



**University of Minnesota**

**Department of Aerospace Engineering  
& Mechanics**

**UAV Research Group**

Paw Yew Chai

March 23, 2009



## Contents

<b>1</b>	<b>Background</b>	<b>3</b>
1.1	Research Area . . . . .	3
1.2	People involved . . . . .	3
1.3	Open source philosophy . . . . .	3
<b>2</b>	<b>UAV Testbeds</b>	<b>4</b>
2.1	Yardstik UAV . . . . .	4
2.1.1	Airframe . . . . .	4
2.1.2	Instrumentation . . . . .	4
2.2	Ultrastick UAV . . . . .	6
2.2.1	Instrumentation . . . . .	6
<b>3</b>	<b>Flight Computer system</b>	<b>8</b>
3.1	Flight Computer . . . . .	8
3.2	IMU/GPS Sensor . . . . .	9
3.3	Software . . . . .	11
3.3.1	Operating System . . . . .	11
3.3.2	Program module . . . . .	11
<b>4</b>	<b>Ground Control Station</b>	<b>13</b>
4.1	GCS Hardware System . . . . .	13
4.2	GCS Software Program . . . . .	13
<b>5</b>	<b>Simulation Testing Facilities</b>	<b>15</b>
5.1	Software-in-Loop Simulation . . . . .	17
5.2	Processor-in-Loop Simulation . . . . .	18
5.2.1	PIL Simulation System Setup . . . . .	18
5.2.2	PIL Software System Architecture . . . . .	18
5.2.3	PIL Hardware System Architecture . . . . .	19
5.3	Simulator Flight Testing . . . . .	22
<b>6</b>	<b>Wind Tunnel Testing Facility</b>	<b>24</b>
<b>7</b>	<b>Flight Testing</b>	<b>26</b>
7.1	Indoor Testing . . . . .	28
7.2	Outdoor Testing . . . . .	28



## 1 Background

The UAV group research focuses on development, implementation and validation of new control, navigation and guidance strategies for small autonomous UAV. The focus of research is not only to address issues from technological standpoint, but also ensure the system is built mostly out of commercial of the shelf components to maximize cost benefits, share the knowhow of design and development of hardware and software with researchers across the world with open source philosophy.

### 1.1 Research Area

- Nonlinear robustness analysis of flight control laws
- Adaptive and nonlinear flight control algorithms development
- Modeling, analysis and validation
- Fault detection and isolation
- Attitude determination
- Sensor fusion algorithms

### 1.2 People involved

The people involved in the UAV group are the professors and graduate students in the system group within the AEM department. Undergraduate from different disciplines are also actively involved in the activities in the research group. Beside the researchers from AEM department, we are in collaborating with universities in Europe, which are Budapest University of Technology and Economics (Hungary) and University of Sannio at Benevento (Italy).

### 1.3 Open source philosophy

An open source philosophy is adopted for the research work done so that we can share all our development efforts (software codes and system integration architecture) with researcher around the world who are interested in using these results. All these information are available online on the UAV website (<http://www.aem.umn.edu/~uav/>)



## 2 UAV Testbeds

Different aircrafts are used for different research projects. Most of the aircraft used are Commercially-off-the-shelve(COTS) RC planes and they are modified to carry required instrumentations and payloads for various research purposes. Customized aircrafts are designed and build when the COTS RC planes do not fulfill the required performance. 2 of the UAV testbeds used are:

### 2.1 Yardstik UAV

The Yardstik UAV testbed (Figure 1(a)) is used for indoor UAV research that demand challenging autonomous flight in enclosed indoor environment.

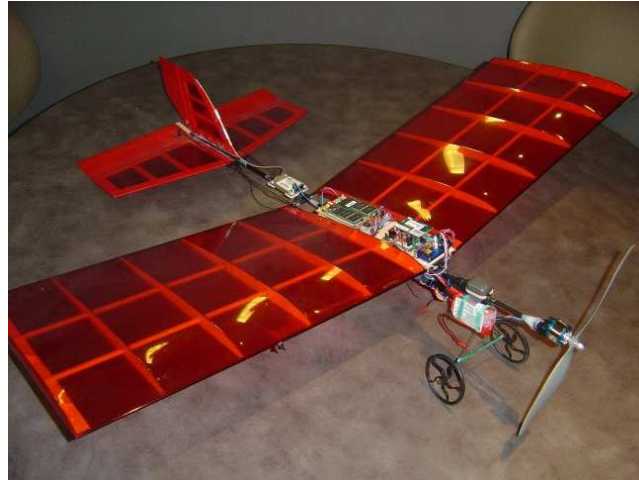
#### 2.1.1 Airframe

The Yardstik is a commercially available remote control plane, Yardstik, manufactured by Great Planes. The Yardstik is an electric powered park flyer plane which is suitable for a beginner to fly. The Yardstik is build up of wooded tail and wing sections. The wings are reinforced by carbon fibre rods in leading and trailing edges with a transparent film skin covering. The fuselage is made of a simple carbon rod and the wings are joined together with a dihedral angle of 22 degrees. The control of the airplane is done using 2 control surfaces, the elevator and rudder control surfaces located at the horizontal and vertical tail. The control surfaces are actuated using the Futaba S3108 micro servos. The specifications of the aircraft are as follows:

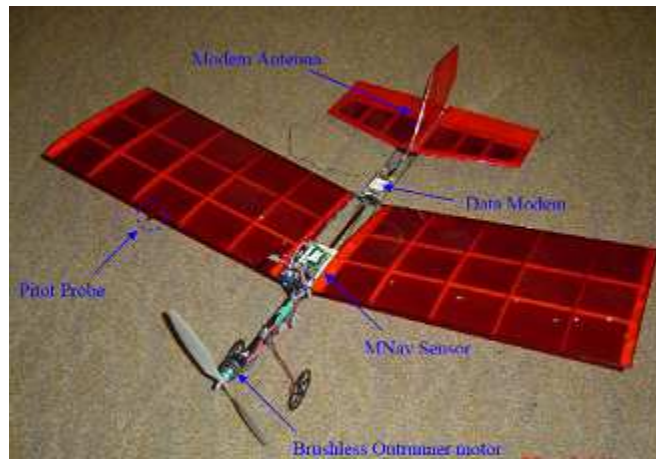
- Length: 0.91 *m*
- Wing Span: 1.08 *m*
- Empty weight: 350 *g*
- MTOW (Tested): 550 *g*
- Endurance: 5 to 10 min
- Cruise speed: 7 to 11 *m/s*

#### 2.1.2 Instrumentation

Figure 1(b) shows the avionics hardware integrated onto YardStick R/C airframe . Note that physical separation of the components has been done to minimize Radio Frequency Interference (RFI) . A pitot probe is placed on the right hand side of the wing with a distance of more than a diameter of propeller length from the center line of the fuselage so that the airspeed measured is not affected by propeller wash effect.



(a) Yardstik UAV testbed



(b) Yardstik UAV instrumentation

Figure 1: Yardstik UAV



## 2.2 Ultrastick UAV

The *Ultra Stick 25E* R/C (shown in Figure 2(a)) is a COTS RC plane manufactured by Hanger9. The plane has a conventional horizontal and vertical tail with rudder and elevator control surfaces respectively. The aircraft has a symmetrical airfoil wing, which has both aileron and flap control surfaces. All the control surfaces are actuated by *Hitec* servos. The plane is propelled by a 600 watts electric *E-Flite* outrunner motor with an *APC* 12 x 6 propeller. The specifications of Ultrastick are as follows:

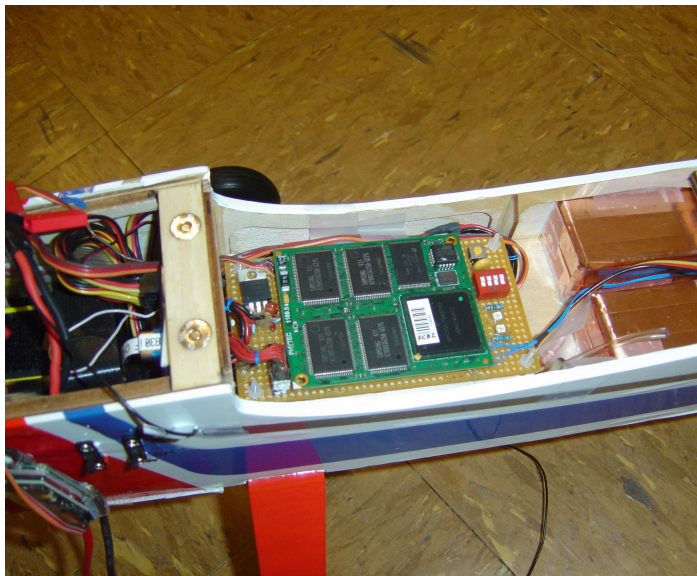
- Length: 1.05 *m*
- Wing Span: 1.27 *m*
- Empty weight: 1.50 *kg*
- MTOW (Tested): 2.04 *kg*
- Endurance: 15 to 20 min
- Cruise speed: 15 to 20 *m/s*

### 2.2.1 Instrumentation

Figure 2(b) shows avionics hardware components integrated onto Ultrastick R/C airframe. This is the same set of instrumentation that is used in the Yardstick UAV.



(a) Ultrastick UAV testbed



(b) Ultrastick UAV instrumentation

Figure 2: Ultrastick UAV



### 3 Flight Computer system

The current architecture of the flight computer system is shown in Figure 3 and Table 1 gives a listing of the individual component in the system. The flight computer runs on *eCos* (embedded configurable operating system) real-time operating system and takes in sensor data and performs attitude determination using a 7 states Kalman Filter. At the same time, the flight computer also executes flight control algorithms and outputs Pulse-Width Modulated (PWM) signals to control the servo actuators and sends data information to serial communication port 2 for the data modem and datalogger for flight test data collection purpose. Dual channels datalogger is used to log both raw sensor data (50Hz) and flight control data (20 Hz) so that consistent and reliable data can be recorded. The flight control data is also send through data modem to ground control station for real-time health monitoring purpose during flight testing. A failsafe switching board is used as safety precaution for overriding flight computer command to manual pilot mode if necessary.

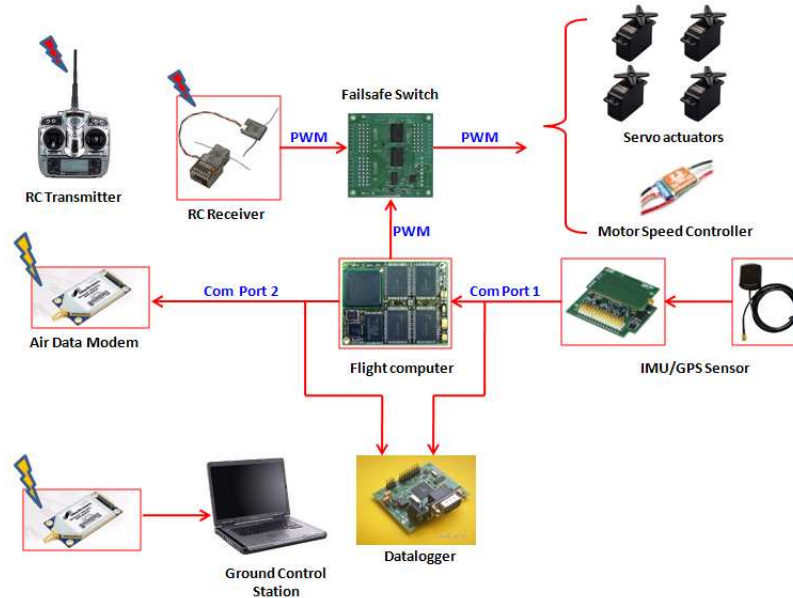


Figure 3: Flight Computer System architecture

#### 3.1 Flight Computer

The current flight computer used is phyCore MPC555 32-bit PowerPC micro-controller. The MPC555 has a clock frequency of 40 Mhz and performs floating point computation. It has 2 serial ports and CAN bus each for communication





Table 1: Flight computer system components

Component	Module
Flight computer	Phytec MPC 555 microcontroller
IMU/GPS sensor	Crossbow Micronav sensor
Data Modem	Maxstream 900 Mhz modem
R/C telemetry	Spektrum DX-7 2.4 Ghz RC system
Failsafe switch	RxMux board
Datalogger	Antilog RS232 serial datalogger

with external devices. The physical dimension of the board is  $72 \times 57$  mm and weighs 25 g. Details on the specifications of the MPC 555 can be found in <http://www.phytec.com/products/sbc/PowerPC/phyCORE-MPC555.html>.

### 3.2 IMU/GPS Sensor

The IMU/GPS sensor (Micronav) from Crossbow is a low cost inertia sensor system integrated with GPS receiver unit. The specifications of the MicroNav is shown in Figure 4. This sensor provides the following measurement data:

- Angular rates:  $p$ ,  $q$  and  $r$
- Accelerations:  $a_x$ ,  $a_y$  and  $a_z$
- Magnetic fields:  $H_x$ ,  $H_y$  and  $H_z$ ,
- Airspeed and barometric altitude:  $V_s$  and  $h$ ,
- GPS velocities (ENU format) and positions:  $v_e$ ,  $v_n$ ,  $v_u$  and  $p_x$ ,  $p_y$ ,  $p_z$ .



Specifications	MNAV100CA	Remarks
<b>Performance</b>		
Update Rate (Hz)	2-100	User Programmable
Angular Rate Range	± 150	
Acceleration Range X/Y/Z (g)	± 2	
Inertial Sensor Bandwidth (Hz)	> 25	-3 dB point
Magnetometer Range (G)	± 0.75	
Altitude Range (m,MSL)	0-5000	
Airspeed Range (m/s)	0-80	
GPS Accuracy (m)	3	CEP
<b>Environment</b>		
Operating Temperature (°C)	-5 to +45	
<b>Electrical</b>		
Input Voltage (VDC)	3.7 to 16	
Power Consumption (W)	< 0.8	at 5 VDC
Digital Output Format	RS-232	
<b>Physical</b>		
Size (in)	2.25 x 1.80 x 0.44	
(cm)	5.70 x 4.50 x 1.10	
Weight (g)	33	
Connector	15X3 Array of 0.1 inch square pins	

Figure 4: Specifications of Micronav sensor



### 3.3 Software

#### 3.3.1 Operating System

The operating system used onboard of the MPC555 flight computer is eCos realtime operating system. The reasons of using eCos are as follows:

- Open source, freely available
- Highly configurable runtime environment
- Real-time kernel
- POSIX, ITRON 3.0 compatible API
- ISO C support
- Supports multi processor architectures!

#### 3.3.2 Program module

Different program modules are written for the flight computer system to perform required tasks. These modules are written in C-codes in which they are compile so that binary executables are built and uploaded to the MPC555. Since each module has its own task, multiple tasking has to be performed with the limited computation resources. The multitasking is implemented through multi-threading such that each process has different priority level and they are run at different rates. Currently, there are 5 main threads that are running onboard of the flight computer, which are:

1. IMU DAQ: This thread does data acquisition of sensor data from the sensor into the flight computer module through serial port.
2. AHRS: This thread does attitude determination using the sensor data using a 7 states Extended-Kalman Filtering (EFK).
3. CLAW: This thread does flight control commands computation using the flight control algorithms implemented.
4. INS-GPS: This thread does INS/GPS navigation filtering algorithms.
5. Telemetry: This thread does packing and sending of required telemetry data to the ground control station.

Table 2 provides the priority of the threads and the frequency these processes are running.



Table 2: Thread priority

Priority	Thread	Frequency (Hz)
1	IMU DAQ	50
2	AHRS	50
3	CLAW	25
4	INS-GPS	10
5	Telemetry	20



## 4 Ground Control Station

The ground control station is used in the flight testing to monitor the flight progress (waypoint segments) and health status of UAV during the flight test. At the same time, it provides telemetry datalogging for the flight testing for post flight analysis.

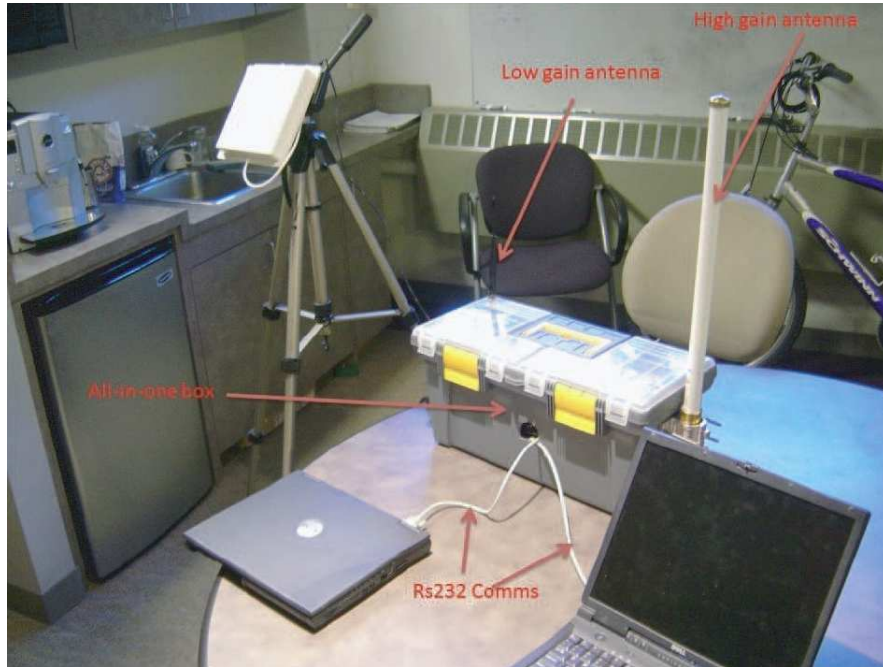
### 4.1 GCS Hardware System

The ground control station consist of a laptop computer that is hooked up to a ground data modem through a RS232 serial port. The data modem has a high gain omnidirectional antenna. All the system on the ground are powered by a heavy duty 12 volts lead acid battery. Figure 5(a) shows the hardware setup of the ground control station.

### 4.2 GCS Software Program

The GCS software program that runs on the laptop computer is based on Open Source Glass Cockpit Project or OpenGC. The website for the project is [www.opengc.org](http://www.opengc.org). It is designed to give vital flight information in real time to observers in order to assess the flight performance of an aircraft during a flight test. The software also logs the flight data to text files for further analysis. The actual code is written in C++ with the OpenGL Graphics Library used to render the flight displays.

In addition to the Primary Flight Display (modified from the original to display low speed and low altitude flight) that came with the original version of OpenGC, the modification is done to the OpenGC so that it allows receiving of data from serial communications, a moving map to display location relative to a set of waypoints, servo rate gauges to display the activity of up to seven control parameters, and buttons to display autopilot mode information. The reasons for using OpenGC as a base for the ground control station are to provide an open source, customizable, and effective means for in flight visualization for the UAV. Figure 5(b) shows a screenshot of the GCS software running.



(a) GCS hardware



(b) Screenshot of GCS software

Figure 5: Ground control station



## 5 Simulation Testing Facilities

The proof of success UAV system design and development is in the demonstration of the UAV achieving the required performance specifications. Flight trial validation represents the true valid assessment whether the flight control design meets the design requirements in true environment and verifies that the design requirements used are practical and valid. However, flight trials are resource intensive and expensive and with modern flight control system getting more complex, there is a need to use other validation approaches to support and augment the validation process.

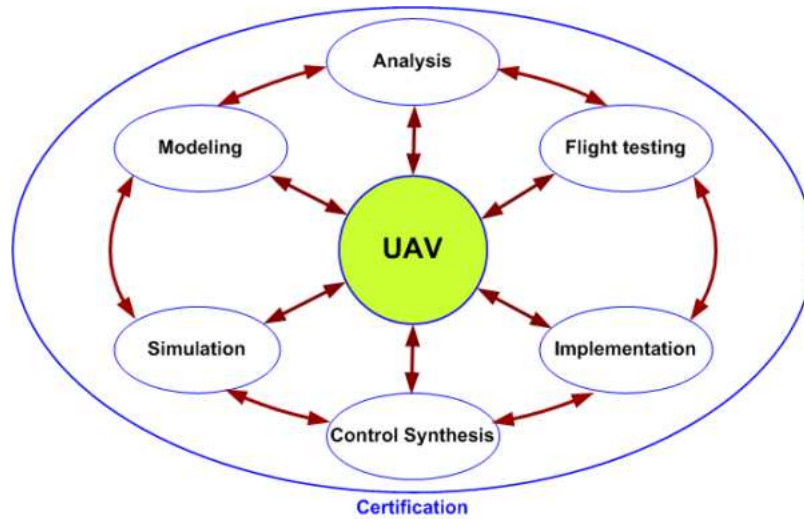


Figure 6: Integrated modeling, simulation, analysis and validation framework

The integrated framework approach (Figure 6) for flight control synthesis and validation wraps around different design and analysis tools on the nonlinear and linear simulation model that provide an iterative UAV software algorithms design and validation environment. At the same time, the linear and nonlinear models are being updated. The parallel effort in redesigning, validation and updating of models facilitate rapid system design, analysis and implementation process with the latest updated model. With the ability to update and accurately model the aircraft's aerodynamics and its individual system in the nonlinear simulation model, this provides an approach to augment the flight validation process through the use of simulation-based testing. The use of simulation-based testing helps to save time and effort prior to the actual flight trials as it can help to validate the working of flight control law, software and hardware implementation within the bench test environment. This help to ensure the UAV system is of high integrity and free of bugs prior to actual flight



trials. An example of simulation testing used is in the integrated framework approach in flight control development (Figure 7).

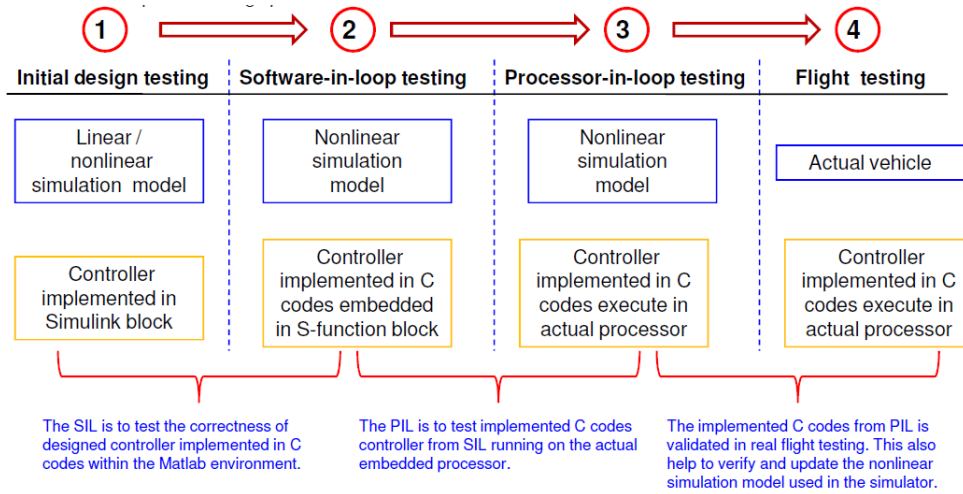


Figure 7: Application of simulation testing for flight control development





### 5.1 Software-in-Loop Simulation

The Software-in-the-Loop (SIL) setup is almost similar to Processor-in-the-Loop (PIL) setup except that all the simulation components reside in the simulation environment, including the flight control C-code, which is implemented in S-function block. Figure 8 shows the difference between the SIL and PIL simulation. The details of the setup will be discussed in the PIL simulation section.

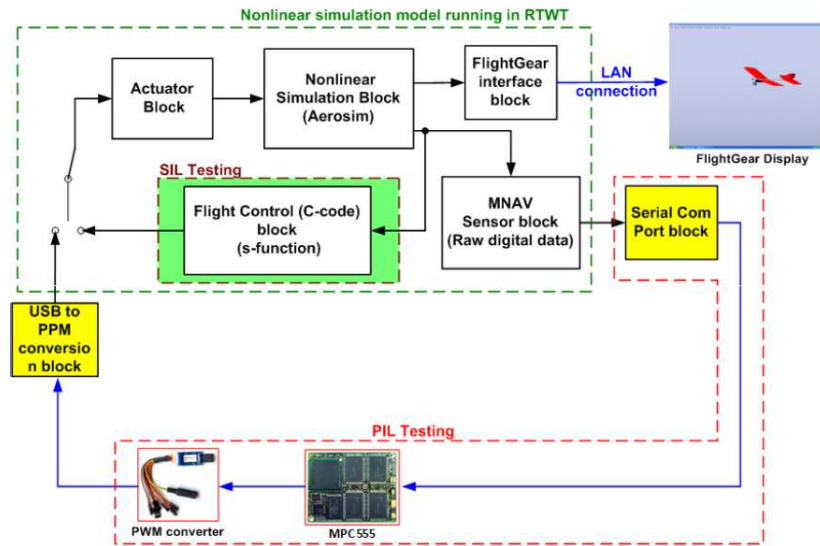


Figure 8: Block structure of SIL and PIL simulation



## 5.2 Processor-in-Loop Simulation

The PIL simulation provides an intermediate step to implement the synthesized controller on actual target hardware processor before the full-scale flight testing of the UAV with the synthesized flight control system. This approach offers the following advantages:

- Ability to test and identify controller implementation issues before the flight testing. This helps to determine controller implementation limitations on actual hardware and provides important information for controller redesign.
- Provides a real-time testing environment for synthesized controllers.
- Provides a good test-bed for integration of hardware and software system and test the functionality of the component at system level. This helps to ensure the integrated system is of high integrity and free of fault.
- Provides environment for pilot and flight test engineers to prepare, train and understand the scope of flight trial and gain confidence of the overall system.

Beside testing, debugging and validating the control design and implementation, the PIL simulation is used for post-flight analysis in validating the simulation model that is recursively updated from flight test data. Once the simulation model has been sufficiently well validated, it can be used to augment and substitute many of the flight testing and this helps to cut the risk and development cost of the system.

### 5.2.1 PIL Simulation System Setup

The PIL simulation is an extension of SIL simulation that includes actual embedded target processor (flight computer) into the simulation setup. In general, the simulation simulates the aircraft model flying in an environment defined by the user. During the simulation, the simulation environment will export sensor output data to the target processor through a communication link which is running the actual embedded software code in real-time. The flight computer will use the sensor data to generate output control signal back to the simulation model using another communication link to control the aircraft model in the simulation model to achieve the required flight trajectory. Hence the software simulation and flight computer formed a closed-loop control system (as shown in Figure 9), just like the actual UAV system flying.

### 5.2.2 PIL Software System Architecture

To make the simulation model runs real-time on desktop computer, Real-Time Windows Target (RTWT) toolbox is used. The RTWT software allows the real-time execution of the generated C code on Windows operating system and interacts with actual hardware system using the I/O devices from the desktop

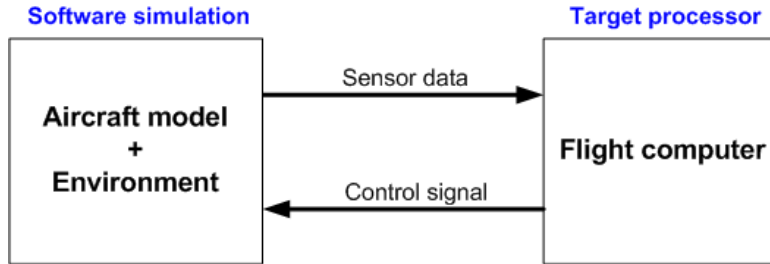


Figure 9: Overview of PIL simulation concept

computer. The entire code generation and real-time executable binary file are automated using Simulink toolboxes.

I/O blocks from RTWT toolbox are added to the nonlinear simulation model. These I/O blocks enable the nonlinear simulation model to interface with the data input and output to the flight computer and simulator. These blocks provides the I/O device drivers for the Real-Time Workshop during the auto-generation of C code. This integrated environment provides easy implementation without any hand coding or debugging. Figure 10 shows the Simulink blocks layout used for the auto-code generation in the PIL simulation. The stream data blocks are the I/O blocks from the RTWT toolbox that provide specific communication protocols (UDP (User Datagram Protocol), joystick and serial) to communicate with the peripherals to the desktop computer. In the auto-code generation for creating the executable Windows target, discrete sampling time has to be used and this is set to 0.02 seconds. This is the maximum time step size that can be used since the PIL simulation has to output sensor data at 50 Hz.

### 5.2.3 PIL Hardware System Architecture

The PIL hardware system setup is a duplicate of the actual flight computer system on the UAV except for the IMU/GPS sensor and data modem. Figure 11(a) shows the hardware architecture of the setup. The desktop computer is running both the RTWT simulation executable and flight simulator programs in real-time. Flight simulator uses UDP communication within the computer to receive data at 0.2 Hz from the RTWT simulation. Serial communication port 1 provides serial sensor data output at 50 Hz with baud rate of 38.4 Kbps to the MPC 555 processor while the USB port provides the servo commands input via joystick communication protocol to the simulator.

The failsafe board serves a multiplexer to switch between the manual pilot control and autopilot modes using the RC transmitter and this allows the PIL setup to duplicate the same scenario from a manual pilot takeoff to switching to autopilot mode during the flight at any point of time. The ground control station provides monitoring of flight data from the flight computer in real-time

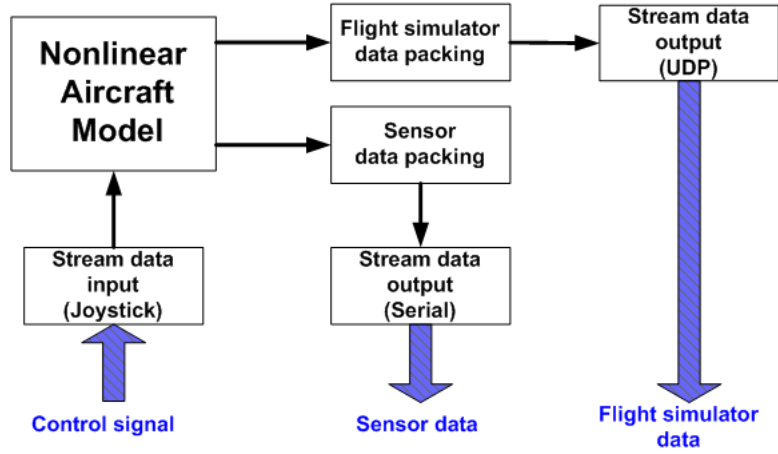
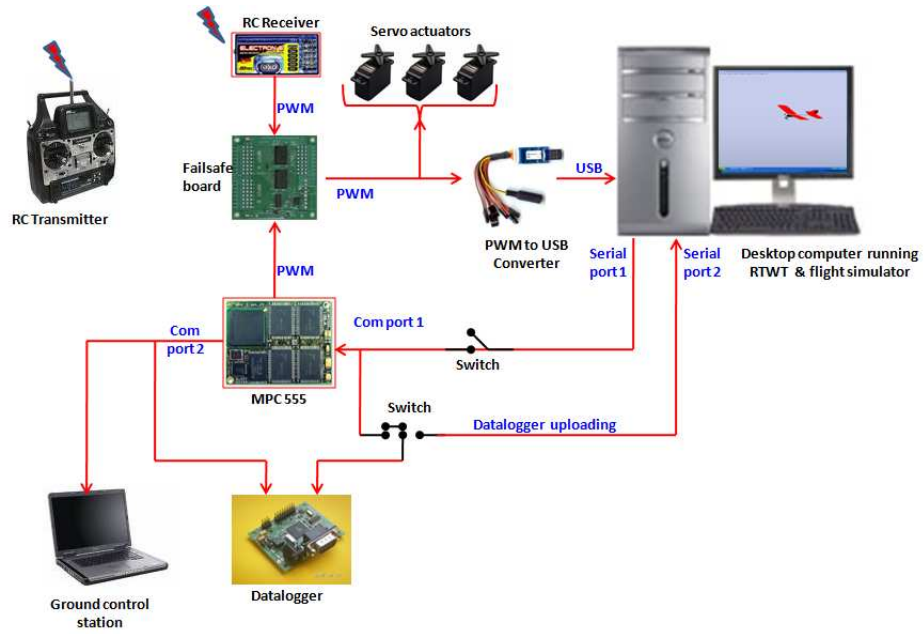


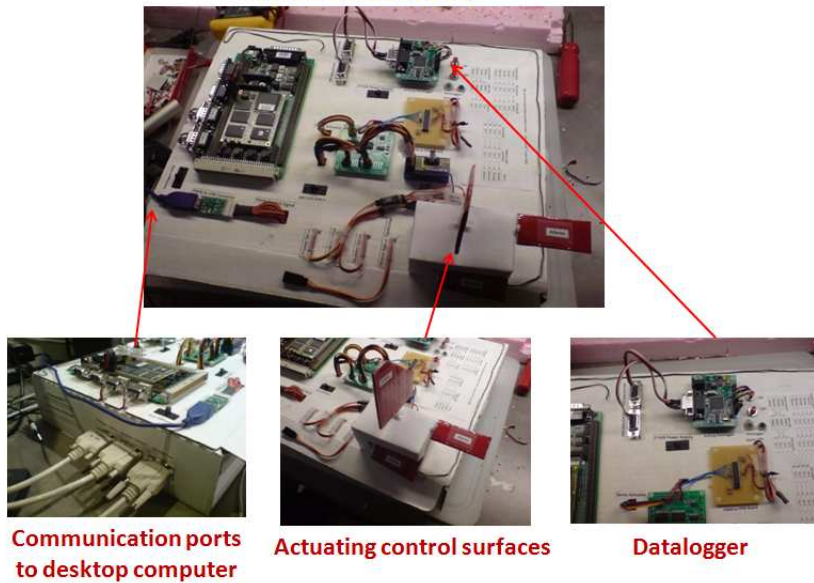
Figure 10: PIL software Simulink blocks architecture

while data can be logged from the datalogger or ground control station for analysis purpose. Figure 11(b) shows the physical setup of the PIL simulator.



(a) PIL hardware system layout

### PIL Simulator Setup



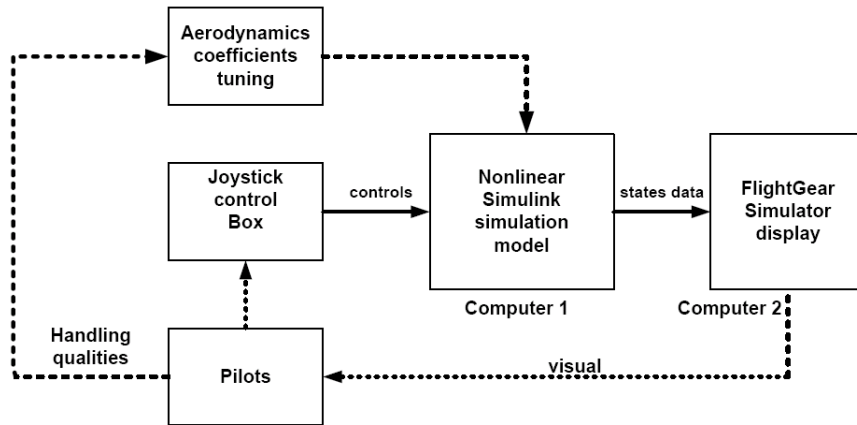
(b) PIL System setup

Figure 11: PIL simulator setup

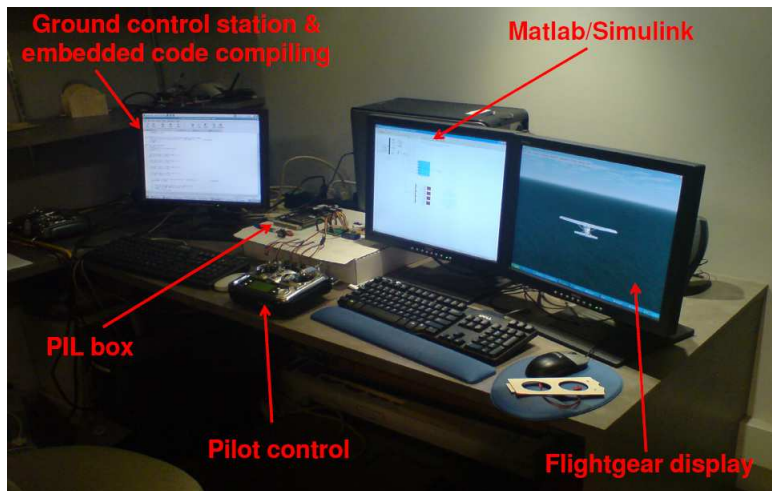


### 5.3 Simulator Flight Testing

The PIL simulation setup can be used as a RC flight simulator testing. An example of the use of simulator flight testing is in simulator parameters tuning, such as estimate of aerodynamic coefficients. Figure 12(a) shows the schematic of the tuning process. The joystick control box used is the same radio control box used to fly the actual vehicle. The control stick signals are input to the nonlinear Simulink model. The outputs from the simulation model are state responses of the vehicle and these data are used to run FlightGear simulator display that provides a visualization of the vehicle's motion. Two RC test pilots who have been flying the actual vehicle were asked to fly the simulator with different maneuvers. Based on their handling experiences with the actual vehicle, the dynamic aerodynamic coefficients for the simulation model were tuned iteratively until they felt the simulator model has almost similar handling qualities as the actual vehicle. Figure 12(b) shows the hardware setup of the simulator.



(a) Schematic of simulator parameters tuning



(b) Simulator station

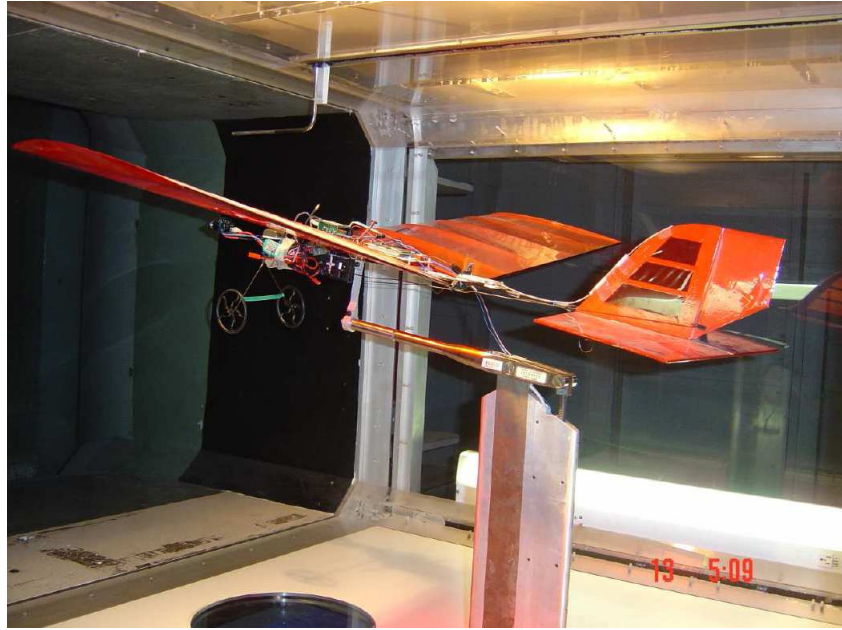
Figure 12: Simulator flight testing setup



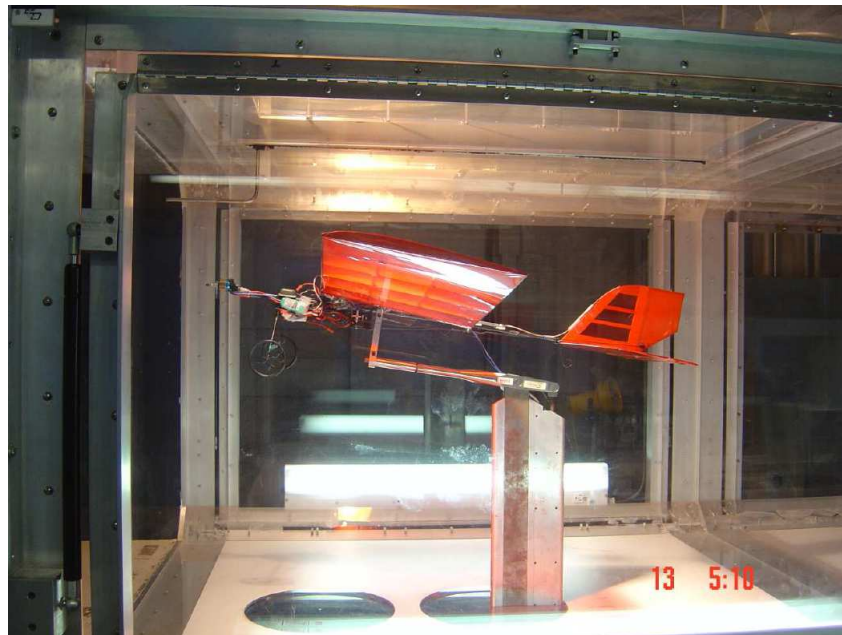
## 6 Wind Tunnel Testing Facility

The wind tunnel testing facility provides the capability of experimental determination of static aerodynamics coefficients. This provides the aerodynamic coefficients parameters required in modeling of the UAV for simulation as well as determining some of the stability characteristics of the UAV. Figure 13 shows the setup for wind tunnel testing of full-scale Yardstik UAV. The UAV is mounted on a force-moment balance which is used to measure forces and moments generated by the aerodynamics of the aircraft and this is used to compute various aerodynamic coefficients for the UAV.





(a)



(b)

Figure 13: Wind tunnel testing of Yardstik UAV



## 7 Flight Testing

Flight testing is an essential part of the UAV development cycle to collect flight test data for model identification, model validation, flight control validation, guidance law validation etc. All flight tests are performed by RC pilots. A toggle switch on the RC transmitter is used to toggle between computer control or pilot control. This toggle switch signal is used to control the failsafe board for switching the control mode of the UAV. At anytime of testing, the RC pilot can take over the control from the flight computer by toggling this switch on the RC transmitter and this ensures that the RC pilot has full RC control of the UAV at all time during the flight testing.

Figure 14 shows the schematic layout of the data acquisition system for measuring and recording time history input and output data and flight condition (airspeed and altitude) required. The IMU/GPS sensor output measurement data at 50 Hz. This includes the control stick input signals acquired from the RC receiver that are used to control the servo actuator which drives the control surfaces. The data are recorded on channel 1 of the dual channels datalogger. Since the IMU/GPS sensor does not give attitude angles ( $\phi$ ,  $\theta$  and  $\psi$ ), the flight computer is used to perform real-time attitude determination. The attitude angles computed are datalogged on channel 2 of the datalogger at 20 Hz. The reason of logging the attitude angles at 20 Hz is due to real-time limitation of the flight computer in outputting the data through the second serial port at a higher data rate. At the same time, important flight condition data such as airspeed and altitude information are sent real-time to ground monitoring station through data modems and this provides information for monitoring the flight test so that it is conducted at the required operating condition.

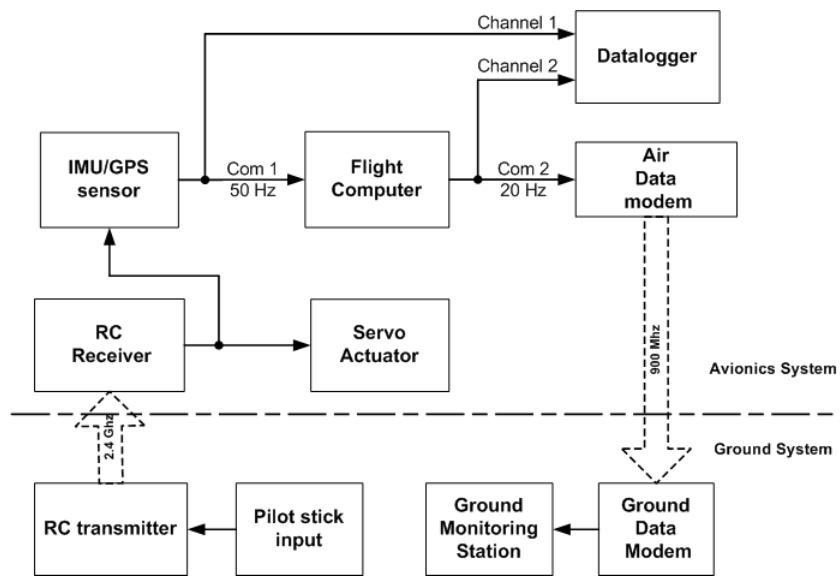


Figure 14: Schematic layout of data acquisition system



### 7.1 Indoor Testing

The indoor flight testing is conducted at the indoor running track facility (Field-house) at the UMN recreational facility. The airspace for the indoor flight is about 90 m long by 30 m wide and 25 m high. Figure 15 shows the Yardstik UAV during an indoor flight.



Figure 15: Yardstik UAV indoor flight

### 7.2 Outdoor Testing

Outdoor flight testing is conducted at different fields, depending on the flight test objective. Figure 16 shows the typical setup for the flight testing and Figure 17 shows the Ultrastick during an outdoor flight.



(a) GCS setup



(b) Ultrastick UAV

Figure 16: Ultrastick UAV flight setup



(a)



(b)

Figure 17: Ultrastick UAV flight