirmed by Yaggy & Mort (1963). This unsteadiness can cause significant blade flapping, uncommanded drop in descent rate, loss of control effectiveness, and excessive thrust fluctuations.

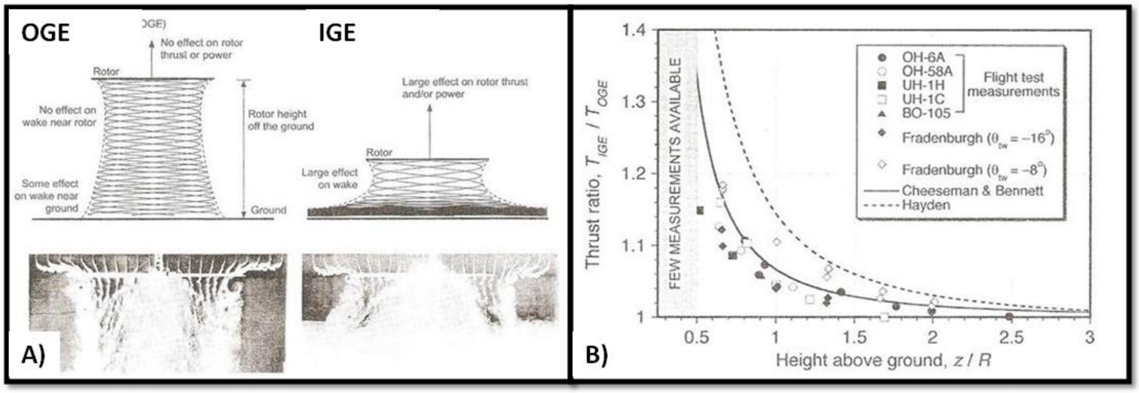


Figure 1: (A) Wake behaviour OGE/IGE; (B) Thrust ratio Vs Distance

Ground effect can also affect the performance of a helicopter. Figure 1A shows the wake behavior from a hovering state In and Out of ground effect and figure 1B shows the increase in thrust ratio at different hovering height as tested by Fradenburgh (1972) and Hayden (1976). These results suggest significant effects on hovering performance for heights of less than one rotor diameter, where there is a sharp increase in thrust.

2.2. PID System

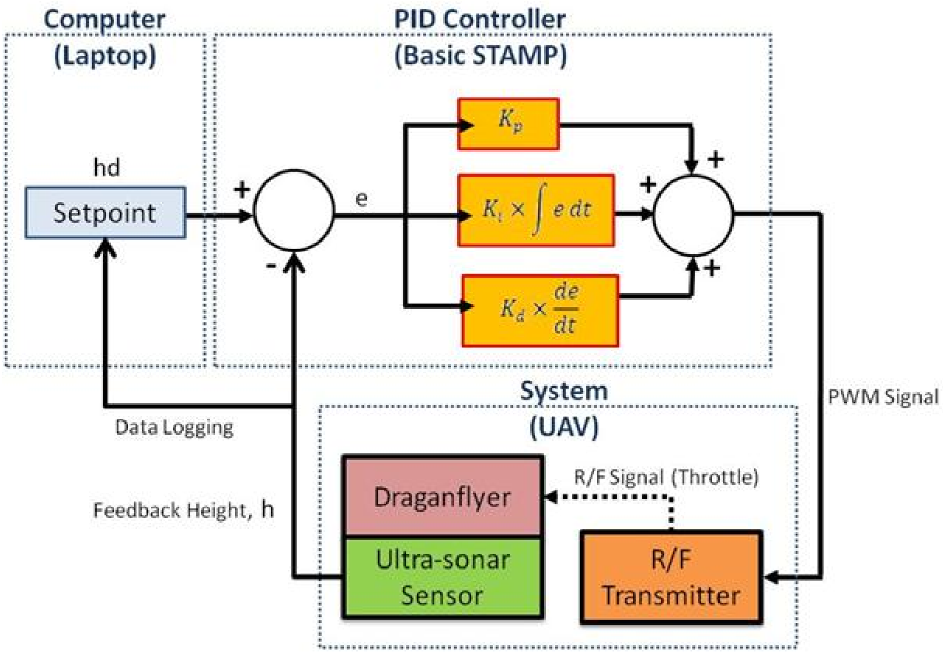


Figure 2: Block Diagram of PID Control system

The classical control theory with a closed loop PID controller is used to control the altitude of the UAV as illustrated by the block diagram in figure 2. The ultrasonic sensor attached to the UAV provides range feedback for the closed loop system and its data is also captured by the computer for data logging purposes. Several methods of tuning the gains of the controller are proposed.

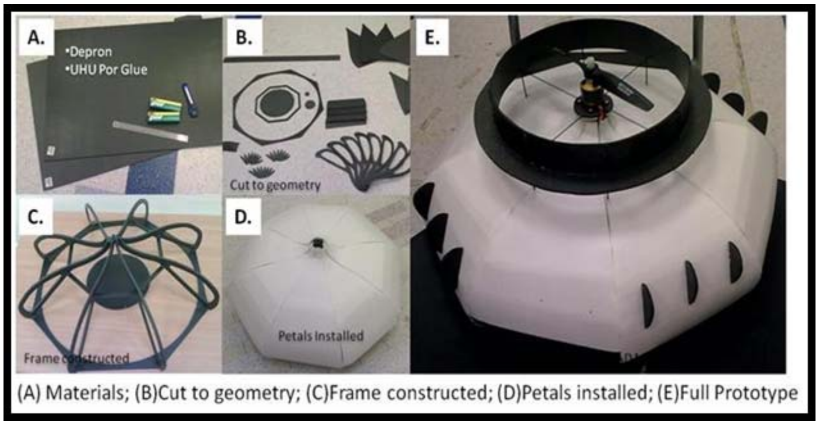
3. Hardware and Setup

3.1. UAV Platform

For this project, a rotary type platform was selected as it is able to maintain a hovering position before the landing command is initiated. During the initial stage, a hovercraft which uses coander effect to produce lift was explored. However, due to the nature of this project, as well as the problems encountered, it was eventually replaced with another off the shelf platform, the draganflyer V Ti.

3.1.1. Coander Effect Flying saucer

The phenomenon which causes flow near limiting surface to follow the geometrical shape of these surfaces is known as the Coander effect. According to researcher Jean Louis Naudin, the VTOL platform was able to attain good control, stability and thrust which is comparable or even better than a conventional RC helicopter. In accordance with his built plan, a similar platform was modelled using CAD software, Solidworks in Appendix I, figure 29, and built carefully as shown in figure 3.



(A) Materials; (B)Cut to geometry; (C)Frame constructed; (D)Petals installed; (E)Full Prototype

Figure 3: Construction of the Coander Effect UAV

However, subsequent tests conducted on the platform concluded that the coander effect was unable to produce sufficient lift as promised. Detailed experiments that were conducted to verify the lift, as well as some minor modifications to the UAV are attached to Appendix V for reference.

3.1.2. Draganflyer Ti V

The flight controls of the Draganflyer operate solely on differential thrust between the front-rear and left-right motor, whereby a net resultant moment can be generated along the roll axis (x-axis) or the pitch axis (y-axis) to produce a roll or a pitch motion as shown in figure 4.

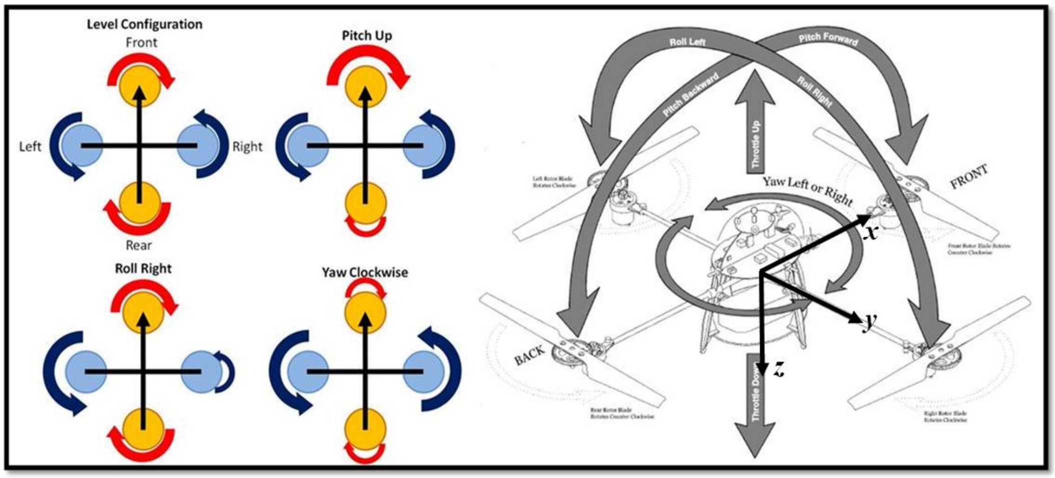


Figure 4: Flight control operation of the Draganflyer V Ti

For instance, to pitch the UAV up, the front motor will rotate faster than the rear motor. Another advantage of this platform is that no mechanical linkage is required as there is no control surface.

3.2. Test bed construction

In order to investigate and tune the entire UAV system, different jigs with different degree of freedom (DOF) are required. For example, in order to tune the PID controller for the roll motion, a test bed which restricts the UAV motion to the roll axis only is required. Generally, the PID controller must be tuned separately for the individual roll, pitch and yaw axis [11].

For this project, the PID controller that must be tuned is in the translational z-direction, with reference to figure 4. Therefore, a test bed with 1DOF in the translational z-direction is required. The test-bed design is modelled using Solidworks as shown in Appendix I, figure 30. The dimension of the ceiling frame is modelled according to the actual dimension in the laboratory. The UAV is mounted on a carbon fibre crossbar, and the crossbar is joined to the vertical polystyrene supports by four prismatic joints, which allows translational displacement along the vertical supports only. Therefore, this setup can effectively constrain the UAV to the translational z-direction as shown in figure 5.

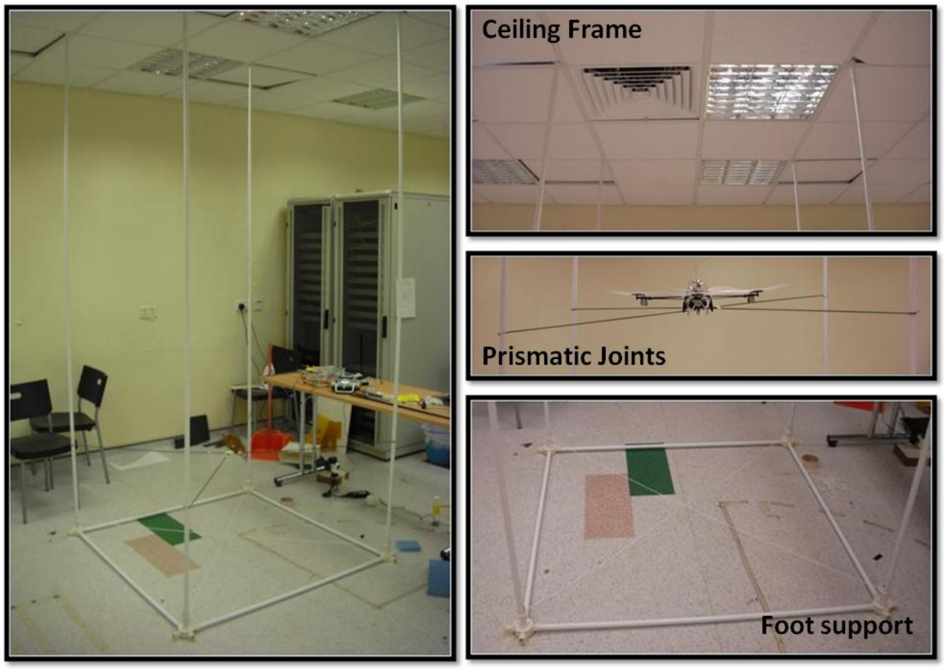


Figure 5: Constructed Test bed

3.3.Sensor with mounting

The PING))) Ultrasonic range sensor is attached to a mounting made of balsa wood and high density foam is used to protect the sensor from impact with the ground.

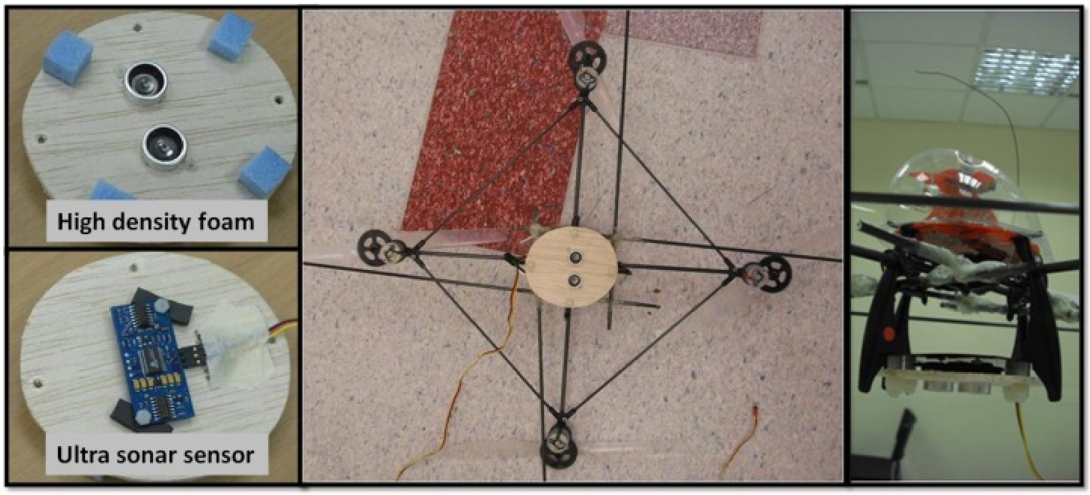


Figure 6: Ultrasonic distance sensor attached to UAV

3.4.Microcontroller: Basic Stamp

The microcontroller Basic stamp BS2px and BS2pe from parallax was used to process the information from the ultrasonic sensor and to generate the required signal.

3.5.Communication

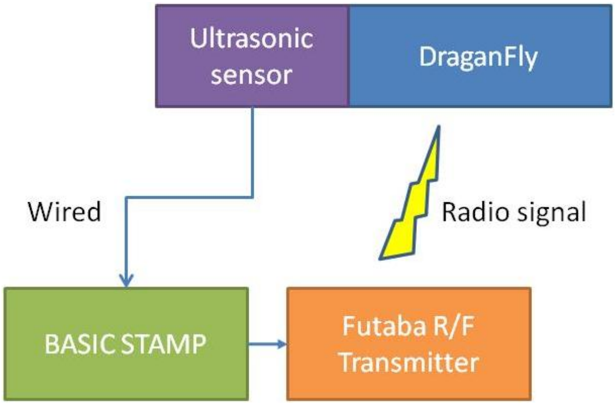


Figure 7: Communication of control system

Figure 7 illustrates the original communications of the landing system. To simplify the problem, the connection between the ultrasonic sensor and the microcontroller was physically wired. Initially, a single microcontroller Basic stamp BS2px was used to calculate the distance from the range sensor as well as processing the data and generating the necessary signal to the Radio frequency (RF) transmitter. The transmitter will then relay the signals to the UAV via the trainer port.

A Pin-Out diagram of the Futaba transmitter is shown in Appendix I, figure 31. In order to control the UAV using the microcontroller, the six pin plug is modified and only three of the pins are used: Pin 2 (PPM Out), Pin 3 (PPM In) and Pin 7 (Ground shield). Pin 2 is used to monitor the signals generated by the transmitter and Pin 3 is used to receive the signals from the microcontroller into the transmitter. The Radio Frequency signal is finally transmitted to the receiver onboard the UAV.

3.6. Problems encountered

3.6.1. Pause width

For basic stamp microcontroller, the command “PAUSE” will cause a delay where no signal will be transmitted to the pin. However, the smallest unit is 1 millisecond whereas the pause required between channels is only 0.4ms. To overcome this problem, another pulse signal was sent to another dummy pin, which will result in an induced delay to the actual pin that is connected to the transmitter.

3.6.2. Trimmed signal

It was observed that the trimmings on the transmitter affect the signals that are generated quite significantly. Appendix I, figure 33 shows the Futaba transmitter with the trim control. Thus, the UAV attached to the test bed should be properly trimmed to maintain hover before the generated signals are measured.

3.6.3. Lag time between Pulse Train

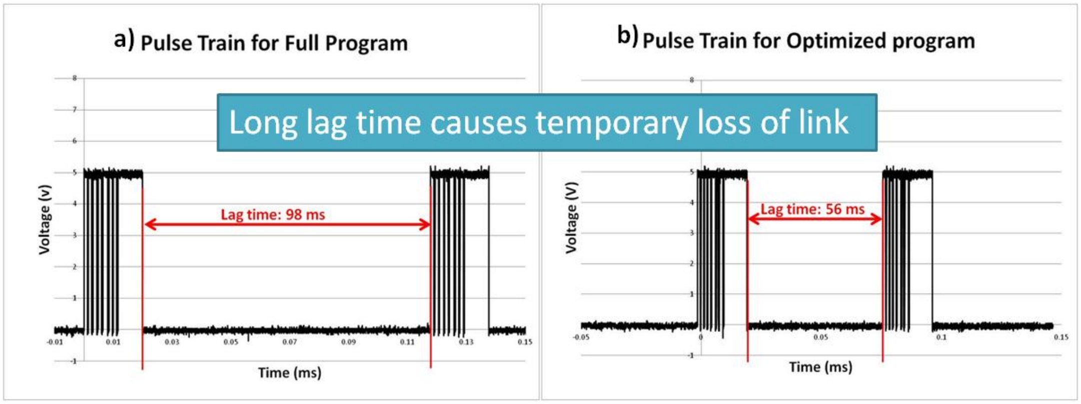


Figure 8: Lag time between pulse train

Figure 8a shows the lag time of 98ms for the full landing program and 8b shows the reduced or optimized program, having a lag time of 56ms. However, it is still far from the delay required of 0.4ms.

To solve this problem, an alternative communication system is required. Figure 9 shows the new communication system that requires two microcontrollers to operate simultaneously. BS2pe was used to collect feedback data from the ultrasonic sensor and acts as the PID controller to processes the retrieved data. It is also connected to the computer which will save the information for data logging purposes. Another microcontroller BS2px was connected to BS2pe through serial communication. The primary function of BS2px is to generate regular pulses of signals to the RF transmitter. This is very crucial as long lag time will register as a temporary loss of link which will cause the UAV to become uncontrollable and twitches violently. Therefore, the fastest stamp BS2px with processing speed of up to 19,000 instructions per second was employed to perform this duty.

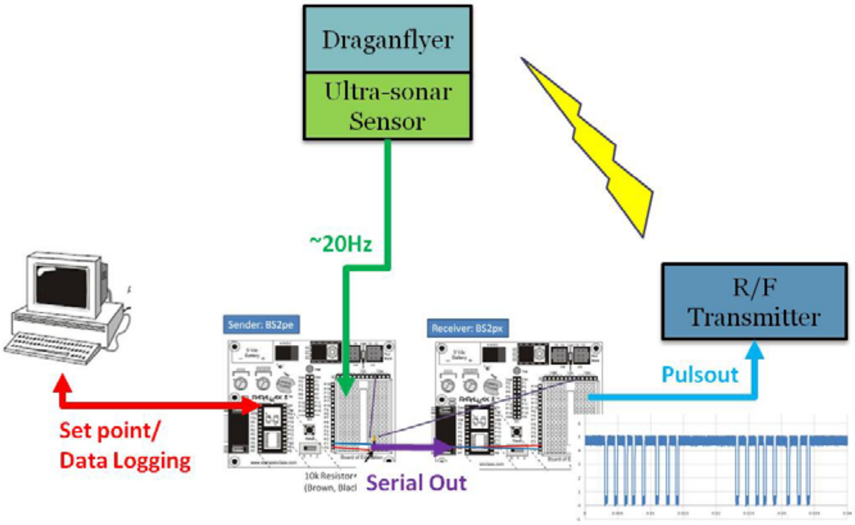


Figure 9: New communication system

Upon further investigation, it was deduced that the ultrasonic sensor, running at about 20Hz, is causing the long lag time. Thus, the usage of two basic stamps can effectively isolate the inherent lag time problem that is caused by the sensor. Appendix I, figure 34 shows the connections between the two stamps with flow control. One of the limitations of Stamp is that when it is sending or receiving data, it cannot execute other instruction and vice versa. The Stamp does not have a serial buffer which is present in other computers. Also, even when running at the highest serial baud rates, there is insufficient time for the Stamp to receive data, process it and execute another receiving data command in time to catch the next stream of data, unless there are significant pauses between data transmissions. Fortunately, flow control can be used to overcome this problem whereby the receiver can tell the sender when it is ready to receive the next stream of data. Also, stamp can only transmit a single byte at a time. Therefore, the signal that will be sent was scaled down by 10 in order to stay within the single byte limit of values between 0-255.

3.7.Programming algorithm

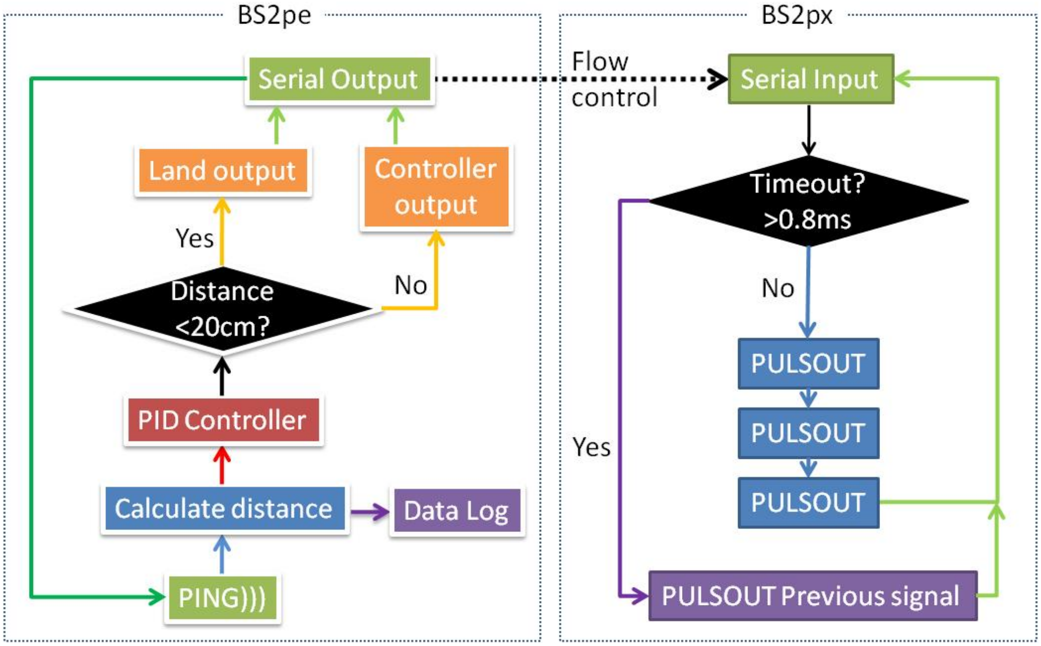


Figure 10: Flow chart of autonomous landing system

The flow chart above summarizes the programming algorithm for the entire autonomous landing system. A timeout function is provided to account for cases where lag time exceeds 0.8ms. In such cases, this function will generate a similar pulse train as the one before. This is to ensure that there is a regular generation of pulse signal to prevent loss of link.

Another condition that is used in this algorithm is to check whether the calculated distance is less than 20cm. If it is true, the throttle which corresponds to landing is initiated instead of the PID controller output.

4. Experiments conducted

4.1. Operation of the Futaba RF transmitter

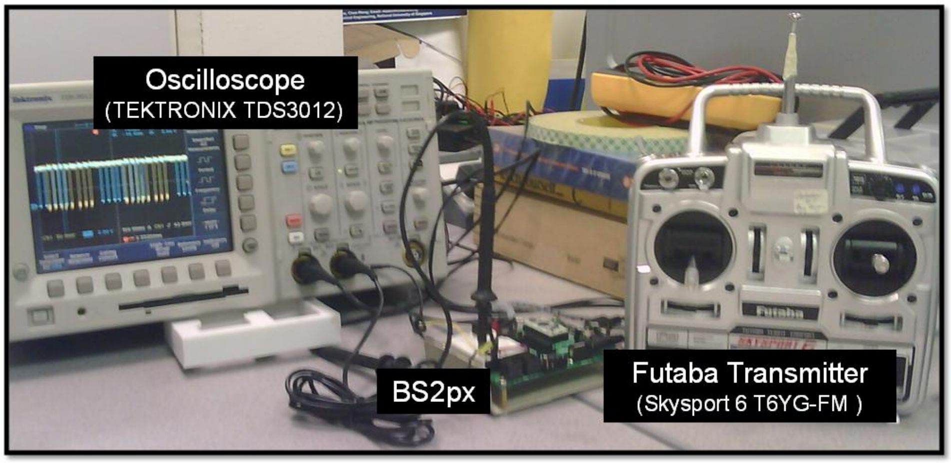


Figure 11: Experimental setup to measure PWM signal

Experiment 1: Measurement of PWM signal from radio transmitter

The signals generated by the transmitter are Pulse Width Modulation (PWM) signals. This experiment involves the measurement of the PWM signal generated by the Futaba RF transmitter, at different stick settings. The equipment that was used in this experiment is shown in figure 11. Using a probe to tap the signal from Pin 2 at the trainer port, the respective pulse width at different stick positions can be measured using a digital oscilloscope (Tectronix TDS3012).

4.2. Experiments on Draganflyer

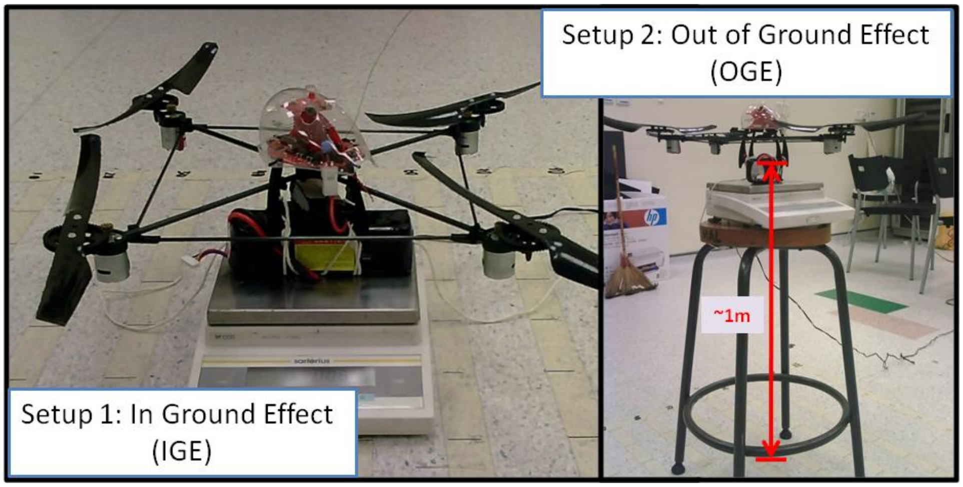

Setup 2: Out of Ground Effect (OGE)
~1m
Setup 1: In Ground Effect (IGE)

Figure 12: Experimental setup to measure generated thrust

Experiment 2: Determine relationship between Throttle and Thrust

A simple experiment was set up to estimate the generated thrust at different throttle setting as shown in figure 12. The UAV was attached to a weight and the platform was placed onto a digital weighing scale. The resultant thrust can be measured by the weighing scale at different throttle positions. Two different setups are designed to measure the thrust generated at two different altitudes. This is done to estimate the thrust generated IGE and OGE. The assumption is that the attached weight and weighing scale do not interfere with the thrust generated by the rotors.

Experiment 3: Testing and calibration of ultra sonar sensor

The Parallax PING))) ultrasonic sensor is used to measure the altitude of the UAV. It is chosen primarily because of its availability in the laboratory as well as the suitable range of 2cm to 3m which satisfy the requirement of this project. Also, it is fully compatible with the microcontroller Basic Stamp. In this experiment, the ultrasonic sensor is tested and calibrated with the actual distance.

Experiment 4: Determine system plant model

The aim of this experiment is to determine the plant model of the system. From a hovering altitude of 100cm, the throttle was stepped down and the height variation data was logged down using the software StampPlot Pro. From these raw data, the rate of descent (R.D) can be determined. The magnitude of the steps was varied and the corresponding R.D was analysed to determine the system's plant model.

Experiment 5: Simulations using Matlab, Simulink

Using the derived system plant model, the system was modelled using Matlab, Simulink as shown in figure 13.

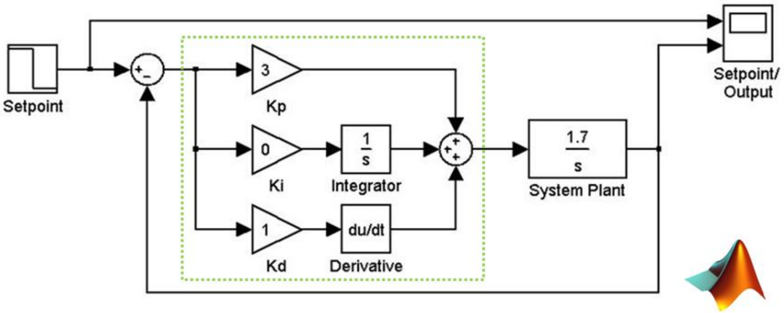


Figure 13: System plant modelled in Simulink

Simulations are carried out to examine the theoretical output response of the system given a step down input. The gains for the PID controller are varied to determine the system response. The two methods of tuning the PID controller, the Ziegler Nichols method and the Trial and error method, was simulated.

Experiment 6: Flight test

Flight 1: Scheduled Proportional control

Two different values of proportional gains are used depending on the height of the UAV. For height of above 45cm, a gain of 2.0 is used and for height below that, a gain of 1.0 is used. This is to ensure that the rate of descent is reduced as it approaches the ground.

Flight 2: Throttle reduction by 20%

The hovering throttle of the UAV was reduced by 20% and the rate of descent of the UAV was investigated.

Flight 3: Proportional-Derivative control

It was determined that a PD controller is sufficient as the integral term which will compensate for steady state error is not required in this autonomous landing system. The appropriate proportional and derivative gains that are obtained from the simulations using Simulink are used in this experiment. Also, the actual system response can be compared with the simulated response.

Flight 4: Hover about set point

Using similar PD controller, UAV was commanded to hover at a desired set point.

5. Results and Discussions

5.1. Testing and calibration of ultrasonic sensor

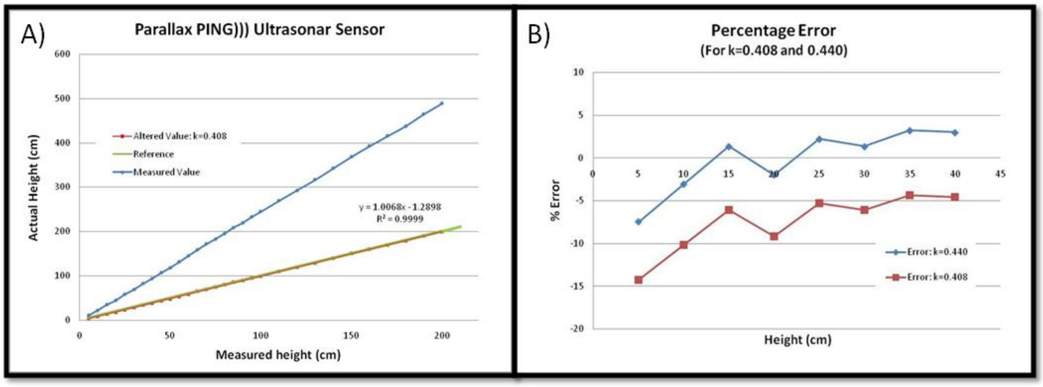


Figure 14: (A) Ultrasonic sensor results; (B) Percentage error below 40cm

It was observed that no calibration of the ultrasonic sensor is required if the Stamp used to receive the sensor data is BS2pe. However, calibration is required if the Stamp used is BS2px. Referring to figure 14A, the measure distance (Blue) is found to be deviated from the actual distance (Green). Since the deviation is linear, a correction factor of 0.408 was determined empirically. The corrected distance is depicted by the red line. It was also observed that the percentage error increases significantly when the sensor was below 40 cm. Subsequently, another correction factor of 0.440 was determined the same way for distance below the range of 40 cm. The difference in percentage error plot is shown in figure 14B.

5.2. Determine relationship between throttle and thrust

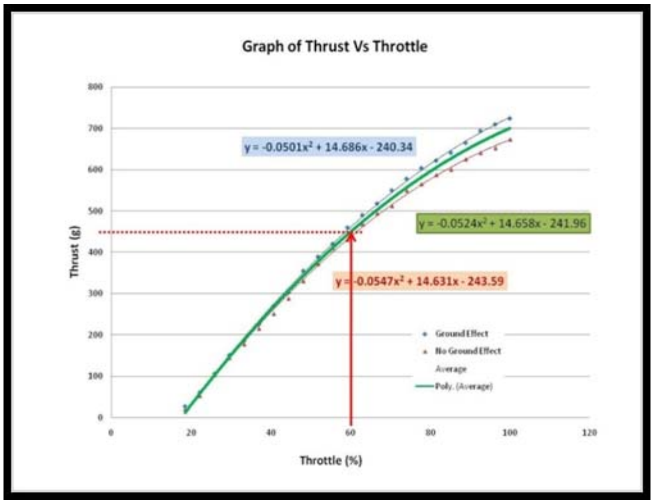


Figure 15: Plot of Thrust Vs Throttle

Figure 15 shows the relationship between the throttle stick position and the resultant thrust produced by the four rotors. With this estimation, the approximate throttle position that is required to descend the UAV of varying weights can be calculated accordingly.

5.3. Measurement of PWM signal from radio transmitter

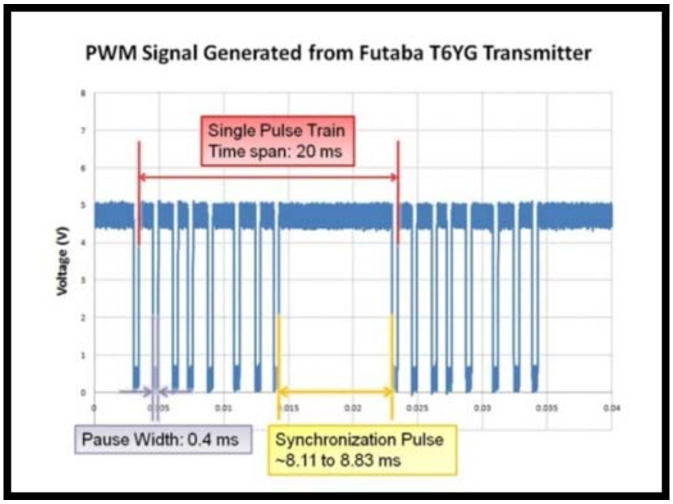


Figure 16: PWM Signals from Futaba transmitter

Figure 16 shows the PWM pulse train that was captured by the oscilloscope. It was observed that the time span for a single pulse train is 20 ms regardless of the stick configurations and there are a total of seven channels. Also, the pause width between different pulses is 0.4ms. However, the synchronization pulse varies from 8.11ms to 8.83ms, depending on the different stick combination. This is a more accurate representation of the synchronization pulse as compared to the constant value of 9ms that was proposed by KarWei [14]. The actual synchronization pulse can be calculated as follows:

$$Sync.\,Pulse = Timespan - \sum_{Ch=1}^{7} Pulsewidth - 8 \times Pausewidth$$

Figure 17 shows the seven pulses that are present in a single pulse train. Each of these pulses corresponds to a channel and therefore, there are a total of 7 channels. Channel 1 controls aileron, channel 2 controls elevator, channel 3 controls throttle, channel 4 controls rudder, channel 5 controls thermal intelligence and lastly channels 6 and 7 controls the landing gear switch and flap knob which are not used in this UAV. Also, the range of pulse width for the throttle control is measured to be from 0.74 ms to 1.4ms.

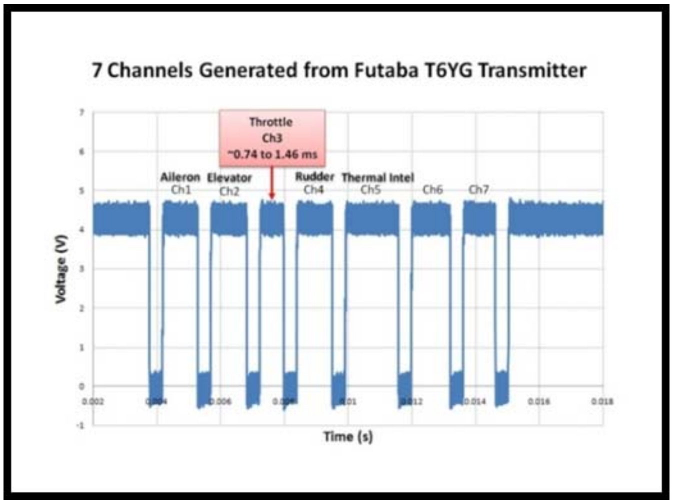


Figure 17: Seven channels from single pulse train

5.4. Relationship between helicopter controls and pulse width

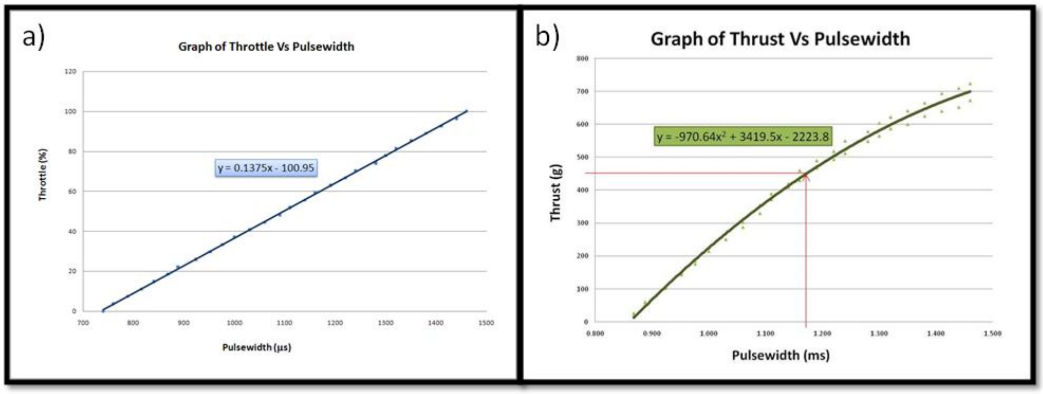


Figure 18: (a) Plot of Throttle Vs Pulse width; (b) Plot of Thrust Vs Pulse width

Since this project involves the manipulation of the throttle control, the different throttle positions which correspond to the pulse width of channel 3 is examined in detail. The result of Percentage throttle against Pulse width is shown in figure 18a. Using information from section 5.2, the plot of Thrust against Pulse width is obtained and plotted in figure 18b. From this result, the required thrust can be generated with the correct pulse width.

The rest of the pulse widths from other channels that corresponds to the 2 extremes of the stick position are plotted in Appendix I, figure 35. For instance, full aileron right corresponds to a minimum pulse width of 0.78 ms and full aileron left corresponds to a maximum pulse width of 1.42ms.

5.5. Determine system plant model

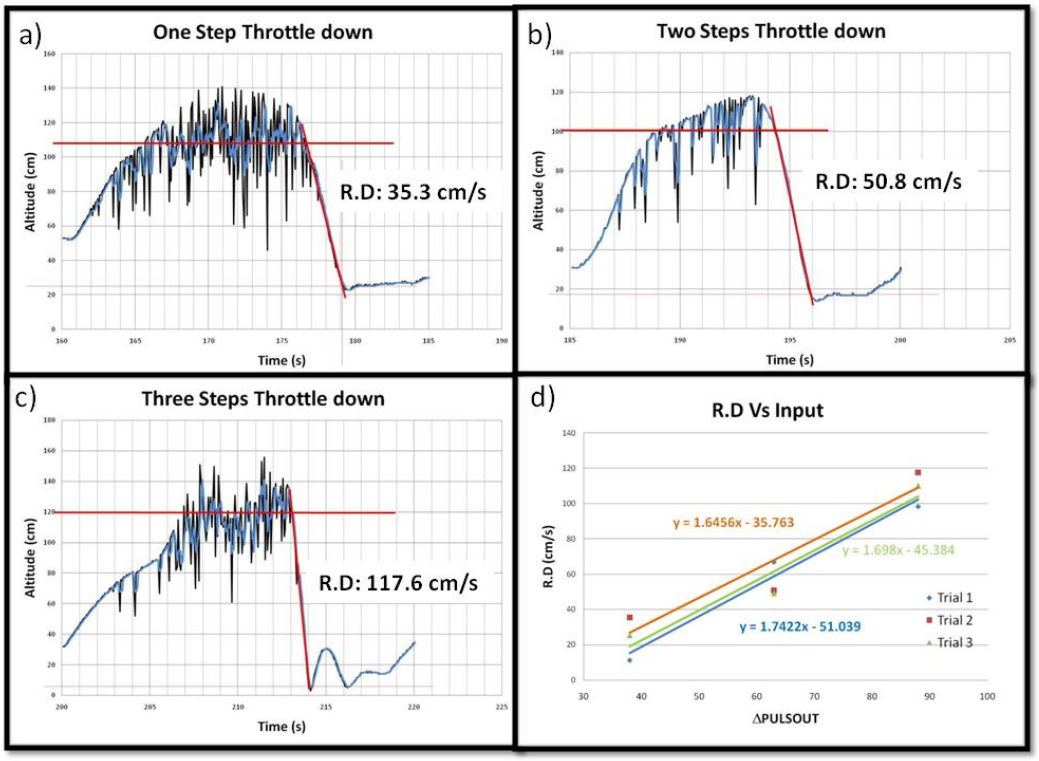


Figure 19: Plots of Rate of descent Vs Pulsout

The descent rates at different step throttle down in shown in figure 19a-c. A total of three trials are conducted and results are plotted in figure 19d. The system's plant model can be derived from the gradient of the line which is about 1.7.

The relevant equations to derive the plant model are shown below:

$$\dot{h} = K.\Delta Throttle$$

where $R.D$ can be represented by $\dot{h}$ and $\Delta Throttle$ represented by $u(t)$

$$h(t) = K.u(t) \rightarrow \text{Laplace transform} \rightarrow sH(S) = K.U(S)$$

$$\therefore H(S) = \frac{K}{S}U(S)$$

Since K is 1.7 , the plant model can be represented by the block diagram below.

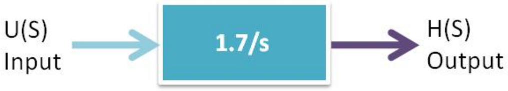


Figure 20: System's plant model

5.6. Simulations using Matlab, Simulink

5.6.1. Ziegler Nichols method

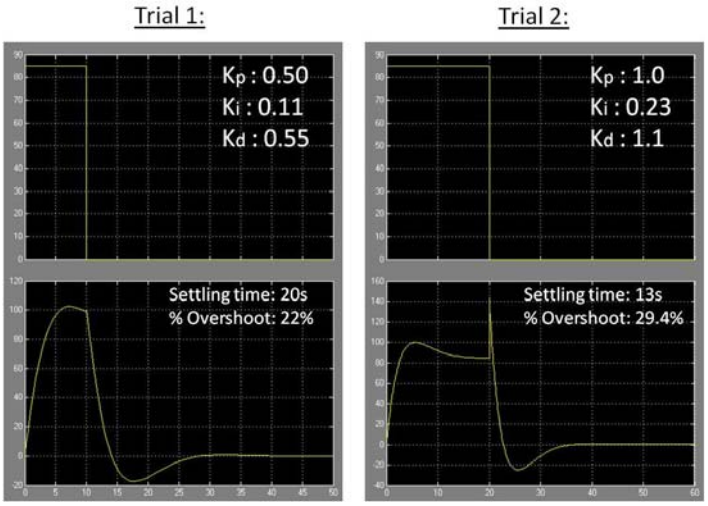


Figure 21: Simulations using Ziegler Nichols method

The simulated results using the Ziegler Nichols method is shown in figure 21. Both results shows high overshoot with comparatively long settling time.

5.6.2. Proportional-Derivative control: Trial and Error method

The simulation results using the PD controller is shown in figure 22. The settling time is much more acceptable compared to the previous method. Thus, this method will be used in the actual flight test.

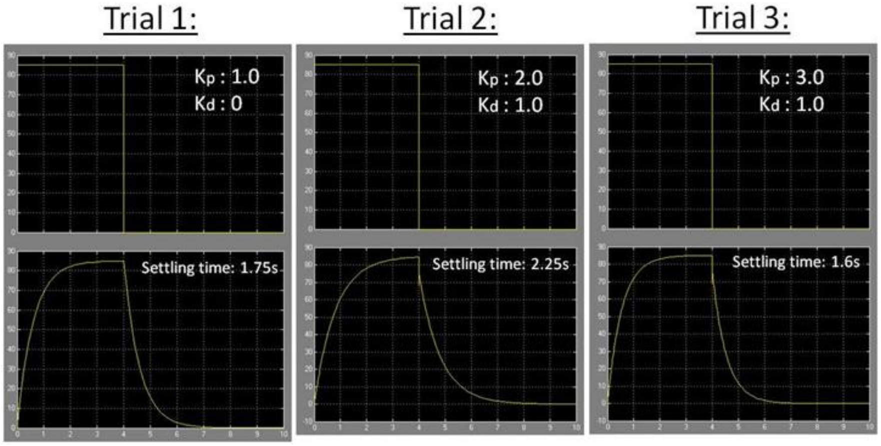


Figure 22: Simulations using PD control

5.7. Flight test

5.7.1. Flight 1: Scheduled proportional control

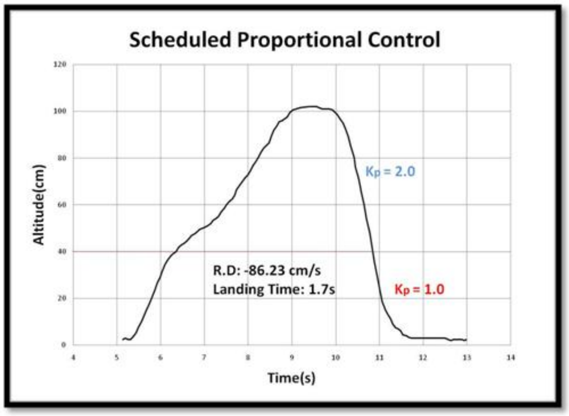


Figure 23: Results using scheduled proportional control

Figure 23 shows the actual flight results using scheduled proportional control method as described earlier. The result is good with fast settling time of about 1.7s and controlled smooth landing.

5.7.2. Flight 2: Throttle reduction of 20%

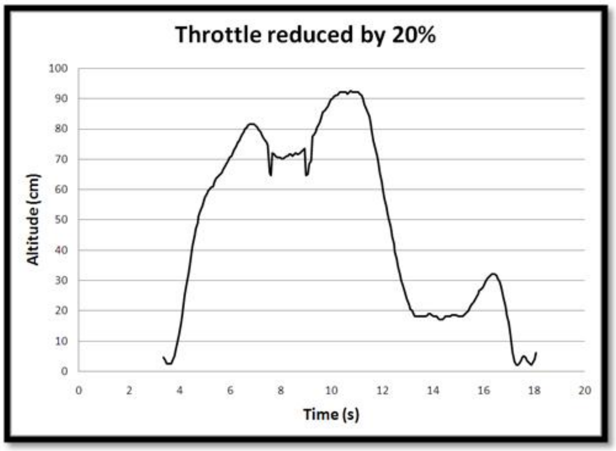


Figure 24: Results using throttle reduction method

The landing using throttle reduction of 20% shows similarly smooth descent. However, the UAV hovers to a point of about 20cm where it encounters ground effect. As a result, manual control is required to force it to land.

5.7.3. Flight 3: Proportional-Derivative control

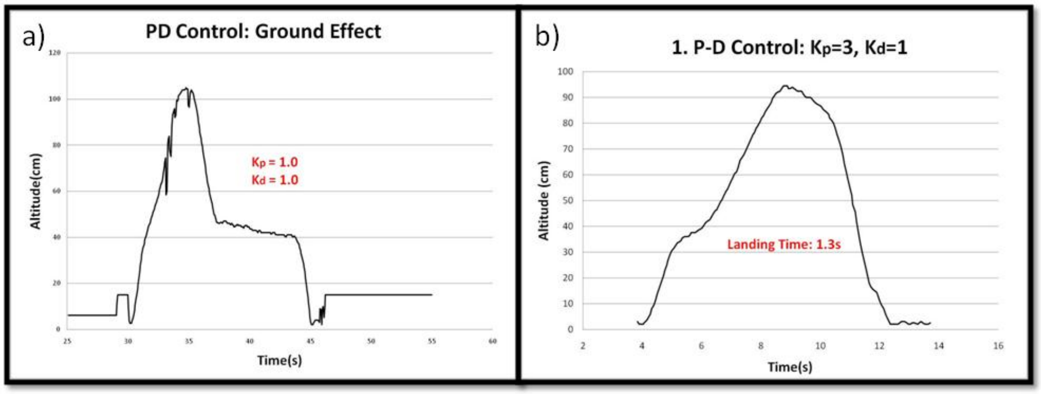


Figure 25: Results using PD control: (a) Kp:1.0/Kd:1.0; (b) Kp:3.0/Kd:1.0

The same ground effect is encountered using the PD controller with proportional gain of 1.0 as shown in figure 25a. For this case, the UAV hovered to a height of about 40cm and refuses to descend any further. This can be overcome by increasing the proportional gain to 3.0 in figure 25b where the landing is smooth with quick settling time of about 1.3s.

5.7.4. Flight 4: Hover to set point

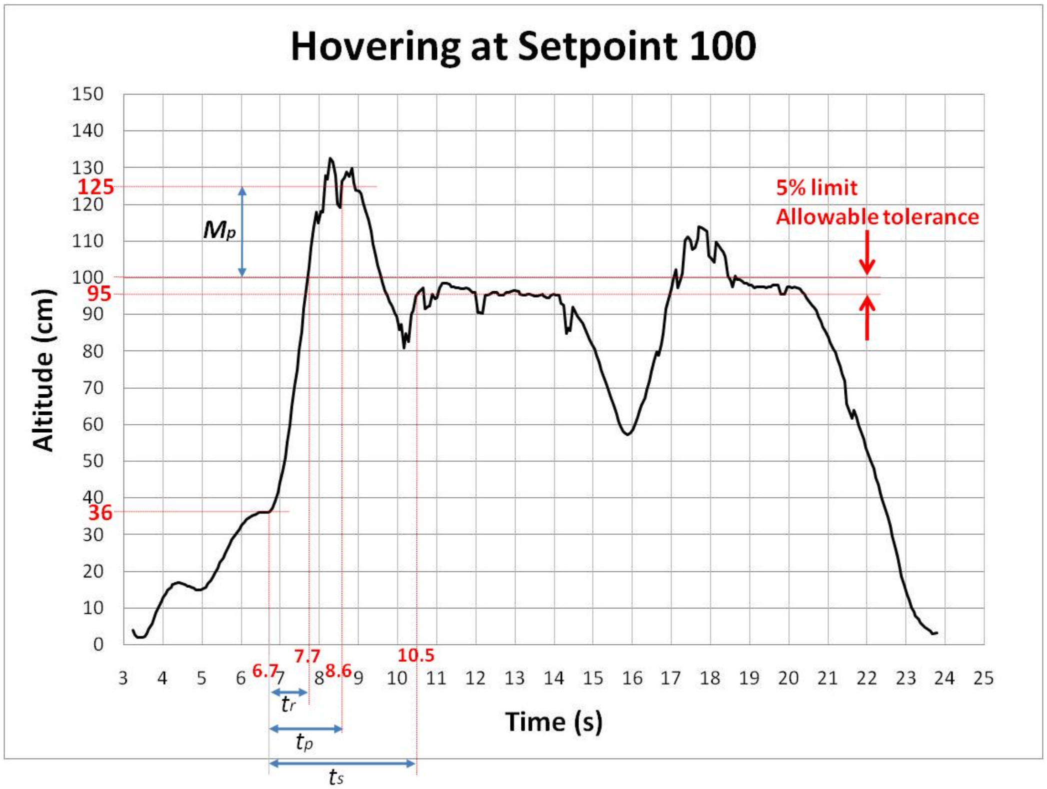


Figure 26: Hovering to set point using PD control

Using the same PD controller with proportional gain of 3.0 and derivative gain of 1.0, the UAV is commanded to a desired height as shown in figure 26. The command was initiated at a height of 35cm and it rises quickly to the desired altitude after 1 oscillation.

The transient performance is calculated as follows:

Rise time: $t_r = 7.7 - 6.7 = 1.0s$ (From 0% to 100%)

Peak time: $t_p = 8.6 - 6.7 = 1.9s$

Settling time: $t_s = 10.5 - 6.7 = 3.8s$ (Time to reach and stay within 5% limit)

Maximum % overshoot: $M_p = \frac{h(t_p) - h(\infty)}{h(\infty)} \times 100\% = \frac{125 - 95}{95} = 31.6\%$

This system can be further tuned to reduce overshoot by lowering the proportional gain and reducing the steady state error by including an integral term into the controller. However, the steady state error is reasonable as it is within the 5% limit.

5.8. Analysis with simulated plot

5.8.1. Comparing simulation and actual plot

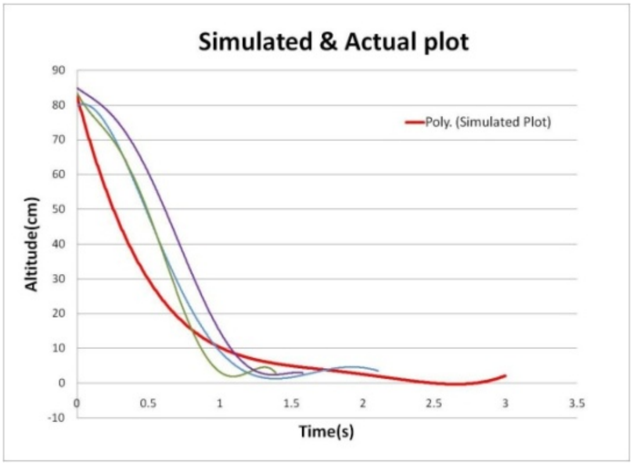


Figure 27: Comparison between simulated and actual plot

The simulated and actual landing profile is plotted for comparison in figure 27. The red line shows the simulated plot using Matlab and the rest of the plots correspond to the actual test flight data. It can be concluded that the actual plots fit the simulated plot reasonably well.

5.8.2. Root locus analysis

Referring to figure 13, the transfer function of the system is derived as:

Assuming unity feedback: $H = 1$

Proportional-Derivative Control: $G_c(s) = K_p + K_d s$

Open Loop transfer function: $G_{OL}(s) = (K_p + K_d s)\frac{K}{s}$

Closed Loop transfer function:

$$G_{CL}(s) = \frac{GH}{1 + GH} = \frac{(K_p + K_d s)\frac{K}{s}}{1 + (K_p + K_d s)\frac{K}{s}} = \frac{KK_d s + KK_p}{(1 + KK_d)s + KK_p}$$

Taking $K_p = 3$, $K_d = 1$, $K = 1.7$,

Closed Loop transfer function: $G_{CL}(s) = \frac{1.7s + 5.1}{2.7s + 5.1}$

The open loop transfer function can also be written in the form: $G_{OL}(s) = K\frac{(s + z_1)}{(s + p_1)}$

where z and p are the zero and pole of the open loop transfer function. As the gain changes, the values of the close loop poles will change and thus the transient response and stability. The root locus plot is a plot of the loci of the close loop poles on the s-plane as the gain K varies from 0 to infinity.

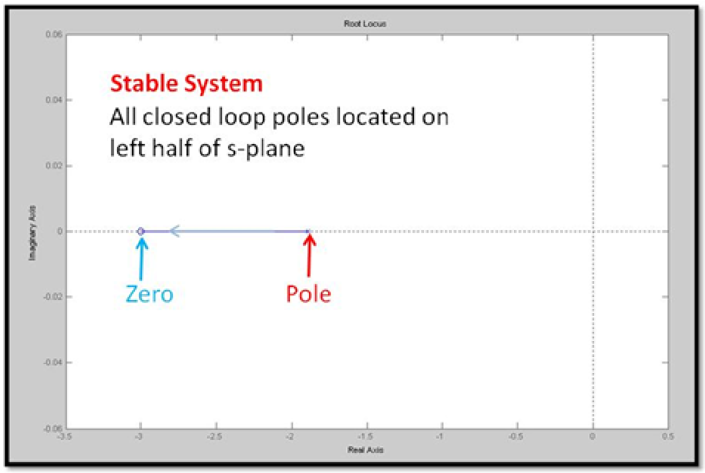


Figure 28: Root locus plot

A root locus is plotted using Matlab as shown in figure 28. The plot shows a stable system as all the closed loop poles are located on the left half of the s-plane.

6. Conclusions

This thesis has presented a complete system to control the height of a UAV using feedback control from an ultrasonic sensor. The method of communication between the different microcontrollers and transmitter were discussed as well as some of the problems that are associated with the communications. Different methods of tuning the PID control system were explored with the aid of simulations using Matlab, Simulink. Finally, actual test flights are carried out to validate the autonomous landing system. All the objectives of this project were well met.

7. Recommendations for Further Work

7.1. Integrating controller for Roll, Pitch and Yaw

In this project, a height sensor is used to provide feedback for the PID controller for altitude control. However, this can be further extended to the roll, pitch and yaw motion. In order to do so, additional sensors are required to provide feedback information about the UAV's orientation or angular velocity. Tilt sensors or rate gyro can be used in such cases to provide compensation for the roll, pitch and yaw motion. In the thesis by KarWei [14], he had successfully used a tilt sensor to compensate for the roll and pitch moment.

Another alternative method is to extract the raw data directly from the three piezo electric gyro that are already present on the circuit board Draganflyer V Ti. However, this would require knowledge about the circuit board to prevent damage on any of the components on it.

7.2. Camera system to locate landing zone

A camera system can be integrated into the UAV to locate the landing zone from another position. In the thesis by Zhikang [17], he had used color segmentation to segment a colored landing zone and a pan and tilt camera to track on to the target. Finally, the position of the UAV was corrected to the landing zone and subsequently landed. In order to communicate between a computer and a RF transmitter, an interface PCTx by Endurance was used.

References

[1] Haniph Latchman (2003, January 17). Brief history of UAVS. Retrieved March 10, 2009, from http://aln.list.ufl.edu/uav/UAVHstry.htm

[2] ReinHardt J.R., James J.E. & Flanagan E.M. (1999). Future Employment of UAVs, Joint Force Quarterly. Retrieved March 10, 2009, from http://www.dtic.mil/doctrine/jel/jfq_pubs/0822b.pdf

[3] Joshua Hintze (2004, April). Autonomous landing of a rotary unmanned aerial vehicle in a non-cooperative environment using machine vision. Brigham Young University, Department of Electrical and Computer Engineering

[4] Tein Hau, Tan (2008, May). Autopilot Unmanned Aerial Vehicle. National University of Singapore, Department of Mechanical Engineering

[5] J. Gordon (2006). Principles of Helicopter Aerodynamics. Cambridge University Press.

[6] A.R.S. Bramwell, G. Done & D. Balmford (2000). Bramwell's Helicopter Dynamics. Elsevier Ltd.

[7] Tomas B. Co (2004, February 13). Ziegler-Nichols Method. Michigan Technological University. Retrieved March 11, 2009, from http://www.chem.mtu.edu/~tbco/cm416/zn.html

[8] J.L. Naudin (2007, February 25). The GFS-UAV project, A Coander effect flying saucer. Retrieved March 11, 2009, from http://jlnlabs.online.fr/gfsuav/index.htm

[9] Chris Smith (2009). The Aerodynamics of a Ping pong ball. Retrieved March 11, 2009, from http://www.thenakedscientists.com/HTML/content/kitchenscience/exp/the-aerodynamics-of-a-ping-pong-ball/

[10] W. Klaus and I.S. Martin (1999, May 20). Misinterpretations of Bernoulli's Law. University Frankfurt, Department of Physics. Retrieved March 11, 2009, from http://user.uni-frankfurt.de/~weltner/

[11] Wai Weng, Kong & M.S.b. Zainal (2006). Design and Control of a Quad-Rotor Flying Robot for Aerial Surveillance. Universiti Teknologi Malaysia, Center for Artificial Intelligence and Robotic.

[12] J. Martin, J. Williams, K. Gracey, A. Alvarez & S. Lindsay (2005, February). BASIC Stamp Syntax and Reference Manual Version 2.2. Retrieved March 11, 2009, from www.parallax.com

[13] M. Azfar (2007). Aerodynamics and Propulsion of an Indoor UAV. National University of Singapore, Department of Mechanical Engineering

[14] Kar Wei, Chin (2007). Flight Dynamics and Control for an Indoor UAV. National University of Singapore, Department of Mechanical Engineering

[15] Martin Hebel (2009). Primer to Using StampPlot® Pro, SelmaWare Solutions. Retrieved March 11, 2009, from http://www.selmaware.com/stampplot/index.htm

[16] Nick Sacco (2002). How the Draganflyer Flies. Rotory Magazine. Retrieved March 11, 2009, from http://www.rctoys.com/pdf/draganflyer3_rotorymagazine. pdf

[17] Zhikang, Lin (2008). Camera-based vision system for unmanned air vehicles (UAVs) operations. National University of Singapore, Department of Mechanical Engineering

Appendix I. Figures and Tables

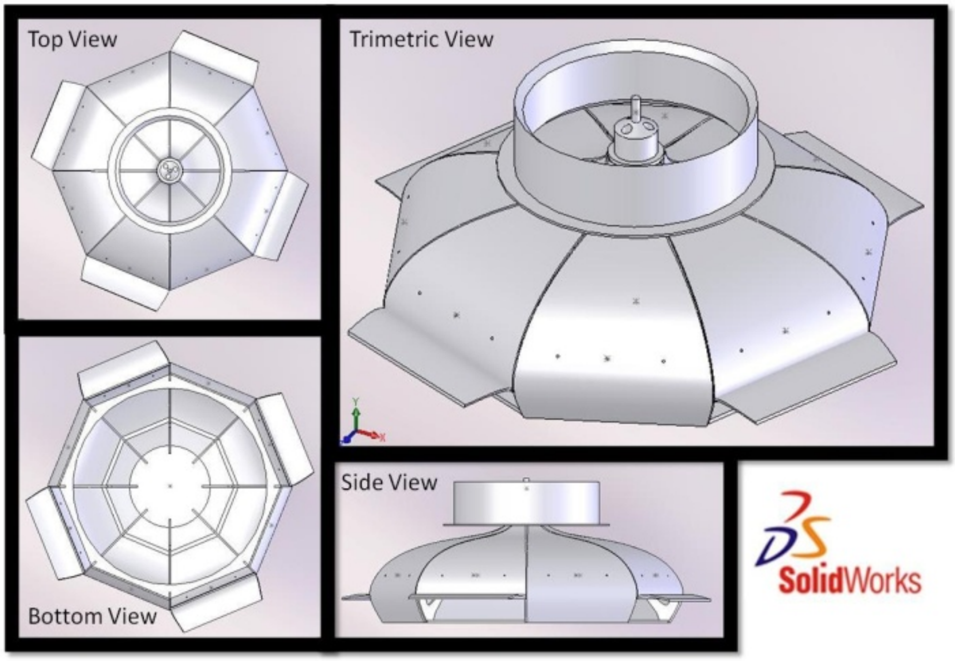


Figure 29: Model of Coander effect UAV using Solidworks

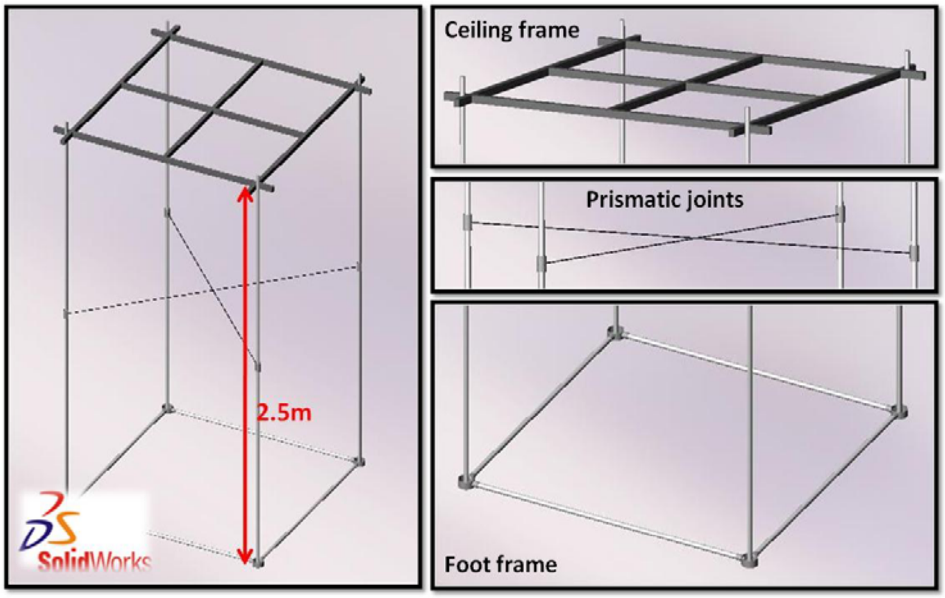


Figure 30: Test bed model in Solidworks

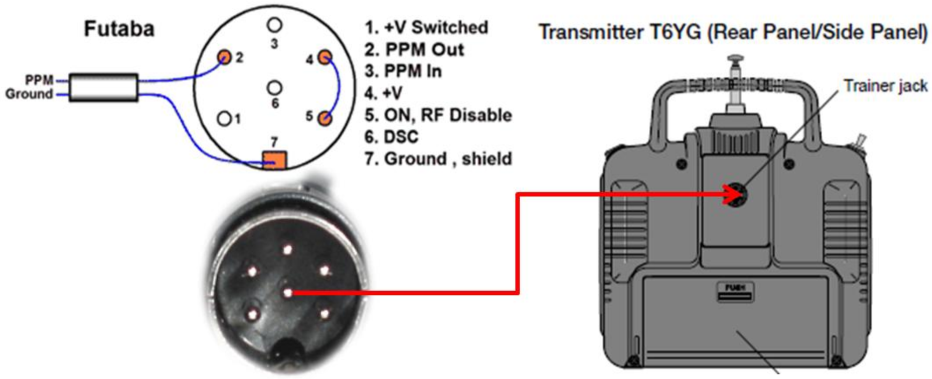


Figure 31: Pin-Out diagram of RF transmitter, Futaba Skysport6 T6YG

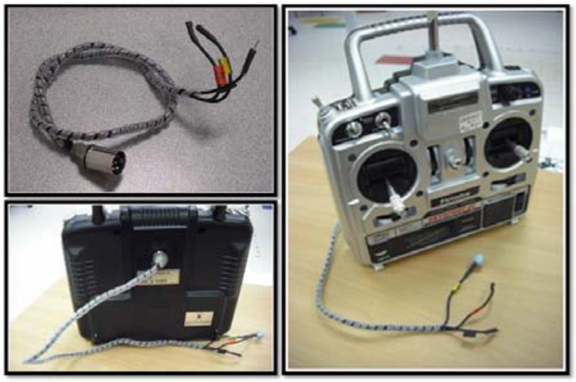


Figure 32: Modified wiring into trainer port

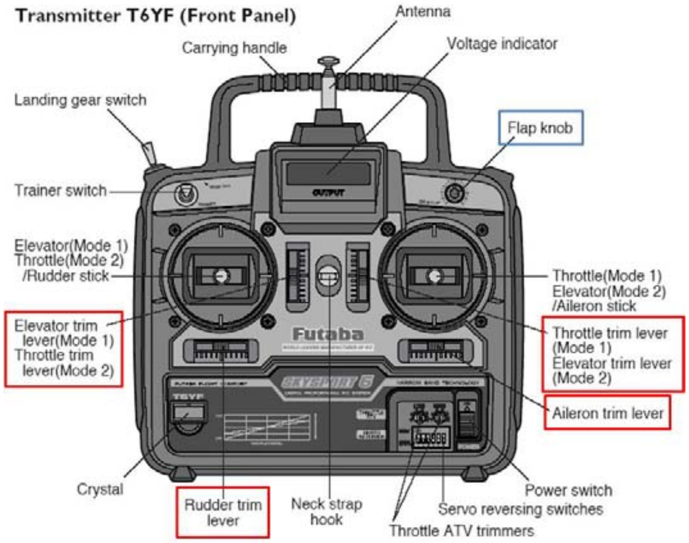


Figure 33: Futaba transmitter controls diagram

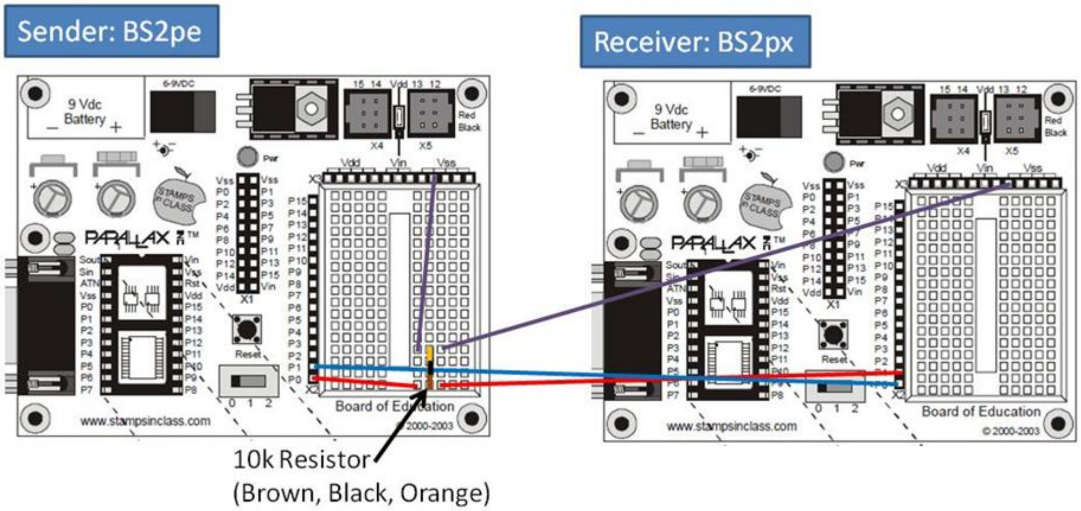
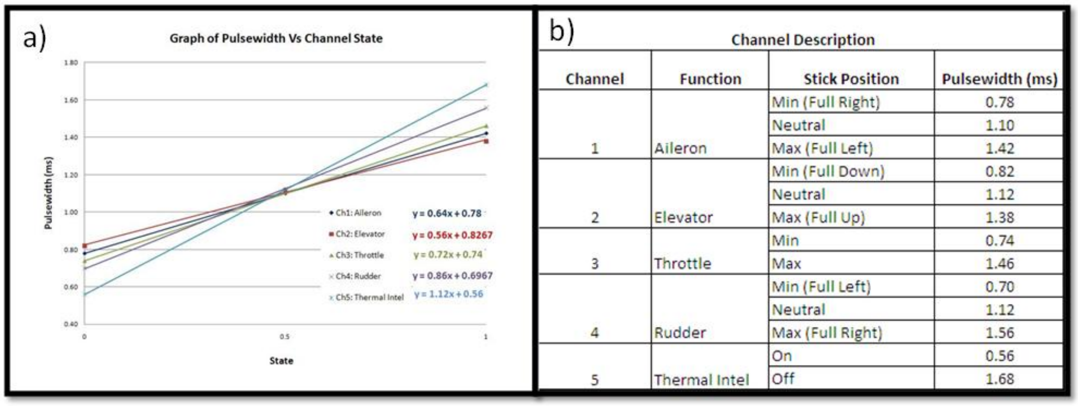


Figure 34: Serial communication with flow control



Channel Description

Channel	Function	Stick Position	Pulsewidth (ms)
1	Aileron	Min (Full Right)	0.78
		Neutral	1.10
		Max (Full Left)	1.42
2	Elevator	Min (Full Down)	0.82
		Neutral	1.12
		Max (Full Up)	1.38
3	Throttle	Min	0.74
		Max	1.46
4	Rudder	Min (Full Left)	0.70
		Neutral	1.12
		Max (Full Right)	1.56
5	Thermal Intel	On	0.56
		Off	1.68

Figure 35: Description of Pulse width Vs Channel

Appendix II. Four working states of a rotor in axial flight

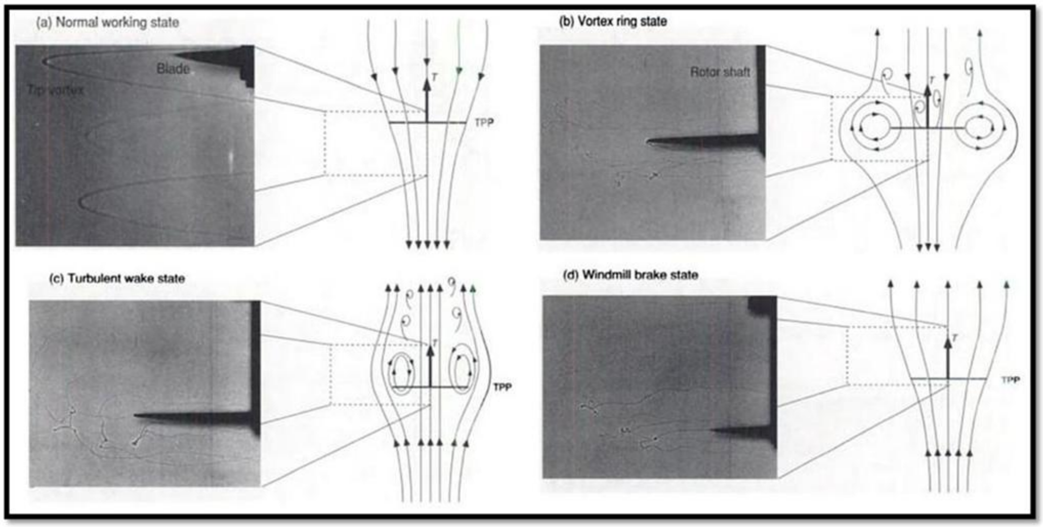


Figure 36: Flow visualization at various descent velocities using Shadowgraphy

Figure 36 shows the four working states of rotor in axial flight and the descriptions of each state are summarized below:

a) Normal Working state:

The tip vortices follow smooth helicoidal trajectories. The flow is highly periodic with a smooth slipstream boundary free of any significant disturbances. This state encompasses climb, with the limit being the hovering state.

b) Vortex Ring state:

For low rate of descent, the tip vortex filaments are convected closer to the plane of the rotor but also move radially outward away from it. At higher descent rates, the tip vortices come very close to the rotor plane and considerable unsteadiness becomes apparent. This can be seen by the contortions in the tip vortex trajectories and the lack of distinct slipstream boundary. This is close to the flow condition known as the vortex ring state (VRS), where the accumulation of tip vortices in the rotor plane resembles concentric sets of vortex ring.

c) Turbulent wake state:

As the descent rate increases further, the wake above the rotor becomes turbulent and aperiodic. This state represents the initial return to a smooth flow with a well defined slipstream boundary. The flow is similar to that associated with a buff body.

d) Windmill brake state:

At even higher descent velocities, a more definite slipstream boundary that expands downstream (above) of the rotor is observed once again. In this state, the vortical wake structure returns to a more regular helical structure. This flow condition is known as the windmill brake state because the rotor extracts energy from the flow and brake the flow velocity like a windmill.

Appendix III. Coander effect

The phenomenon which causes flow near limiting surface to follow the geometrical shape of these surfaces is known as the Coander effect. It only occurs when the flow is not forced to change its direction too abruptly in order to prevent the formation of turbulence and separation. The classic example is a flow across a flat plane with an adjacent half cylinder as shown in figure 37 below.

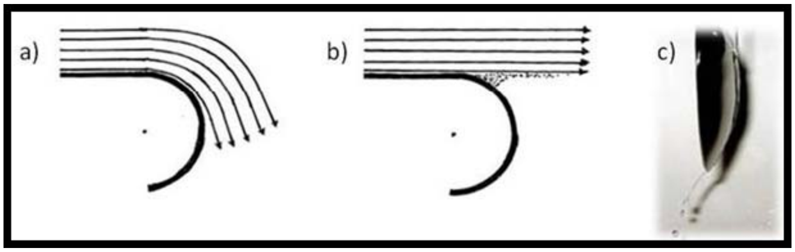


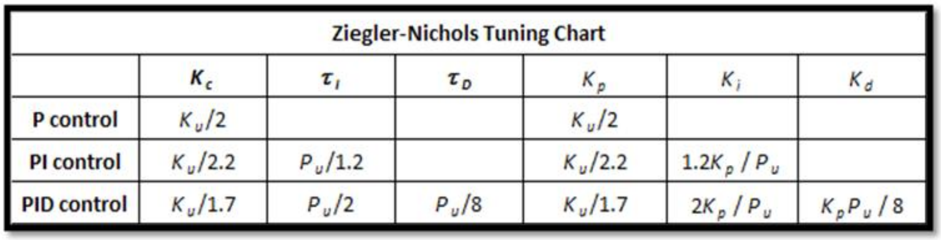
Figure 37: Flow across a limiting surface

In 37a, the flow initially follows the surface of the cylinder which eventually separates at some point on the curved surface. Another simple example is demonstrated in 5c, when water is dripped down the back of a spoon, it will tend to stick to the surface and get deflected. This important behaviour holds for all flows limited by smoothly curved surfaces like aerofoils, streamlined obstacles and sails.

The Coander effect can be explained by considering the effects of viscosity of a fluid. In 37b, due to viscosity, some layers of the adjacent air at the end of the flat surface, depicted by dots, are carried away by the main stream flow. As a result, a region of low pressure is formed which give rise to a net force which will pull the flow towards the curved surface.

Appendix IV. Ziegler Nichols tuning method

Table 1: Ziegler Nichols tuning chart



Ziegler-Nichols Tuning Chart

	K_c	τ_I	τ_D	K_p	K_i	K_d
P control	$K_u/2$			$K_u/2$		
PI control	$K_u/2.2$	$P_u/1.2$		$K_u/2.2$	$1.2K_p\,/\,P_u$	
PID control	$K_u/1.7$	$P_u/2$	$P_u/8$	$K_u/1.7$	$2K_p\,/\,P_u$	$K_pP_u\,/\,8$

where K_c is the proportional gain
τ_I is the parameter that scales the integral controller
τ_D is the parameter that scales the derivative controller
K_U is the ultimate gain
K_p , K_i *and* K_d are the proportional, integral and derivative gains

The tuning chart follows the PID equation:

$$u(t) = K_c\left[e(t) + \frac{1}{\tau_i}\int_0^t e(\tau)d\tau + \tau_d\frac{de(t)}{dt}\right] + b$$

In order to determine the parameters, the following steps are carried out:

1. Set the controller to proportional control only.

2. Step increment the controller gain K_c and wait for a steady state output before increasing by another subsequent step.

3. The critical or ultimate gain K_u, is the value of K_c that will results in a sustained periodic oscillation in the output.

4. The period of this oscillation is known as the ultimate period P_u.

5. The respective K_c, τ_I and τ_D can be determined from the ultimate gain and ultimate period according to the chart above.

Appendix V. Coander Effect flying saucer experiment

Modifications to UAV

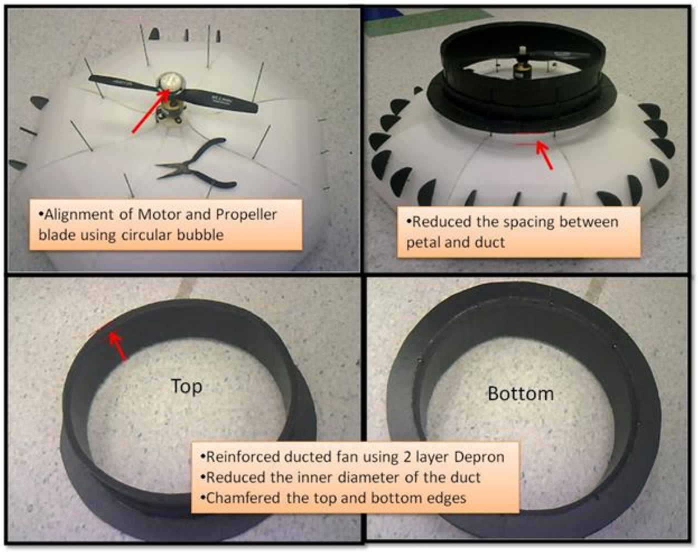

•Alignment of Motor and Propeller blade using circular bubble
•Reduced the spacing between petal and duct
Top
Bottom
•Reinforced ducted fan using 2 layer Depron
•Reduced the inner diameter of the duct
•Chamfered the top and bottom edges

Figure 38: Modification to Platform

Some modifications were carried out to improve the UAV after the initial motor test.

1. The motor as well as the propeller are realigned to ensure that it was in the horizontal plane. This is important as slight misalignment will cause the propeller to oscillate and vibrate exceedingly when the motor rotates. This was done using a circular air bubble.
2. The spacing between the petal and the duct was reduced to increase air flow.
3. The air duct was double layered. This was done to improve on its rigidity to maintain its cylindrical structure during flight. As the inner diameter of the duct was also reduced to improve on the airspeed, any slight deformation of the duct will cause the propeller to collide with the duct. Lastly, the entrance and exit of the duct was chamfered to improve airflow.

J.L. Naudin Experiment

Experimental setup

A thrust experiment designed by Jean Louis Naudin was published on his website.

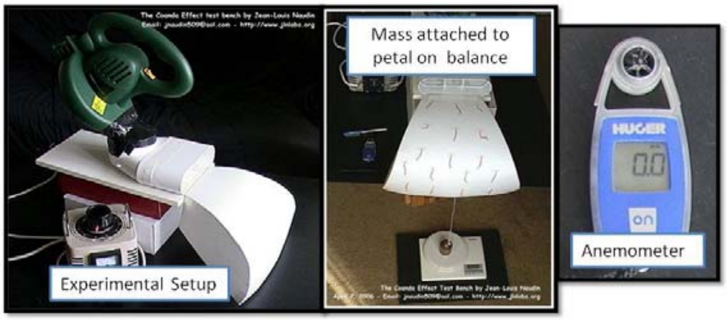


Figure 39: J. Naudin's experimental setup

In his experiment, he used a blower to blow across one petal to simulate air that moves through the air duct over the surfaces of the petals. A weight was attached to the other edge of the petal and was placed onto a weighing scale. An anemometer was used to measure the airspeed at the different locations of the petal.

Results

The result that was obtained from his experiment is plotted in the graph below:

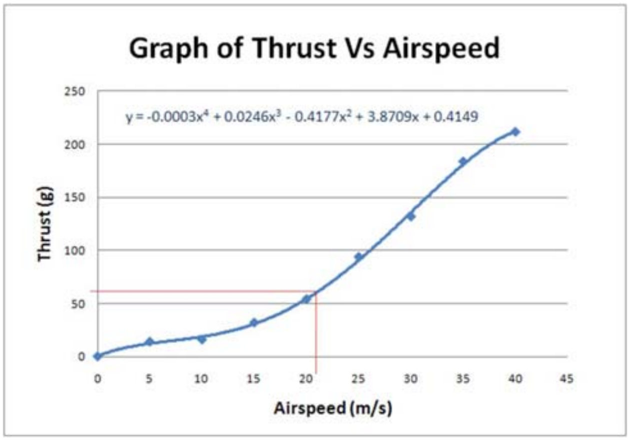
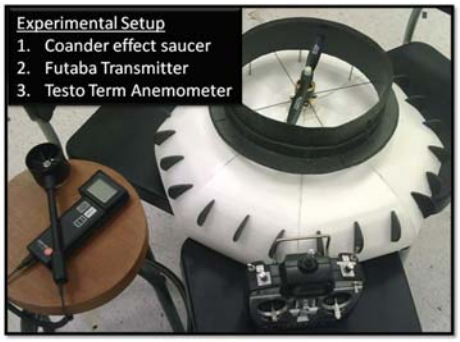


Figure 40: Naudin's result

From this plot, it can be deduced that a minimum airspeed of 21m/s is required at the outlet of the ducted fan in order to generate 533 g of thrust, which was the overall weight of his platform. In other words, the take off thrust required is 533g.

Experiment conducted: Air speed measurement

Experimental setup and Results

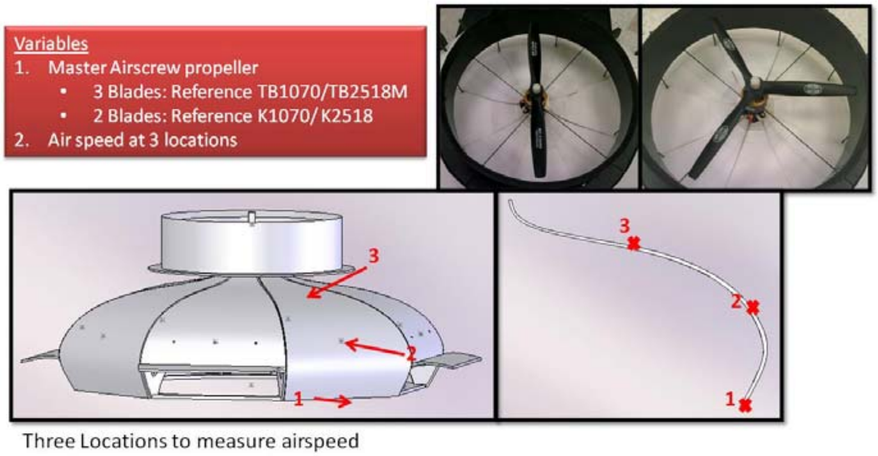


Experimental Setup

1. Coander effect saucer
2. Futaba Transmitter
3. Testo Term Anemometer

Figure 41: Experimental setup for airspeed measurement

A simple experiment was conducted to verify the airspeed at the different positions of the petal, in order to deduce the thrust that would be generated according to Naudin's result.



Variables

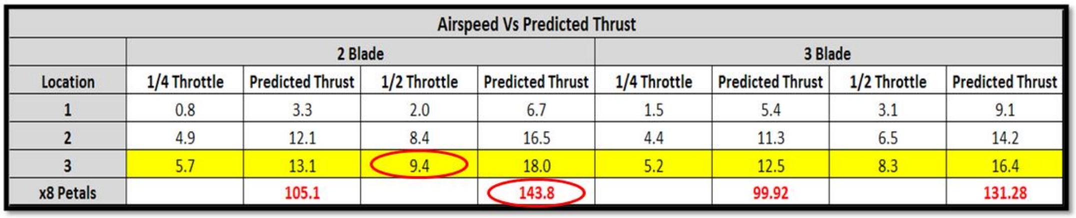
1. Master Airscrew propeller
   - 3 Blades: Reference TB1070/TB2518M
   - 2 Blades: Reference K1070/ K2518
2. Air speed at 3 locations

Three Locations to measure airspeed

Figure 42: Variables of experiment

The airspeeds at the three locations, using 2 different types of propeller, are measured and the result is tabulated below:

Table 2: Airspeed test results



Airspeed Vs Predicted Thrust								
	2 Blade				3 Blade			
Location	1/4 Throttle	Predicted Thrust	1/2 Throttle	Predicted Thrust	1/4 Throttle	Predicted Thrust	1/2 Throttle	Predicted Thrust
1	0.8	3.3	2.0	6.7	1.5	5.4	3.1	9.1
2	4.9	12.1	8.4	16.5	4.4	11.3	6.5	14.2
3	5.7	13.1	9.4	18.0	5.2	12.5	8.3	16.4
x8 Petals		105.1		143.8		99.92		131.28

Predicted thrust is calculated using equation that was derived in his experiment:

$$Thrust = -0.0003Airspeed^4 + 0.0246Airspeed^3 - 0.4117Airspeed^2 + 3.8709Airspeed + 0.414$$

Conclusion

It was observed that the 2 blade propeller was able to produce more thrust than the 3 blade propeller. The maximum thrust that could be produced by all the eight petals was evaluated to be only 143.8g. This was barely sufficient to lift off the platform which has an overall weight of 570g, without including other components such as servos and sensors. Thus, the usage of this platform for this project was ruled out.

Appendix VI. Autonomous landing program (Basic)

Processor

Scheduled proportional control

```
'~~~~~~~~~SENDER/PROCESSOR~~~~~~~~~~~~~~~~~
' {$STAMP BS2pe}
' {$PBASIC 2.5}
' {$PORT COM8}
'~~~~~~~~~~~~~~~~~~~Set Pin~~~~~~~~~~~~~~~~~~~
SO    PIN     1                'Serial Out Pin
FC    PIN     0                'Flow control Pin
'~~~~~~~~~~~~Baud rate Select~~~~~~~~~~~~~~~~~
#SELECT $STAMP
#CASE BS2PE
T4800 CON 16572
T9600 CON 16468
#CASE BS2PX
T4800 CON 17197
T9600 CON 16780
#ENDSELECT
Baud CON T9600
'~~~~~~~~~~~~~~Throttle Config~~~~~~~~~~~~~~~~
'tMin           CON     1600
'tMax            CON     1800
tLand          CON     1400
tHover         CON     1630
throttle        VAR     Word
throtsend     VAR     Byte
'~~~~~~~~~~~~~~~PID Variables~~~~~~~~~~~~~~~~~
setpoint       CON     5
error            VAR     Word
p                  VAR     Word          'Proportional term
'~~~~~~~~~~~~~~~~~~~PING))) Variables~~~~~~~~~~~~~~~
cmConstant     CON     2260
cmDistance      VAR     Word
time                 VAR     Word

PAUSE 500                                   'For Plotting
DEBUG "!SPAN 0,200",CR              'Altitude axis
DEBUG "!TMAX 180",CR                 'Time axis
```

```
DEBUG "!PNTS 4000",CR                  'Data points per plot
DEBUG "!TITL Altitute Log",CR          'Title the form
DEBUG "!SHFT ON",CR                    'Plot shift at max
DEBUG "!TSMP OFF",CR
DEBUG "!SAVD ON",CR
DEBUG "!SAVM ON",CR
DEBUG "!NAMD PlotData.txt",CR
DEBUG "!NAMM PlotMsg.txt",CR
DEBUG "!PLOT ON",CR                     'Enable plotting
DEBUG "!RSET",CR                       'Reset Plot

DO
  PULSOUT 15, 5                               'Tout from PING)))
  PULSIN 15, 1, time                          'Tin to PING))) ->Time variable
  cmDistance = cmConstant ** time             'Distance from PING)))

  DEBUG DEC cmDistance, CR                    'For Plotting

  error = setPoint - cmDistance

  IF cmDistance > 45 THEN                      'Augment Proportional Gains
   p = 2*error
   ELSEIF cmDistance < 45 THEN
   p = error
  ENDIF

  throttle = p + tHover

  IF cmDistance < 20 THEN
    throttle = tLand
  ENDIF

  throtsend = throttle/10
  SEROUT SO\FC, Baud, [throtsend]
LOOP
```

Proportional-Derivative control

```
'~~~~~~~~~~~~~SENDER/PROCESSOR~~~~~~~~~~~
' {$STAMP BS2pe}
' {$PBASIC 2.5}
' {$PORT COM8}
'~~~~~~~~~~~~~~~~~~~~Set Pin~~~~~~~~~~~~~~~~~
SO    PIN     1    'Serial Out Pin
FC    PIN     0                'Flow control Pin
'~~~~~~~~~~~~Baud rate Select~~~~~~~~~~~~~~~~~~
#SELECT $STAMP
#CASE BS2PE
T4800 CON 16572
T9600 CON 16468
#CASE BS2PX
T4800 CON 17197
T9600 CON 16780
#ENDSELECT
Baud CON T9600
'~~~~~~~~~~~~~~~Throttle Config~~~~~~~~~~~~~~~~~
tLand          CON    1400
tHover         CON    1730
throttle        VAR    Word
throtsend      VAR    Byte

'~~~~~~~~~~~~~~~~PID Variables~~~~~~~~~~~~~~~~~~
setpoint       CON     5                       'Setpoint
current       CON     0             'Current error
previous      CON     1                       'Previous error
delta          CON     2             'Change in error
error          VAR    Word(3)              'Error array
p              VAR    Word       'Proportional term
d              VAR    Word       'Derivative term
'~~~~~~~~~~~PING)))Variables~~~~~~~~~~~~~~~~~~~~~
cmConstant      CON    2260
cmDistance      VAR    Word
time              VAR    Word

PAUSE 500
DEBUG "!SPAN 0,200",CR                  'Set for altitude axis
DEBUG "!TMAX 180",CR                    'Set for time axis
```

```
DEBUG "!PNTS 4000",CR                 '4000 data points per plot
DEBUG "!TITL Altitute Log",CR         'Title the form
DEBUG "!SHFT ON",CR                   'Allow plot to shift at max
DEBUG "!TSMP OFF",CR
DEBUG "!SAVD ON",CR
DEBUG "!SAVM ON",CR
DEBUG "!NAMD PlotData.txt",CR
DEBUG "!NAMM PlotMsg.txt",CR
DEBUG "!PLOT ON",CR                   'Enable plotting
DEBUG "!RSET",CR                      'Reset Plot

DO
  PULSOUT 15, 5                         'Tout from PING)))
  PULSIN 15, 1, time                    'Tin to PING))) ->Time variable
  cmDistance = cmConstant ** time     'Distance from PING)))

  DEBUG DEC cmDistance, CR

  error(current) = setPoint - cmDistance              ' Calculate error
  p = 3*error(current)                                'Proportional Gain=3
  error(delta) = error(current) - error(previous)
  d = error(delta)                                    'Derivative Gain=1
  throttle = p + tHover + d     MIN 1500 MAX  1800

  IF cmDistance < 20 THEN
    throttle = 1420
  ENDIF

  throtsend = throttle/10
  SEROUT SO\FC, Baud, [throtsend]

  error(Previous) = error(Current)
LOOP
```

Signal generator

```
'~~~~~~~~~~RECEIVER/ SIGNAL GENERATION~~~~~~~
' {$STAMP BS2px}
' {$PBASIC 2.5}
' {$PORT COM5}
'~~~~~~~~~~~~~~~~~~~~Set Pin~~~~~~~~~~~~~~~~~~~~
SI           PIN     0      'Serial In Pin
FC            PIN     1      'Flow control Pin
PP            PIN     5      'Pause Pin
TP            PIN     13    'Trainer Pin
'~~~~~~~~~~~Stamp-Baud rate Select~~~~~~~~~~~~
#SELECT $STAMP
#CASE BS2PE
T4800 CON 16572
T9600 CON 16468
#CASE BS2PX
T4800 CON 17197
T9600 CON 16780
#ENDSELECT
Baud CON T9600
'~~~~~~~~~~~~~~~CALIBRATION DATA~~~~~~~~~~~~~~
ch1          CON     1115       'Aileron Stick: (1775)Left-Mid(1375)-Right(975)
ch2          CON     1380       'Elevator: (1725)Up-Mid(1400)-Down(1025)
ch4          CON     1400       'Rudder
ch5          CON     2100       'Thermal
ch6          CON     1400
ch7          CON     1400
sync          CON     10305     'Synchro pulse
delay         CON     375        'Pause pulse
lastdelay    CON     260
'~~~~~~~~~Variables from Serial INPUT~~~~~~~~~~~~~~
throttle         VAR Byte
prevthrottle   VAR Word
throtgen        VAR Word

DO
  SERIN SI\FC, Baud, 2, timeout, [throttle]       'timeout: 2 units = 0.8 ms
  throtgen = throttle*10
  PULSOUT TP, ch1
  PULSOUT PP, delay
  PULSOUT TP, ch2
```

```
  PULSOUT PP, delay
  PULSOUT TP, throtgen
  PULSOUT PP, delay
  PULSOUT TP, ch4
  PULSOUT PP, delay
  PULSOUT TP, ch5
  PULSOUT PP, delay
  PULSOUT TP, ch6
  PULSOUT PP, delay
  PULSOUT TP, ch7
  PULSOUT PP, delay
  PULSOUT TP, sync

  prevthrottle = throtgen    'Stores last signal
LOOP

timeout:                     'Generate last signal
    throtgen = prevthrottle
    PULSOUT TP, ch1
    PULSOUT PP, delay
    PULSOUT TP, ch2
    PULSOUT PP, delay
    PULSOUT TP, throtgen
    PULSOUT PP, delay
    PULSOUT TP, ch4
    PULSOUT PP, delay
    PULSOUT TP, ch5
    PULSOUT PP, delay
    PULSOUT TP, ch6
    PULSOUT PP, delay
    PULSOUT TP, ch7
    PULSOUT PP, delay
    PULSOUT TP, sync
RETURN
```