ONLINE NEAR REAL-TIME ONLINE SYSTEM IDENTIFICATION ON SMALL UNMANNED AIRCRAFT SYSTEMS

A Thesis

by

HAN-HSUN LU

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	John Valasek
Committee Members,	Gregory H. Huff
	Mobel Benedict
	Manoranjan Majji
Head of Department,	Rodney D. W. Bowersox

May 2018

Major Subject: Aerospace Engineering

ABSTRACT

An online near real-time system identification system is developed for generating locally linear models of Small Unmanned Air Systems. Automated control surface excitation inputs consisting of doublets, triplets, and frequency sweeps are implemented and used to assure consistency in the excitation and to eliminate errors introduced by user applied inputs. To provide reliable data for processing, a high frequency data acquisition unit is developed and implemented. In addition, a real-time vehicle monitoring system is used to provide a human-in-the-loop model update capability, with a goal of ensuring safety of the vehicle. Flight tests and modeling are demonstrated on a fixed-wing Small Unmanned Air System, with locally linear models generated during flight.

Observer Kalman filter identification is used as the primary identification algorithm with adjustments made for real-time identification purposes. Identified models are both stored and sent to the ground control station for ground control operator for update verification. Results presented in the thesis show that the system provides a capability for generating accurate locally linear models that are suitable for real-time flight control design using model based control techniques and post-flight modal analysis.

DEDICATION

To my wife Heather, my mother, my father for their support and understanding. This research is made possible because of their love and support.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. John Valasek, whose guidance, motivation, and support made this thesis possible. His invaluable suggestions and leadership will continue to be useful for me throughout my career.

I also want to thank my committee members, mentors and teachers, Dr. Moble Benedict, Dr. Majji, and Dr. Gregory H. Huff for giving me advices and directions through many inspiring conversations. Special thanks to Dr. Jer-Nan Juang, who introduced me to system identification and continues to be a great mentor in my research. I would also like to express my gratitude to his wife Lily Juang for the warm hospitality and support on my decision to pursue an graduate degree at Texas A&M University.

It is my pleasure to acknowledge my friends and colleagues at Vehicle Systems and Control Lab (VSCL) for keeping life and research not only fun but also productive. Thanks to my colleagues Vinicius Goecks, Cameron Rogers, Joshua Harris, Ezekiel Bowden, James Henrickson, and Frank Arthurs for always willing to share and guide me on hardware implementation. Douglas Famularo and Dipanjan Saha, you two have provided me valuable insight in adaptive control and control theory, and have been constantly providing me help on both life and research at College Station.

Last but not least, I would like to thank all the co-workers who worked on the precision agriculture project with me at the Texas A&M AgriLife Research's Brazos Bottom Farm. Dr. Yeyin Shi, Mr. Jeff Olsenholler, and many others, you all have been great to work with and I'm more than happy with what we have accomplished.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professors John Valasek, Moble Benedict, and Manoranjan Majji of the Department of Aerospace Engineering and Professor Gregory H. Huff of the Department of Electrical & Computer Engineering.

The data analyzed was collected by myself along with the aid from Ezekiel Bowden, Joshua Harris, and Vinicius Goecks.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported by a research fellowship from Texas A&M University Aerospace Engineering Department.

NOMENCLATURE

$G(s)$	Frequency response function
φ	Phase of frequency response function
ν	Shifted time index
l	Length of data set
H	Hankel matrix
Y	System Markov parameters
$\bar{Y}$	Observer Markov parameters
$\hat{A}, \hat{B}, \hat{C}, \hat{D}$	Identified system matrices
Y^o	Observer gain Markov parameters
$\bar{G}$	Observer Gain
f_N	Nyquist frequency
A_m	Modal system matrix
B_m	Modal participation factor
C_m	Mode shape
Ψ	Eigenvector matrix
λ	Eigenvalue matrix
$\overline{P}_m$	Identified modal observability matrix
$\overline{Q}_m$	Identified modal controllability matrix
q_n	n-th identified quality index
$\mathcal{T}$	Theil Inequality Coefficient

SUAS	Small Unmanned Aircraft System
GCS	Ground Control Station
VSCL	Vehicle Systems and Control Laboratory
MEMS	Micro Electro Mechanical Systems
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
OKID	Observer Kalman filter IDentification
ASCII	American Standard Code for Information Interchange
MRAC	Model Reference Adaptive Control
MPC	Model Predictive Control
CFD	Computational Fluid Dynamics
CIFER	Comprehensive Identification from Frequency Responses
ANN	Artificial Neural Network
SIDPAC	System IDentification Programs for AirCraft
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP/IP	User Datagram Protocol/Internet Protocol
DFTI	Developmental Flight Test Instrumentation
EKF	Extended Kalman Filtering
COTS	Commercial Off The Shelf
API	Application Programming Interface
GLOMAP	GLobal Orthogonal MAPping
PWM	Pulse Width Modulation
APM	ArduPilot Mega

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION AND LITERATURE REVIEW

1.1 Research Background and Motivation

1.1.1 Motivation

Real-time system identification is crucial part of modern flight control on both manned and unmanned vehicles. Vehicle dynamics change during flight due to variations in disturbance characteristics, vehicle angle-of-attack, fuel consumption, and vehicle damage. These factors results in a change of the vehicle characteristics, and often a redesign of the flight controller. Therefore, the capability of updating linear or nonlinear vehicle models enables a more robust, capable and modular control systems. By recursively updating the nominal model at each specified time t, the plant uncertainty can be reduced, which leads to a better flight model. This can be beneficial for most control techniques, and especially important for the ones that require a reference model, such as indirect Model Reference Adaptive Control (MRAC) and Model Predictive Control (MPC).

Also, while performing system identification through flight testing is an expensive endeavor for manned aircraft, SUAS have the advantage of inexpensive flight costs. As a result, in many cases it is both cheaper and faster to obtain models of SUAS through system identification. Doing so requires the ability to record state and control time histories at sufficiently fast rates. For autopilots and control systems, commercial actuators present in SUAS provides a maximum sample rate of around $50\,\mathrm{Hz}$. This sampling rate is sufficient for most standard vehicle designs, however, for agile aircrafts with fast dynamics, a data acquisition system that is capable of collecting sensor data at higher frequencies is needed [2].

Moreover, morphing vehicles with in-flight characteristic changes is also a research area with growing interest. In-flight wing span, camber morphing, or tilt rotor rotations changes the aerodynamic moments and forces applied on the vehicle. The challenge is that the transitional dynamics of morphing structures yield a time varying system. Transitional dynamics are hard to characterize using traditional off-line system identification process due to its inability to capture the transitional modal variations and inertia changes. Inaccurate models of the dynamic system may pose an issue for controller design as well as system stability robustness. Therefore, there is a need for precision aerodynamic modeling and controller design for vehicles with time changing characteristics. Real-time system identification is a promising candidate for yielding updated models for morphing structural control.

1.1.2 Research Background

High fidelity aircraft flight models are essential for modern controller design, aircraft dynamics, and autopilot development. Aircraft simulation models are mostly based on aerodynamic data tables from wind tunnel testing, vortex-lattic methods, or computational fluid dynamics (CFD). Ground based wind tunnel testing and CFD provides a good approximation of the actual flight condition, however, the high cost on time and expenses of wind tunnel testing restricts the number of experimental tests that can be conducted. Other issues with ground based testings are commonly related to the scalability of the test model, wind tunnel wall interference, and different characteristics on the flow [3]. In order to overcome issues related to wind tunnel testings, system identification using flight test data has been widely used for air and space vehicle modeling.

Flight control synthesis for aircraft with large flight envelopes and wide ranges of angle-of-attack and sideslip angle require much ground based and flight based testing in order to generate accurate locally linear models. An agile high speed aircraft modeling process can be seen in Figure 1.1. The conventional aircraft modeling process begins from using 1st order methods such as DATCOM and vortex-lattice to acquire low accuracy linear models. Higher order methods are then applied to construct low speed medium accuracy models via low order CFD and high order CFD or small scale wind tunnel testing to large scale wind tunnel testing. Large scale wind tunnel testing can be then used to acquire aircraft models with higher mach and angle of attach. Flight test is conducted in the final phase of the model development process where models are tested with real-world conditions. The process usually takes years if not decades and requires more iterations when a control law design is needed. The accumulated cost for human resources, test facilities, test materials over the years requires a large group of experts with a abundant funding.

For fast developing SUAS, resources for development are relatively limited, and thus, there is a need for rapid prototyping and rapid modeling. Figure 1.2 illustrates how online, near real-time system identification process is different from the conventional aircraft modeling process. The main drawbacks with the conventional process shown in Figure 1.2a are that the flight controller is designed offline and the flight data is analyzed post flight. This introduces an inability to modify the flight controller and adapt to structural or environmental changes during flight and increases the amount of time and money required for more iterations of re-modeling, re-designing, and flight testing.

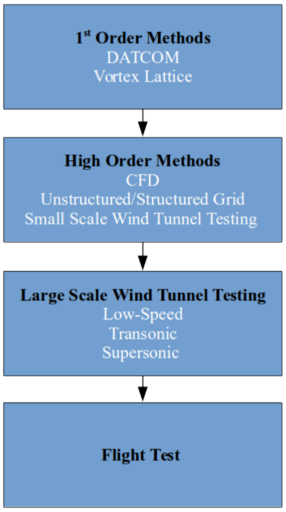
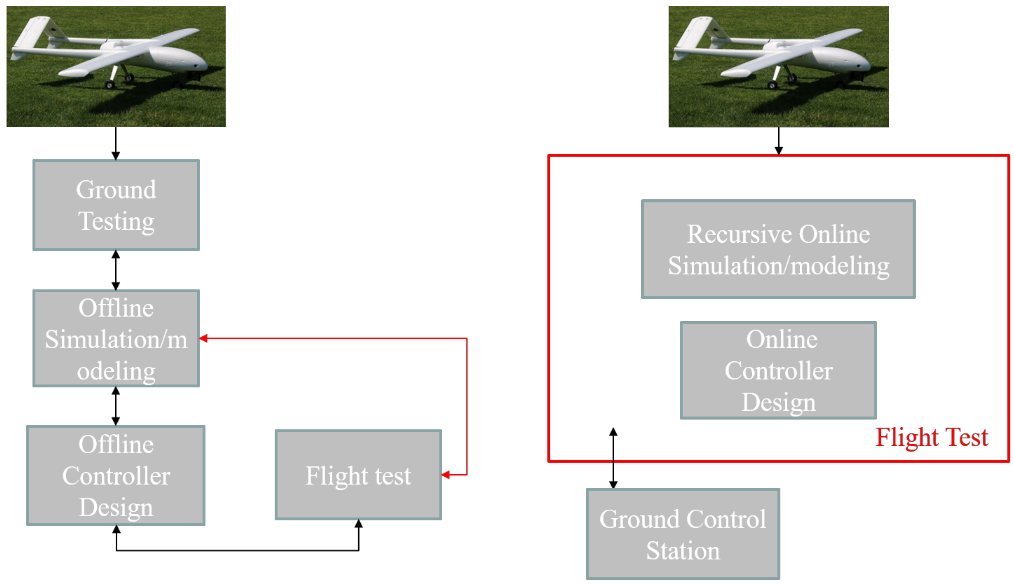


Figure 1.1: Model development process for aircrafts

Online near real-time system identification on the other hand updates the system model on-the-fly and calculations are performed onboard the vehicle as shown in Figure 1.2b. Instead of analyzing the stored data through post flight analysis, more data can be calculated or transferred to the Ground Control Station (GCS) through modern day Commercial Off The Shelf (COTS) products. This is made possible due to the vast development of semiconductors and micro-controllers, and is especially beneficial for applications such as SUAS. Single board computers such as Jetson



(a) Conventional post flight analysis procedure.

(b) Real-time modeling with online calculations procedure.

Figure 1.2: Comparison of modeling and flight test procedure.

TK1 by Nvidia, NUC by Intel, Raspberry Pi, or Beaglebone board ...etc. drastically increased the calculation speed while reducing the onboard computer in both size, weight, and price.

In addition, the advancement of vehicle simulation has made extensive ground testing much more affordable. Simulated aircraft dynamic models can acquired through open-sourced software such as JSBSim [4] or SimGen [5]. Hardware-In-The-Loop (HITL) and Software-In-The-Loop (SITL) testings can also be conducted through software that works with flight computers, these software include Airsim [6], ArduPilot [7], jMAVsim [8], and Gazebo [9]. Ground testing with just a subset of avionics and sensors is made possible while different vehicle responses can also be simulated to predict the actual aircraft behavior before flight testing. These improvements shortens the developmental time for real-time systems and are crucial for both real-time modeling and control design on SUAS.

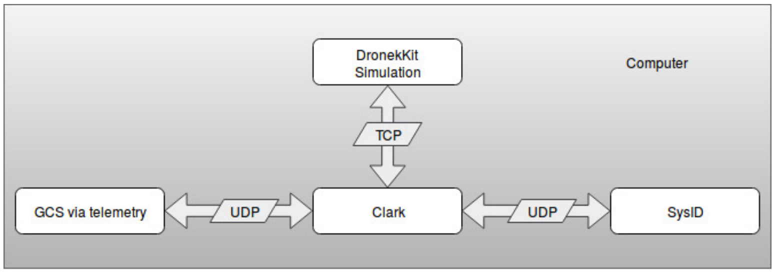


Figure 1.3: Software In The Loop development architecture

1.2 Literature Review: System Identification

Since the development of Kalman-Bucy filters [10] in the early 1960s, a lot of effort have been put in to filtering and estimation [11, 12, 13] and have been widely used for parameter estimation for dynamic models, vehicle navigation, and noise filtering. For system modeling purposes, several classes of algorithms exist for determining models of systems from experimental data. These algorithms can be partitioned into *parameter identification* algorithms, which determine parametric models of a system, and *system identification* algorithms, which determine non-parametric models. Parameter identification algorithms for aerospace systems include System Identification Programs for Aircraft (SIDPAC) [14], least-squares approaches, maximum likelihood approaches, and several neural network based approaches [15, 16]. Common system identification algorithms include the Eigensystem Realization Algorithm (ERA), Observer/Kalman Identification (OKID), the Comprehensive Identification from Frequency Responses (CIFER®) algorithm, Frequency Response Functions (FRF), and Observer/Controller Identification (OCID).[17, 18, 19, 20, 21]. OKID and Observer/Controller IDentification (OCID) were developed at NASA Langley Research Center during the 1990's providing a locally linear system model by directly utilizing input and output time histories. Reference [22] provides a historical overview of system identification approaches for flight vehicles. Also known as parameter identification, the method requires input and output histories of the dynamic system to calculate a model for further estimation.

The goal for system identification is to identify a set of system matrices that provide enough fidelity for on-board controller design on systems. Depending on the modeling technique, the

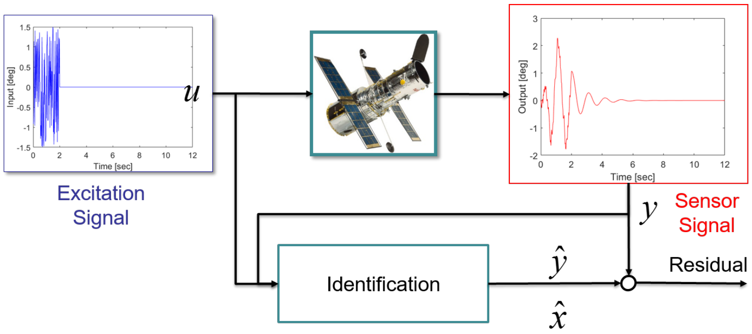


Figure 1.4: General system identification process.

process of determining the system behaviors can be classified as *a priori* modeling and *a posteriori* modeling [23]. *A priori* modeling, or morphological modeling, utilizes simple experiments to inquire the dynamic model that is partially known, while *a posteriori* modeling constructs a model by characterizing the system dynamics through analyzing the input u and output y without previous knowledge of the system. One of the goal of this thesis is to identify vehicle models with limited prior knowledge, therefore, the later method is what is chosen for performing system modeling and will be the referred method when system identification is mentioned in the following chapters. However, *a priori* modeling techniques will also be discussed and used for analysis and comparison.

The process of system identification is shown in Figure 1.4. Precise excitation u is required to excite the dynamic modes of the system without exceeding the identification limitations [24]. Input signal should be tailored according to the method of excitation, identification goal, and flight test environment. It has been shown that by formulating the optimal input design problem for linear system identification as a linear-quadratic optimal control problem, nontrivial solutions related to the maximum eigenvalue exist [25]. Recent studies have also shown promising experimental results by applying inputs that are mutually orthogonal in both time and frequency domain [26, 27].

Aircraft systems are time variant in nature due to the different configuration of controls during flight and the constant decrease in weight as a result of fuel consumption. The process of formulating a consistent algorithm for linear time variant systems is complicated, although novel algorithms have been developed to extend classical algorithms for time varying systems[28], in most practical cases, aircrafts are commonly modeled as a Linear Time Invariant (LTI) system. For system identification, linear and nonlinear system identification are two branches in research. Nonlinear models can be acquired through artificial neural network (ANN) [29, 30], Global Local Orthogonal MAPping (GLO-MAP) [31], or reinforcement learning [32]. It has the advantage of being able to produce a single model that is applicable for a large flight envelope. However, global modeling algorithms tend to seek out regions where the model is erroneously optimistic. For instance, if the model has modeling error in the prediction process, which results in a lower predicted cost compared to the actual cost, model based learning algorithms are keen to go explore states where the cost can be improve the model the most. This may raise an issue in a three dimensional space where there is multiple options for the model to explore, and will lead to inefficiency in finding an optimal solution. Therefore, when the number of nodes and system output dimension increase, the performance of decreases drastically due to the system complexity.

Moreover, for vehicles with high maneuverability, one general model often cannot cover the whole flight envelope. For these types of vehicles, local models are often preferred. Consequently, global nonlinear models may be computationally expensive for systems with time-changing dynamics and changing trim conditions and not suitable for real-time applications. Therefore linear system identification methods have been widely used on real-time systems for system modeling and controller design.

1.3 Literature Review: Online Real-time Online System Identification

Real-time online system identification started to gain interest since the late 1990s on both rotor and fixed wing manned aircrafts. Several studies addressed the feasibility of real-time identification [33, 34]. Although these studies were either limited to reduce order systems or encountered periods of data dropouts due to instrumentation restrictions, they laid the foundation for recent UAS system identification related studies.

The data dropout issue has been resolved benefiting from the evolution of flight test instruments and improvement in computational speed. In 2003, Valasek and Chen [35] showed that OKID is feasible for on demand, online system identification for aircraft longitudinal and lateral/direction modes for fixed wing manned aircrafts in simulation. Around the similar time, Morelli approached the same problem with another technique, it was shown that real-time and recursive parameter estimation of a low order equivalent system identification can be applied on flight data to identify moment coefficients [36]. Additionally, Grauer[37] applied an alternative of the SIDPAC using output-error parameter estimation to point out modifications needed to improve the efficiency of real-time system identification.

Although there has been some success on real-time system identification for manned aircraft, it is only until recently have more real-time system identification related work been published. Puttige performed HITL testing using an autoregressive based multi-network approach [38]. In 2009, Debusk applied a Fourier transform regression method to estimate aerodynamic derivatives in the frequency domain [39]. The identification methods that are used are either under the broad recursive least squares parameter estimation category or frequency domain analysis. The current state of the art research in real-time UAS modeling is lead by NASA Langley Research Center. The NASA Learn-to-Fly project pointed out that the ability to model behaviors due to structural/propulsive/ aerodynamic/flight control interactions, etc., is crucial to the successful and efficient development of future airplanes with increased performance as a goal. The goal of the project is to autonomously develop vehicle characterization and control strategies, up through the ability to fly a vehicle, with minimum human interaction and time [40]. As one of the key components, real-time system identification has become a more popular topic. In 2016, Morelli showed that using multivariate orthogonal functions along with real-time recursive modeling orthonormalization based on QR decomposition real-time nonlinear modeling can be acheived [41, 42]. This method is based on SIDPAC along with reinforcement learning to perform effective estimation of the nondimensional aerodynamic force and moment coefficients.

Rather than applying the well established frequency domain method to perform real-time system identification, this thesis focuses on approaching problem with a different method. By applying the time domain OKID method, the goal is to extend the previous offline system identification work at Texas A & M University Vehicle Systems and Control Laboratory (VSCL) and develop near real-time online identification capabilities especially for SUAS. At VSCL, there have been success in modeling SUAS [43, 44]. In order to generate the good quality models offline via post processing, a flight test instrumentation system is developed to address these data dropout issues. This thesis introduces an updated flight test instrumentation system for SUAS parameter and system identification flight, superseding an early system used in previous work by the authors [45, 46]. The proposed instrumentation system requires no external or proprietary software for data logging. It offers modularity and extra digital and analog input/output ports for additional sensors. Its high-frequency sampling capability, at a maximum of $100\,\mathrm{Hz}$, is also ideal for the system identification and modeling applications. Using updated avionics and flight control system along with an recursive OKID algorithm, it is intended to avoid the need of extensive model training for ANN methods or vehicle knowledge such as inertia measurements. Due to the *a posteriori* nature of OKID, the algorithm does not require vehicle specific knowledge system modeling but simply rely on correlations of the input-output data, and therefore, is a good candidate for online near real-time system identification for vehicles with unknown or changing characteristics.

1.4 Approach Summary

The proposed research scopes both the software and hardware development of a highly portable real-time system identification system that is capable of providing in-flight model updates in real-time while maintaining reasonable fidelity. The proposed online real-time identification system should be capable of modeling systems with real-time changes such as changing flight conditions without extensive ground testing and can be used for future novel UAS developments.

The main objectives of this thesis is to:

1. Develop data acquisition unit with the ability to provide high frequency data for system identification on small UAS with direct α, β measurements and direct control surface measurements in degrees or radians of deflection.
2. Develop software functions capable of performing precise servo actuation with the goal to eliminate pilot imprecise excitations for system identification.
3. Use the OKID structure to develop a system capable of performing on-board recursive system identification with real flight test data.
4. Perform online near real-time system identification with high rate of successfully generating linear vehicle models during flight comparing to offline methods.
5. Develop and implement a human-in-the-loop real-time identification procedure with online calculations, real-time data transfer, and near real-time system identification with the goal to update vehicle models in the during flight test.

The paper is organized as follows. Chapter 1 introduces the development of system identification and indicates the need for online near real-time system identification. The system design and the system characteristics are addressed in Chapter 2, while the hardware selection and integration is detailed in Chapter 3. Chapter 4 and Chapter 5 explains the fundamentals of aircraft dynamics along with system identification used for online and offline data analysis. Ground test and flight testing results are presented in Chapter 6, comparing onboard and offline calculation results. Finally, conclusions and future recommendations are discussed in Chapter 7.

2. SYSTEM DESIGN

2.1 Challenges and Current Developments

The challenges for the system can be sorted into three categories: efficient online mode excitation, maintaining the consistency of the identified model, and the implementation of real time system identification on a SUAS system.

Excitation of aircraft modes is crucial for system identification, the challenge lies in the success rate of exciting the targeted dynamic modes. Human piloted maneuvers have shown to be inconsistent and the well established input methods requires minimal environmental disturbances. The developed system will resolve the issue by introducing an automated excitation mechanism and explore alternative excitation methods to precisely excite the system modes.

Due to the inherent concept of online near real time system identification, the model is updated in a near real time fashion. The real question is that after the modes are excited, is the identification method capable of capturing the dynamic modes using the input-output pair consistently, and will this system be reliable enough to ensure flight critical functionalities are not altered. Therefore, in order to reduce the risk of losing a vehicle during flight, certain criteria and procedures are to be designed to ensure that a poorly identified model does not effect the vehicle system model.

Moreover, the development of a sensor independent, modular, and fast responsive system is another challenge. The main issue is that the existing COTS products either lack modularity or does not provide a reliable data stream for online system identification purpose. Existing real-time systems do not provide a high enough sampling rate and does not meet the payload requirements of small scale SUAS. Accordingly, it is challenging to develop a fully integrated system that is expandable, modular and light weight, while providing enough sample frequency as well as real time ground air communication for online identification using a SUAS.

2.2 Software Development

Three subsystems coordinate with each other simultaneously on the UAS in order to perform online system identification. The subsystems include a communication subsystem, the data collection subsystem, and the system identification subsystem. Figure 2.1 displays the flow of data between each subsystem, as well as how data is logged for each individual program. The highlighted blue blocks are the data logs that are not utilized online but availble to access after the vehicle has landed. The orange blocks are logged data types that are accessed or referenced in the online calculation process. Finally, the line styles differentiate the operational commands(in dashed lines) from data (solid lines) that is transfered, and the arrows indicate the data flow being one or dual-directional flow.

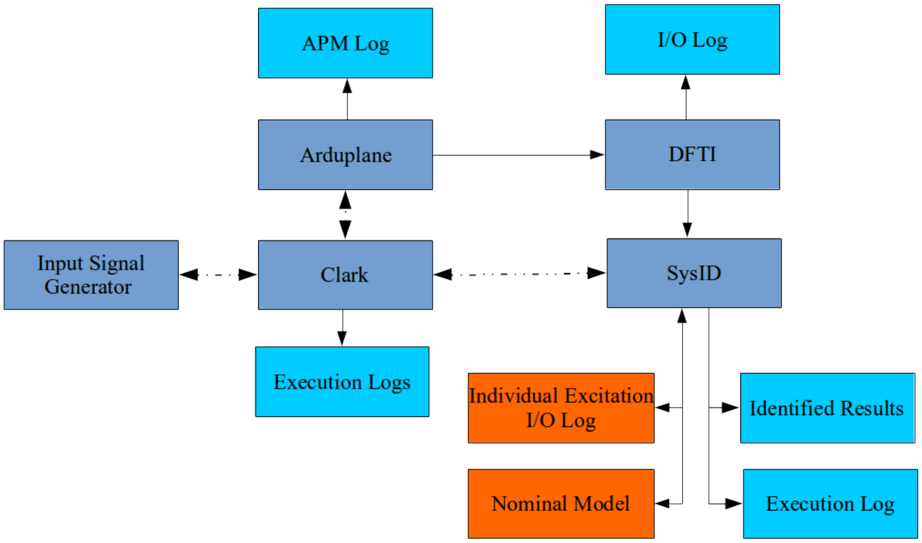


Figure 2.1: Data transfer between different subsystems

2.2.1 Data Logging

There are several different types of log file with various purposes. Data log types for the online system identification structure include execution logs, standard data collection logs, and result logs. Execution logs records the runtime processing info and system errors, both Clark and the system identification software have its individual execution logs. On the other hand data collection logs are recorded on different parts of the subsystem but only the system identification subsystem use portions of the recorded data to perform online calculations. The individual excitation input/output log, nominal model log, and identified result logs are conditional logs. That is, only when the specified execution conditions are met will these logs be activated to initiate recording. The individual I/O log records the input and out signals for each excitation. The process starts logging when the pilot flips the excitation switch from LOW to HIGH and ends when the excitation status is returned to LOW. After the excitation status returns to low, the system identification subsystem will generate the identified results to the result file. Failure in identification will return a matrix of the expected size with values of 1 in all entities. This matrix is stored in the result log file and at the same time sent back to Clark buffer to wait to be transferred to the GCS. The result sent back to the GCS operator is shown on the GUI to indicate the result of the previous identification.

Each log has its unique file name, this file name is a combination of file number, identification type, batch identification decision, and date for ease of offline data comparison and debugging. The nominal model is logged and read in a slight different way. The log not only records the current nominal model that have been identified through the identification process, but also perform a simple comparison of the data

2.2.2 Communication Subsystem: Clark

Clark [47] is a multi-agent control framework that provides a user with the ability to autonomously command various inputs from a mission control station. It is executed as a separate command line application on the BeagleBone Black and is written/operated in Python 2.7. This specific Python version was selected due to the dependencies on Dronekit and Google Protocol

Buffers that the current software structure heavily rely on for communication between the autopilot and different agents. The framework consists of a three-tiered software architecture connecting the autopilot, radio transceiver, and connected payloads — an abstraction layer provides a specific location for implementing control. Figure 2.2 depicts this three-tier architecture. The top tier is the mission tier where the mission action is assigned. Vehicle control actions are calculated and appointed in this layer, and therefore, system identification related functions such as the input signal generator function are executed and servo inputs are assigned in this layer.

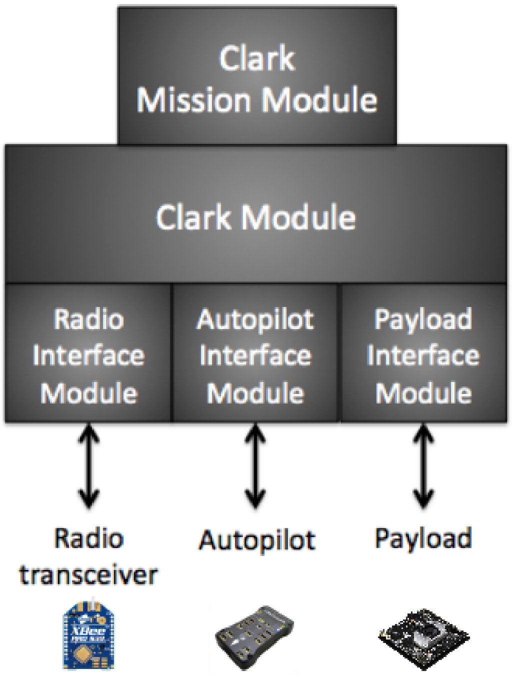


Figure 2.2: Three tier architecture of the Clark framework

The Clark module handles the communications between the autopilot, radio, and payload interface using individual TCP/IP or UDP/IP connections. The lower layer interface classes then call their related functions to perform the the action. For example, after the radio interface class received a command to perform a sinusoid sweep excitation, the Clark layer will then update the identification type in Clark Mission layer to calculate the designated input signal. After the calculation, this signal will then be passed to the autopilot interface via Clark to perform the servo actions through Dronekit. From the example it can be seen that to perform a simple action all three layers of Clark needs to be highly integrated and maintain efficient communication.

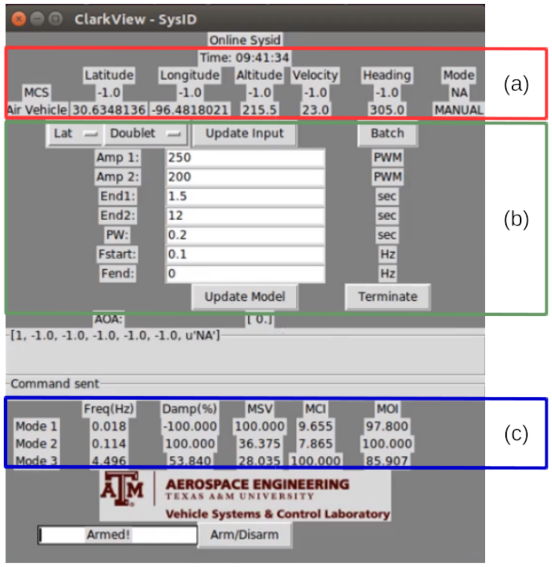
2.2.3 Communication Interface

There are three goals for the communication interface. The first one is to be capable of monitoring basic vehicle status and receive updates in near real-time fashion. For system identification in general, the flight altitude, flight speed, and angle of attack of the aircraft changes the vehicle dynamical behavior. The implementation of the vehicle status can be seen in part (a) of Figure 2.3. Another goal is to perform real-time modifications on the excitation inputs while the vehicle is in the air, it is crucial for online applications to enable the GCS operator to make in-flight tweaks on the excitation signal. This can be seen in the part (b) of the same image. Finally, in order to close the loop for the process of online system identification it is also important to get near real-time feedback from the air vehicle on the identification quality. Part (c) of the GUI is designed for this purpose. The identified natural frequency, damping ratio, and mode quality indexes are shown in order to provide GCS operator feedback on the previous execution. Note that the mode quality selection methods will be explained in Section 5.2.

2.2.4 Data Acquisition Subsystem: DFTI

The modular software is designed to support multiple sensors and airframes while interfacing with other systems through UDP/IP or TCP/IP communications. DFTI was implemented as a Linux command line application in C++11 using the Qt5 framework to allow for easy concurrency and threading with the signal/slot paradigm. While this introduces some overhead due to the Qt5 event loop, testing has demonstrated that the system is able to log sensor data at the desired $100\,\mathrm{Hz}$ although the actual rate at which data is written to the output files is slightly lower. The supported sensors and the sensor configurations are detailed in the Chapter 3.

The software package is configured to work with a Debian based single-board computer but can



Online Sysid

Time: 09:41:34

	Latitude	Longitude	Altitude	Velocity	Heading	Mode
MCS	-1.0	-1.0	-1.0	-1.0	-1.0	NA
Air Vehicle	30.6348136	-96.4818021	215.5	23.0	305.0	MANUAL

Lat

Doublet

Amp 1:	250	PWM
Amp 2:	200	PWM
End1:	1.5	sec
End2:	12	sec
PW:	0.2	sec
Fstart:	0.1	Hz
Fend:	0	Hz

AOA: [0.]

[1, -1.0, -1.0, -1.0, -1.0, -1.0, u'NA']

Command sent

	Freq(Hz)	Damp(%)	MSV	MCI	MOI
Mode 1	0.018	-100.000	100.000	9.655	97.800
Mode 2	0.114	100.000	36.375	7.865	100.000
Mode 3	4.496	53.840	28.035	100.000	85.907

Armed!

Figure 2.3: Clark communication interface breakdown

be modified to work on other Linux based operating system via UART ports. A configuration file is required to assign sensors to the designated port. It is important to assign the correct configuration to the correct port to guarantee quality data. Initially DFTI was developed to accurately log vehicle flight data for offline system identification. Therefore, data was logged on to a CSV file and can only be accessed after flight. Due to the need of real-time data streaming for online system identification, an additional server function was added to support this requirement. The UDP server can be turned on or off via the configuration file, and the native byte order and 1-byte padding is used for the structure with no conversions made to the network byte order. UDP is a connectionless oriented protocol without handshakes between the server and the client. Although no flow control or guarantee of receipt, for a confined system it reduces the overhead and has the benefit of providing time-sensitive data without package delays due to retransmission.

2.2.5 System Identification Subsystem

The system identification subsystem takes the data acquired by DFTI and performs the identification, calculates the identification quality indexes, and generates the log files for the identified results. The script is codded in Python3 due to the intensive math and matrix calculations involved in applying OKID.

The OKID algorithm used is based on the System/Observer/Controller Identification Toolbox (SOCIT) [48] that is codded in Matlab, and is modified as well as recoded into Python3 to meet the needs for a SUAS weight and computation limitations.

Communication between DFTI and Clark are established through UDP/IP and serves as a client for both communication ports. To eliminate overhead on the communication part of the subsystem and reduce the lag time for data packing and unpacking, the data does not use Protobuf to pack and unpack messages, but analyzes the received string directly. All the user-defined identification related parameters are passed from Clark to the subsystem, this includes the identification type, batch decision, and nominal model update decision. The identification type provides the identification subsystem whether to identify a lateral/directional, longitudinal, or a full lat/lon model. The decision will alter the size of the truncated system matrix and the number of expected modes. While the batch decision determines whether to stack the input and output data from different experiments to perform OKID or simply use the input-output set from a single excitation, the nominal model update decision updates the saved nominal model of the system.

After the system identification is finished, the program returns a string with the identified frequency, damping ration, identification quality indexes, and eigenvalues to Clark. The information is then packed into several Clarklink messages to be transmitted down to the GCS.

3. FLIGHT TEST INSTRUMENTATION*

The flight test instrumentation consists of an embedded computer and a selection of sensors to record system states and control effector positions. The instrumentation was selected and integrated on a commercial-off-the-shelf (COTS) SUAS. The electronic components were chosen to meet the requirements set in Appendix B for the purpose of providing high frequency control and state measurements for near real-time system identification. It is desired to keep the flight test instrumentation system separate from the flight control system so that faults in the instrumentation do not affect anything in a flight-critical path for safety-of-flight reasons. A related benefit of this separation is the ability to upgrade the flight control system separately from the instrumentation. The previous system integrates additional sensors into the flight software on a Pixhawk autopilot, which requires a custom firmware to be flashed prior to system identification flights and precludes easy firmware updates. In this Chapter, detailed hardware information as well as vehicle specifications will be presented. Note that the system presented in this chapter has been previously published.

3.1 Vehicle Description

The 1/4 scale Hangar-9 PA-18 Super Cub SUAS is the base platform for testing the flight test instrumentation system. The $700 COTS Super Cub has a wingspan of 8.8 feet, empty weight of 16.6 pounds, and endurance of 30-45 minutes with extended batteries. The aircraft has a $295\,\mathrm{kV}$ E-Flite Power 110 electric brushless motor, $85\,\mathrm{A}$ HV brushless Electronic Speed Controller (ESC), and an APC 19×10E propeller. Due to its rugged construction and excellent low speed flying qualities it is suitable for rough field operation and short takeoff and landing. The Super Cub also excels at carrying multiple sensor payloads. Figure 3.2 shows the Super Cub airframe used in this paper.

---

*Reprinted with permission from "Online Near Real-Time System Identification on a Fixed-Wing Small Unmanned Air Vehicle" by Han Hsun Lu, Joshua Harris, Vinicius G Goecks, Ezekiel Bowden, and John Valasek, in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on. IEEE*, pp. 1696-1705, copyright 2017, IEEE

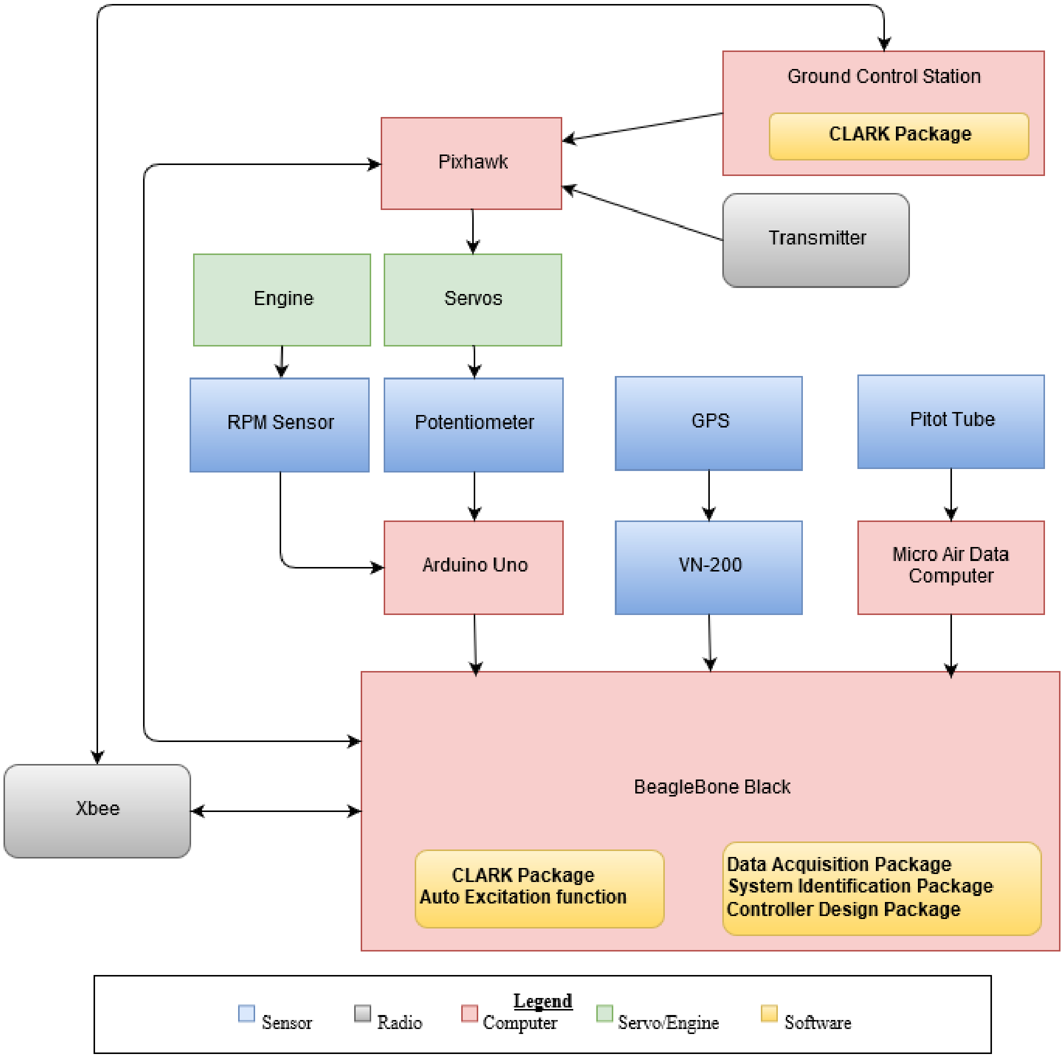


Figure 3.1: Diagram of the integrated ssensors and microcomputers.

3.2 Developmental Flight Test Instrumentation

The Developmental Flight Test Instrumentation (DFTI) [49] is a modular data acquisition system developed for the purpose of sensing and logging the system states and control effector positions at 100 Hz while remaining compact for easy transplant between different SUAS. DFTI, as seen in Figure 3.3, currently support sensor communications including

- Air Data: Aeroprobe Micro Air Data Computer (RS-232)
- INS: VectorNav VN-200 INS (UART/RS-232/SPI)

Figure 3.2: Hangar-9 1/4-Scale PA-18 Super Cub

- Control Surface Deflections: linear potentiometers via Arduino
- Engine RPM: EagleTree Brushless RPM Sensor V2 via Arduino
- Autopilots: Mavlink-based autopilots

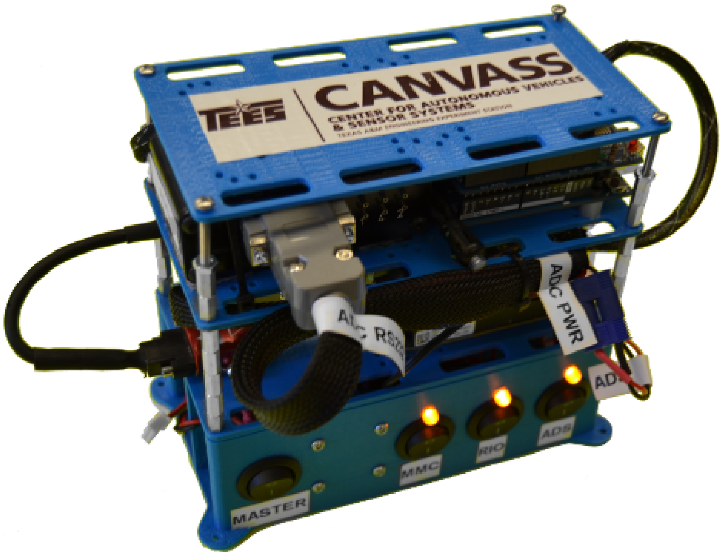


Figure 3.3: DFTI sideview

The empty weight of the system is 0.44 lbs and requires a 3-cell 11.1 V Lithium Polymer (LiPo) battery to power the sensors. To accommodate different needs of sensors for different missions, the system is mounted on a three tier 3D printed structure with each tier designed for easy add and removal. The powering system is on the bottom layer of the structure using three LED lights to indicate the on/off state of the switches. Additionally, a voltage regulator is integrated to provide both 5 V and 12 V needs for different applications.

3.2.1 Flight Mission Computer

The flight mission computer serve as the hub of all the sensors, communication hardware, and perform online calculation. The mission computer is designed to perform multiple tasks including data logging, ground-to-vehicle and vehicle-to-ground communication setup, and online system identification. Although any single board computer that can meet the specified requirements can be interchanged for different applications. The Beaglebone Black Rev C is selected as the micro processor for this research weight, size, and number of UART ports.

The flight mission computer along with the additional developer capes are displayed in Figure 3.4. A TITAN Expansion I/O cape is stacked for extra storage and USB connections, and a Sparkfun developer cape (red board on top layer) is added with soldered connectors, data logging switch, and a Real-Time Clock (RTC) chip.

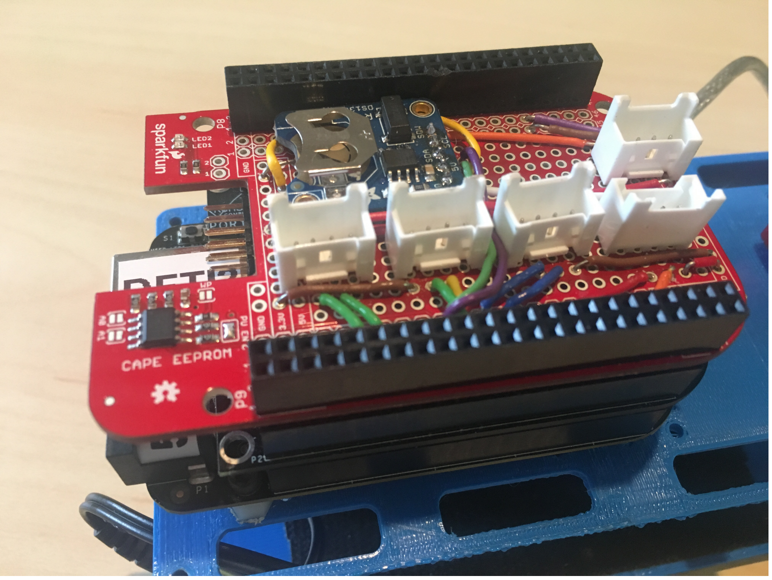


Figure 3.4: Beaglebone Black as the mission computer

3.2.2 Sensors

A VectorNav VN-200 is chosen as the IMU providing attitude angles (ψ, θ, ϕ), angular rates (p, q, r), accelerations (A_x, A_y, A_z) and GPS time. The MEMS-based 10-axis 6-DOF IMU incorporates a 3-axis gyroscope, 3-axis magnetometer, and a barometric pressure sensor, the sensor has a navigation processor that runs proprietary Extended Kalman Filtering (EKF) to optimally combine high bandwidth inertial measurements with high accuracy low bandwidth GPS measurements. The VN-200 supports both ASCII and binary protocols for encoding the sensor data. The protocol choice and sensor outputs to communicate can be configured in the VectorNav Sensor Explorer software. Binary encoding of the data was used to maximize the data rates and the signals selected for logging include:

- GPS time (ns)
- attitude quaternion
- body-axis angular rates ($\mathrm{rad\,s^{-1}}$)
- latitude/longitude/altitude ($\mathrm{deg/deg/m}$)
- North-East-Down velocity ($\mathrm{m\,s^{-1}}$)
- body-axis accelerations ($\mathrm{m\,s^{-2}}$)

The GPS time signal from the INS is used to match times with other systems and can also be used to set the system clock on the BeagleBone Black to the correct date and time when network time is unavailable.

The VN-200 IMU unit is mounted as close to the vehicle center of gravity as possible while the GPS antenna is placed near the top of the fuselage to minimize interference. The inertial navigation sensor is shown in Figure 3.5.

Aeroprobe five-hole probe (5hp) along with the μADC air data computer in Figure 3.6a and Figure 3.6b are used for measuring the total velocity V_t, angle of attack α, and sideslip angle β.

Figure 3.5: VectorNav VN200.

The μADC outputs data over an RS-232 serial connection in an ASCII data format. An RS-232 to 3.3 V TTL converter is used to shift the voltage levels of the serial connection for interfacing with the instrumentation computer. Initially an RS-232 to USB converter was used, but testing demonstrated that the fast sampling of the μADC along with other sensors was sufficient to saturate the USB bus on the computer. This caused the watchdog timer to count down and trigger an interrupt. The μADC is mounted inside the fuselage of the Super Cub while the 5HP is mounted under the right wing and is connected by pressure tubing though the half-span to the μADC. This is a compromise between accuracy and convenience as long tubing lengths reduce the sensor accuracy.

Previous iterations of VSCL instrumentation recorded commanded surface positions as pulse width modulated signals. This has several drawbacks, including identifying the effect of the actuator dynamics as part of the aircraft dynamics and the inability to detect failures in the actuator. Direct measurement of control surface deflections is desired to address these issues. In combination with the actuator commands logged on the flight control system, direct surface position measurements can also be used to identify models of the actuator dynamics. Additionally, control surface deflection measurements can provide additional feedback signals for techniques such as

(a) 5hp mounted on the wing (b) μADC air data computer

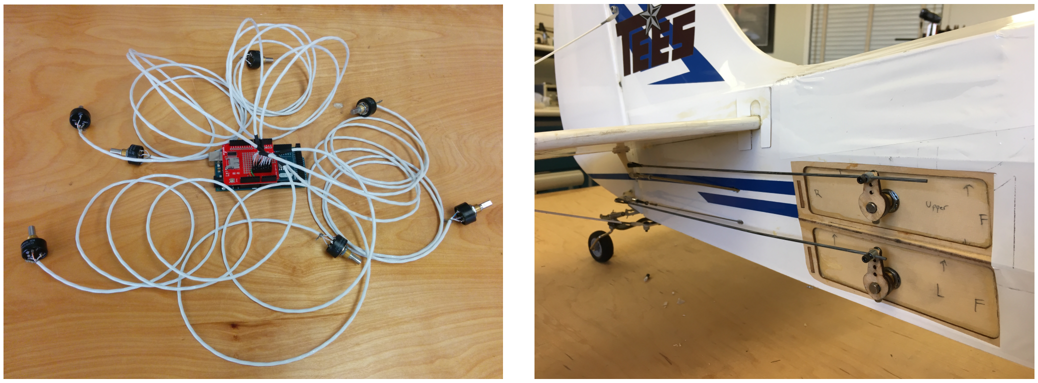
Figure 3.6: Aeroprobe air data system.

Control Rate Weighting [50] to prevent phenomena such as pilot- induced oscillation, as well as reducing wear on actuators. The Super Cub model has the following aerodynamic control surfaces: left aileron $\delta_{\mathrm{A_L}}$, right aileron $\delta_{\mathrm{A_R}}$, left trailing edge flap (TEF) $\delta_{\mathrm{TEF_L}}$, right TEF $\delta_{\mathrm{TEF_R}}$, left elevator $\delta_{\mathrm{E_L}}$, right elevator $\delta_{\mathrm{E_R}}$, and rudder δ_{R}. The aileron, elevator, and TEF surfaces are ganged so there are the three standard aerodynamic controls (aileron, elevator, and rudder) and the TEFs are deflected symmetrically. Using the convention found in Klein and Morelli [14], Eq. (3.1) is one such combination for aileron and elevator. Positive surface deflections generate negative aerodynamic moments under this convention. Individual control effectors follow a right-hand rule for determining positive deflections.

$$\delta_{\mathrm{A}} = \frac{1}{2}(\delta_{\mathrm{A_R}} - \delta_{\mathrm{A_L}}) \qquad \delta_{\mathrm{E}} = \frac{1}{2}(\delta_{\mathrm{E_L}} + \delta_{\mathrm{E_R}}) \tag{3.1}$$

Ganging is accomplished through RC servomechanism Y-splitters and servo orientation. However, since each surface is controlled by its own servomechanism the airframe could be modified to allow independent actuation of each control surface by a control law and control allocation algorithm. Independent actuation would also allow differential deflection of roll control effectors to reduce adverse yaw effects. As a result each surface has a potentiometer mounted for position feedback in order to allow for identifying models with each surface as an input.

Although the Beaglebone Black provides several analog input ports, the analog ports ranging from 0 V (GND-ADC) - 1.8 V (VDD-ADC) does not provide enough resolution. Therefore



(a) Arduino attached to potentiometers (b) Integrated potentiometer

Figure 3.7: Real-time input measurement system

the real-time input measurement system utilizes an Arduino UNO to provide 5 V ADC encoding. The system encodes measured data from seven BI Technologies 6127V1A360L.5FS linear potentiometers and an brushless motor engine rpm sensor made by EagleTree before sending it to DFTI through UART. The measured potentiometer angles are then mapped to acquire the aircraft control surface deflections. The potentiometers are rigidly mounted to the airframe and are connected to the control surfaces by standard RC control horns and rods. Figures of the connection and installation of the potentiometers is shown in Figure 3.7.

Specifications of the individual sensors and processors can be found in Table 3.1. The capability to measure all flight critical states minimizes the probability of estimation error due to approximations. For control surface deflections, an Arduino UNO is used for acquiring analog measurements of the control surfaces directly. Previous flight test instrumentation at Texas A&M University record commanded surface positions, usually as pulse width modulated signals, for providing control time histories for system identification. This has several drawbacks, including identifying the effect of the actuator dynamics as part of the aircraft dynamics and being unable to detect failures in the actuator. Direct measurement of the control surface deflections is desired to address these issues.

Table 3.1: DFTI Specifications

Data Collection System	
Instrumentation Computer	BeagleBone Black Rev. C
Processor	TI AM335x Sitara $1\,\mathrm{GHz}$ ARM® Cortex A-8
RAM	512 MB DDR3
I/O	USB, Fast Ethernet, HDMI, 2×46 GPIO, 4 usable 3.3 V TTL UART
Air Data System	
Air Data Computer	Aeroprobe Corporation Micro Air Data Computer μADC
Flow Angle Range	$\pm 20°$
Airspeed Resolution	$0.36\,\mathrm{m\,s^{-1}}$
Max Flow Angle Error	$\pm 1.0°$
Max Calibrated Airspeed	$64\,\mathrm{m\,s^{-1}}$
Current Draw	<$390\,\mathrm{mA}$ @ $12\,\mathrm{V}$ DC
Inertial Navigation System	
INS	VectorNav VN-200
Accuracy (Pitch/Roll)	$0.5\,°\mathrm{RMS}$
Accuracy (Heading)	$0.5\,°\mathrm{RMS}$
Angular Resolution	<0.05°
Gyro Noise Density	$0.0035\,°\,\mathrm{s^{-1}}\,\sqrt{\mathrm{Hz}}$
Gyro Alignment Error	±0.05°
Gyro Resolution	<0.02°
Remote Measurement System	
Instrumentation Computer	Arduino Uno Rev. 3
Potentiometers	TT Electronics/BI 6127V1A360L.5FS
Linearity (Potentiometers)	$\pm 0.5\,\%$
Control Surface Rotation Angle	$0° - 360°$, Continuous
Engine RPM sensor	Eagle Tree Brushless Motor RPM Sensor V2
Accuracy (Engine RPM)	$\pm 100\,\mathrm{rpm}$

3.2.3 Avionics

The flight control system of the Super Cub SUAS consists of the open-source ArduPlane firmware running on the open-hardware Pixhawk flight controller. The hardware specifications of the Pixhawk are listed in Table 3.2. The ArduPlane flight control software is operated in the MANUAL flight mode during identification flights, which is a direct stick-to-surface mode with no augmentation [51]. Since the Super Cub aircraft is dynamically stable, augmentation is undesirable for identification flights as the objective is to identify the open-loop model.

Figure 3.8: Pixhawk2 autopilot.

Table 3.2: Pixhawk Autopilot Specifications [1]

Processor	32-bit STM32f427 Cortex M4
Process Speed	186Hz
Memory	256 kB RAM
Failsafe Processor	32-bit STM32F103
Gyroscope	16-bit ST Micro L3GD20H
Accelerometer/Magnetometer	14-bit ST Micro LSM303D
Accelerometer/Gyroscope	Invensense MPU 6000 (3-axis)
Barometer	MEAS MS5611

3.2.4 Radio Communication

For data transceivers used for system identification, the XBEE transceivers are connected to the flight mission computer via USB connection. The Digi XBEE Pro 900 HP is used to operate with the DigiMesh protocal under API mode.

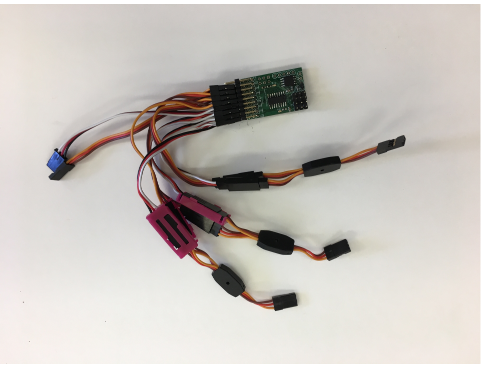


Figure 3.9: Failsafe multiplexer connection

3.2.5 Failsafe Multiplexer

To ensure safety of the vehicle during automated input excitations, a 3DR failsafe multiplexer is integrated between the autopilot and the servos to provide hardware failsafe. The multiplexer can be installed to either between the rc receiver signal and rc channels or between rc channels and auxiliary channels. The later is applied due to the number channels needed and the limited rc receiver channels. The multiplexer is a 8×4 PWM pass through with an additional port for signal switching. During a system identification flight the pilot can switch between "Automated Excite Mode" and "Manual Mode" by flipping a switch on the radio transmitter. The connection of the failsafe multiplexer is shown in Figure 3.9

4. SYSTEM IDENTIFICATION METHODS

4.1 Aircraft State-Space Modeling

Aircraft can be modeled as rigid bodies using classical mechanics as a set of first-order nonlinear ordinary differential equations Eq. (4.1):

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}) \tag{4.1}$$

In Eq. (4.1), $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^m$ is the control vector, and $\boldsymbol{f} : \mathbb{R}^+ \times \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ is a nonlinear function of the state and control. Interested readers can consult Refs. [52, 53, 54] for an in-depth discussion of flight mechanics and aircraft stability and control.

4.1.1 Aircraft Nonlinear Equations of Motion

The aircraft body-fixed frame $\mathcal{B} : \{\hat{\boldsymbol{x}}_\mathrm{b}, \hat{\boldsymbol{y}}_\mathrm{b}, \hat{\boldsymbol{z}}_\mathrm{b}\}$ is defined such that $\hat{\boldsymbol{x}}_\mathrm{b}$ points out the nose, $\hat{\boldsymbol{y}}_\mathrm{b}$ points out the right wing, and $\hat{\boldsymbol{z}}_\mathrm{b}$ points down completing a right-handed coordinate system, as seen in Figure 4.1 through the definition of the total velocity vector V_T and its components.

Following from these definitions, the *stability axis* system is constructed from the body-axis system by rotating along the angle-of-attack α such that the new $\hat{\boldsymbol{x}}_\mathrm{s}$ axis is aligned with the projection of the total velocity vector V_T along the xz plane. In this section, the aircraft equations of motion will be developed in the body axis system, with later linear models expressed in stability axes [14]. The dynamics of an aircraft can be written compactly in vector matrix form as Eq. (4.2),

$$\begin{gathered} m(\dot{\boldsymbol{V}}_\mathrm{T} + \boldsymbol{\omega} \times \boldsymbol{V}_\mathrm{T}) = m\boldsymbol{g} + \boldsymbol{F}_\mathrm{A} + \boldsymbol{F}_\mathrm{T} \\ I\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times I\boldsymbol{\omega} = \boldsymbol{\ell}_\mathrm{A} + \boldsymbol{\ell}_\mathrm{T} \end{gathered} \tag{4.2}$$

Here, $\boldsymbol{V}_\mathrm{T} = U\hat{\boldsymbol{x}}_\mathrm{b} + V\hat{\boldsymbol{y}}_\mathrm{b} + W\hat{\boldsymbol{z}}_\mathrm{b}$ is the translational velocity vector, $\boldsymbol{\omega} = P\hat{\boldsymbol{x}}_\mathrm{b} + Q\hat{\boldsymbol{y}}_\mathrm{b} + R\hat{\boldsymbol{z}}_\mathrm{b}$ is the angular velocity vector, $\boldsymbol{g}$ is the gravity vector, $\boldsymbol{F}_\mathrm{A}$ and $\boldsymbol{F}_\mathrm{T}$ are the applied forces due to aerodynamic and thrust effects, $I \in \mathbb{R}^{3\times 3}$ is the inertia tensor, and $\boldsymbol{\ell}_\mathrm{A}$ and $\boldsymbol{\ell}_\mathrm{T}$ are the applied moments due

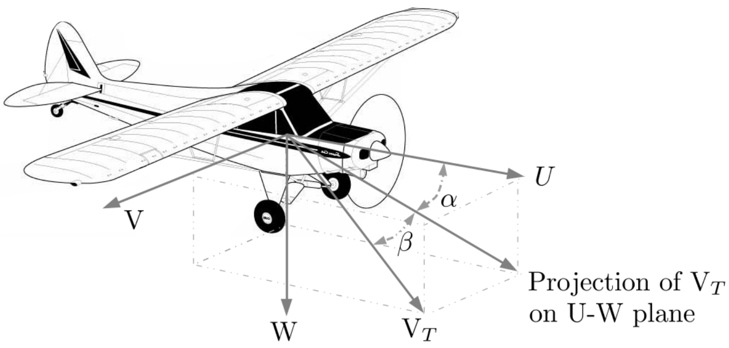


Figure 4.1: Definition of body-axis velocity components and flow angles.

to aerodynamic and thrust effects. The components of the moment vectors are denoted $\{L, M, N\}$ for rolling, pitching, and yawing moments respectively [55].

To fully specify the aircraft response a set of six kinematic equations (three translational and three rotational) are required. These equations are dependent on the choice of position-level coordinates. An inertial position vector in a North-East-Down (NED) frame and the 3-2-1 Euler angle set $\{\psi, \theta, \phi\}$ are common choices. The kinematic differential equations for the 3-2-1 Euler angles can be found in any textbook on flight mechanics such as Ref. [54], and are reproduced below:

$$\begin{aligned} \dot{\Phi} &= P + Q\sin\Phi\tan\Theta + R\cos\Phi\tan\Theta \\ \dot{\Theta} &= Q\cos\Phi - R\sin\Phi \\ \dot{\Psi} &= (Q\sin\Phi + R\cos\Phi)\sec\Theta \end{aligned} \tag{4.3}$$

From Eqs. (4.2–4.3), dropping position-level coordinates for the translational motion, the aircraft state vector is Eq. (4.4),

$$\boldsymbol{X} = [U, V, W, P, Q, R, \Psi, \Theta, \Phi]^{\mathsf{T}} \tag{4.4}$$

4.1.2 Decoupling the Equations of Motion

For a conventional aircraft at a steady, level flight condition at near-zero bank angle, the equations of motion can be decoupled into two sets: longitudinal (i.e. pitch axis) and lateral/directional (lat/d, i.e. roll and yaw axes). The longitudinal state vector consists of the variables U, W, Q, and Θ in body-axes, and the lateral/directional state vector consists of the variables V, P, R, Φ, and Ψ in body-axes. For conventional aircraft the longitudinal controls are throttle δ_{T} and elevator δ_{E}, and the lat/d controls are aileron δ_{A} and rudder δ_{R} [53]. For non-conventional aircraft the available control surfaces can be used directly or ganged together to form pseudo-control effectors [56]. For the latter approach a control allocation algorithm is usually required to obtain the actual surface deflections needed to obtain a pseudo-control command [57].

4.1.3 Linear Aircraft Models

The aircraft dynamics can be trimmed at a flight condition with states $\boldsymbol{X}_1$ and controls $\boldsymbol{U}_1$ such that the dynamics reduce to $\mathbf{0} = \boldsymbol{f}(\boldsymbol{X}_1, \boldsymbol{U}_1)$. An appropriate linearization technique can be applied to Eq. (4.1) to generate an LTI model

$$\dot{\boldsymbol{x}}(t) = A(t)\boldsymbol{x}(t) + B(t)\boldsymbol{u}(t) \tag{4.5a}$$

$$\boldsymbol{y}(t) = C(t)\boldsymbol{x}(t) + D(t)\boldsymbol{u}(t) \tag{4.5b}$$

where $\boldsymbol{x}$ and $\boldsymbol{u}$ are perturbations on the state and control respectively. The full nonlinear state is then $\boldsymbol{X} = \boldsymbol{X}_1 + \boldsymbol{x}$, and the nonlinear control is $\boldsymbol{U} = \boldsymbol{U}_1 + \boldsymbol{u}$; from these relations it is straightforward to convert between full states and perturbed states for control law implementation, plotting, and other uses [53]. In the stability axis system the parametric longitudinal linear state-space equations are:

$$\begin{Bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{Bmatrix} = \begin{bmatrix} X'_{\mathrm{u}} & X'_{\alpha} & X'_{\mathrm{q}} & -g\cos\Theta_1 \\ Z'_{\mathrm{u}} & Z'_{\alpha} & Z'_{\mathrm{q}} & -g\sin\Theta_1 \\ M'_{\mathrm{u}} & M'_{\alpha} & M'_{\mathrm{q}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{Bmatrix} u \\ \alpha \\ q \\ \theta \end{Bmatrix} + \begin{bmatrix} X'_{\delta_{\mathrm{E}}} & X'_{\delta_{\mathrm{T}}} \\ Z'_{\delta_{\mathrm{E}}} & Z'_{\delta_{\mathrm{T}}} \\ M'_{\delta_{\mathrm{E}}} & M'_{\delta_{\mathrm{T}}} \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta_{\mathrm{E}} \\ \delta_{\mathrm{T}} \end{Bmatrix} \tag{4.6}$$

The primed quantities result from decoupling the $\dot{\alpha}$ and $\dot{q}$ equations. The angle-of-attack α is synthesized from the velocity component w and the steady-state velocity U_1 as $\alpha \approx w/U_1$. This relation uses the small-angle approximation and is valid for flight conditions that are steady and level. It is therefore valid for linear models generated for these flight conditions.

The longitudinal dynamics are a fourth-order system that nominally exhibit two standard second-order modes: a high frequency, highly-damped mode exhibited mainly in body-axis pitch rate q and angle-of-attack α (*short period*) and a low frequency, lightly-damped mode exhibited in airspeed u and pitch attitude angle θ (*phugoid*). For aircraft with relaxed static stability it is common for the system to exhibit two first-order modes and a non-standard third-order mode referred to as the *third-oscillatory* mode [53].

In the stability axis system the lat/d linear state-space model is:

$$\begin{Bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{Bmatrix} = \begin{bmatrix} \dfrac{Y_\beta}{U_1} & \dfrac{Y_\mathrm{p}}{U_1} & 1 + \dfrac{Y_\mathrm{r}}{U_1} & \dfrac{g\cos\Theta_1}{U_1} \\ L'_\beta & L'_\mathrm{p} & L'_\mathrm{r} & 0 \\ N'_\beta & N'_\mathrm{p} & N'_\mathrm{r} & 0 \\ 0 & 1 & \tan\Theta_1 & 0 \end{bmatrix} \begin{Bmatrix} \beta \\ p \\ r \\ \phi \end{Bmatrix} + \begin{bmatrix} \dfrac{Y_{\delta_\mathrm{A}}}{U_1} & \dfrac{Y_{\delta_\mathrm{R}}}{U_1} \\ L'_{\delta_\mathrm{A}} & L'_{\delta_\mathrm{R}} \\ N'_{\delta_\mathrm{A}} & N'_{\delta_\mathrm{R}} \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta_\mathrm{A} \\ \delta_\mathrm{R} \end{Bmatrix} \tag{4.7}$$

The primed terms result from decoupling the $\dot{p}$ and $\dot{r}$ equations, and the sideslip angle β is obtained from the approximation $\beta \approx v/U_1$. It is valid for linear models generated for these flight conditions for the same reason stated previously. The linearized heading angle kinematics $\dot{\psi} = r$ from Eq. (4.3) are not shown in Eq. (4.7) because from an identification perspective, the relationship of $\dot{\psi}$ is known exactly and therefore should not be identified. The equations for $\dot{\theta}$ and $\dot{\phi}$ are included in Eqs. (4.6–4.7), respectively, as the other states are not independent of θ and ϕ.

The lat/d state-space model Eq. (4.7) forms a fourth-order system with two standard first-order modes and a standard second-order mode for most conventional aircraft configurations. The *roll* mode is a first-order mode, which, as the name implies, is primarily composed of the aircraft body-axis roll rate p. The other first-order mode is referred to as *spiral* and is typically a very slow mode primarily composed of roll (ϕ) and heading angle (ψ). The second-order mode is known as the *Dutch roll* and is an oscillatory motion exhibited mostly in body-axis yaw rate (r), sideslip angle (β), and body-axis roll rate. It is a mode that often needs improved damping for acceptable flying qualities [54].

Equations (4.6–4.7) are the basic models that are desired to be identified from the techniques presented in this work, and subsequently used to determine modal characteristics. The output equation Eq. (4.5b) is not identified, as 1) full state measurements are available and 2) models are assumed to be strictly proper (i.e. $D = 0$) so the system input-output relationship is fully encoded in the state equation. The user can choose appropriate C and D matrices after identification to represent the measurements obtained from the system after the flight test instrumentation is removed.

4.2 Observer/Kalman Filter Identification

This research utilizes OKID as the system identification method due to its ability to identify systems simply from input/output data without additional ground measurements. A modified version of OKID is developed in Python and a batch identification method is implemented in order to improve the reliability of the identified model.

During the 1980's many system identification methods have been developed to identify linear state-space model for spacecraft and aircraft with flexible structural characteristics. The majority of these methods are based on Fast Fourier Transform(FFT), Maximum Likelihood Estimation, and least squares [58]. A drawback of the FFT and MLE methods is that a somewhat rich input is required to prevent ill-conditioned computation. Developed in the 1990's by Juang [20], the Observer/Kalman Filter Identification algorithm (OKID) method is a direct Kalman filter gain approach. OKID is formulated in time-domain and is capable of handling general response data. This is especially valuable for aircraft modeling since pure impulse excitations are difficult to apply and noise/ signal ratio of sensing data are usually high. It also has the benefit of allowing for nonzero initial conditions and does not require the data to be collected until steady state. The present work is an extension of the concept that OKID can be successfully used to identify state-space models of flight vehicles.

The basic formulation of the OKID algorithm begins with the linearized, discrete-time, state-space equations augmented with an observer gain:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \bar{A}\boldsymbol{x}(k) + \bar{B}\boldsymbol{v}(k) \\ \boldsymbol{y}(k) &= C\boldsymbol{x}(k) + D\boldsymbol{u}(k) \end{aligned} \tag{4.8}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^n$, $\boldsymbol{y}(k) \in \mathbb{R}^m$, $\boldsymbol{u}(k) \in \mathbb{R}^r$, are state, output and control inputs with dimension of n, m, and r respectively with

$$\begin{aligned} \bar{A} &= A + GC \\ \bar{B} &= [B + GD, -G] \\ \boldsymbol{v}(k) &= \begin{bmatrix} \boldsymbol{u}(k) \\ \boldsymbol{y}(k) \end{bmatrix} \end{aligned} \tag{4.9}$$

and $G \in \mathbb{R}^{n \times m}$ is an arbitrary matrix chosen to make the matrix $\bar{A}$ stable. Assuming zero initial conditions, $\boldsymbol{x} = 0$ and integer p satisfying $CA^kB \approx 0$ for $k \geq p$. By substituting and iterating through each time step using Equation (4.8), the Observer Markov Parameters (OMP) comprised of a input-output relationship becomes

$$\bar{\boldsymbol{y}} = C\bar{A}^p\boldsymbol{x} + \bar{Y}\bar{V} \tag{4.10}$$

where

$$
\begin{aligned}
\bar{y} &= \begin{bmatrix} \boldsymbol{y}(p) & \boldsymbol{y}(p+1) & \cdots & \boldsymbol{y}(l-1) \end{bmatrix} \\
\bar{Y} &= \begin{bmatrix} D & C\bar{B} & C\bar{A}\bar{B} & \cdots & C\bar{A}^{(p-1)}\bar{B} \end{bmatrix} \\
\bar{V} &= \begin{bmatrix} \boldsymbol{u}(p) & \boldsymbol{u}(p+1) & \cdots & \boldsymbol{u}(l-1) \\ \boldsymbol{v}(p-1) & \boldsymbol{v}(p) & \cdots & \boldsymbol{v}(l-2) \\ \boldsymbol{v}(p-2) & \boldsymbol{v}(p-1) & \cdots & \boldsymbol{v}(l-3) \\ \vdots & \ddots & \cdots & \vdots \\ \boldsymbol{v}(0) & \boldsymbol{v}(1) & \cdots & \boldsymbol{v}(l-p-1) \end{bmatrix}
\end{aligned}
\tag{4.11}
$$

Since the system Markov parameters are what are really important, the matrix $\bar{Y}$ is partitioned such that

$$
\begin{aligned}
\bar{Y} &= \begin{bmatrix} D \; C\bar{B} \; C\bar{A}\bar{B} \; \cdots \; C\bar{A}^{(p-1)}\bar{B} \end{bmatrix} \\
&= \begin{bmatrix} Y_0 \; Y_1 \; Y_2 \; \cdots Y_p \end{bmatrix}
\end{aligned}
\tag{4.12}
$$

from which the OMP are obtained.

$$
\begin{aligned}
\bar{Y}_0 &= D \\
\bar{Y}_k &= C\bar{A}^{(k-1)}\bar{B} \\
&= \begin{bmatrix} C(A+GC)^{(k-1)}(B+GD) & -C(A+GC)^{(k-1)}G \end{bmatrix} \\
&= \begin{bmatrix} \bar{Y}_k^{(1)} & -\bar{Y}_k^{(2)} \end{bmatrix} \qquad k = 1, 2, 3, \ldots.
\end{aligned}
\tag{4.13}
$$

The general relationship between the actual System Markov Parameters and the OMP can be shown to be:

$$\begin{aligned}
D &= Y_0 = \bar{Y}_0 \\
Y_k &= \bar{Y}_k^{(1)} - \sum_{i=1}^{k} \bar{Y}_i^{(2)} Y_{(k-i)} \quad \text{for } k = 1, \ldots, p \\
Y_k &= -\sum_{i=1}^{p} \bar{Y}_i^{(2)} Y_{(k-i)} \quad \text{for } k = p+1, \ldots, \infty
\end{aligned} \tag{4.14}$$

The next step is to use singular value decomposition (SVD) on the Hankel matrix.

$$\begin{aligned}
H(k-1) &= \begin{bmatrix} Y_k & Y_{k+1} & \cdots & Y_{k+\beta-1} \\ Y_{k+1} & Y_{k+2} & \cdots & Y_{k+\beta} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{k+\alpha-1} & Y_{k+\alpha} & \cdots & Y_{k+\alpha+\beta-2} \end{bmatrix} \\
H(0) &= P_n \Sigma Q_n^{\mathsf{T}}
\end{aligned} \tag{4.15}$$

The Eigensystem Realization Algorithm is then used to solve the Hankel matrix for the desired state-space realization (A, B, C, D):

$$\begin{aligned}
\hat{A} &= \Sigma_n^{-1/2} P_n^{\mathsf{T}} H(1) Q_n \Sigma_n^{-1/2} \\
\hat{B} &= \Sigma_n^{1/2} Q_n^{\mathsf{T}} \\
\hat{C} &= P_n \Sigma_n^{1/2} \\
\hat{D} &= Y_0
\end{aligned} \tag{4.16}$$

Note that $\hat{A}$, $\hat{B}$, and $\hat{C}$ are the estimated system matrices arrived at by the system identification using OKID. The $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ represent the identified discrete linear state-space system when used in Eq. (4.17).

$$\begin{aligned}
\boldsymbol{x}(k+1) &= \hat{A}\boldsymbol{x}(k) + \hat{B}\boldsymbol{u}(k) \\
\boldsymbol{y}(k) &= \hat{C}\boldsymbol{x}(k) + \hat{D}\boldsymbol{u}(k)
\end{aligned} \tag{4.17}$$

Additionally, the observer gain Markov sequence can be computed through the following sequence

$$\begin{cases} Y_1^o = C\bar{G} = \bar{Y}_1^{(2)} \\ Y_k^o = CA^{k-1}\bar{G} = \bar{Y}_k^{(2)} - \sum_{i=1}^{k-1} \bar{Y}_i^{(2)} Y_{k-i}^o \quad k = 2, 3, \cdots, p \\ Y_k^o = CA^{k-1}\bar{G} = -\sum_{i=1}^{p} \bar{Y}_i^{(2)} Y_{k-i}^o \quad k > p \end{cases} \tag{4.18}$$

which yields

$$\bar{G}_k = \left[CA^{k-1}\right]^{\dagger} Y_k^o \tag{4.19}$$

4.3 Frequency Domain Analysis

System identification can be performed in both time and frequency domain on dynamic systems. For parameter estimation, methods such as CIFER or SIDPAC both include frequency domain analysis due to the direct applicability to control design, the physical insight of system behavior, and the computational efficiency comparing with most time domain methods. Time history data can be transformed into frequency domain through Direct Fourier Transformation (DFT), which has the form of

$$F_n = \sum_{k=0}^{l-1} f_k \, e^{-2\pi i n k/N} \quad k = 0, 1, \cdots, l-1 \tag{4.20}$$

where $f_k = f(d_t \cdot k)$. Despite being capable of transforming the time domain data to frequency domain to calculate the Frequency Response Function (FRF), using Equation (4.20) directly is computationally inefficient. Fast Fourier Transformation (FFT) instead is an much faster method of producing the same results as the direct DFT calculation. The number of calculations required reduces from O(N^2) to O($N \log N$) and therefore is widely used in digital signal processing.

To calculate the FRF of a specific input-output pair, correlation functions are required

$$\begin{aligned} R_{uu} &= \frac{1}{l}\sum_{\tau=0}^{l-1} u(\tau)u^T(\tau-\nu) \\ R_{yu} &= \frac{1}{l}\sum_{\tau=0}^{l-1} y(\tau)u^T(\tau-\nu) \\ R_{yy} &= \frac{1}{l}\sum_{\tau=0}^{l-1} y(\tau)y^T(\tau-\nu) \end{aligned} \tag{4.21}$$

in which τ is the time index and ν is the time shift index. The averaged auto and cross spectral densities are formed through

$$\begin{aligned} \bar{S}_{uu} &= \frac{1}{n_s}\sum_{i=0}^{n_s} R_{uu_i}R^*_{uu_i} \\ \bar{S}_{yu} &= \frac{1}{n_s}\sum_{i=0}^{n_s} R_{yu_i}R^*_{yu_i} \\ \bar{S}_{yy} &= \frac{1}{n_s}\sum_{i=0}^{n_s} R_{yy_i}R^*_{yy_i} \end{aligned} \tag{4.22}$$

where n_s is the number of segments for averaging, and $*$ symbols the conjugate of the function. By averaging all the segments, the uniformly distributed spectral density function over all frequency points is acquired. Finally the averaged FRF

$$G = \frac{\bar{S}_{yu}}{\bar{S}_{uu}} \tag{4.23}$$

Note that the inputs are required to be periodic in order to avoid issues related to data leakage. More about data leakage is discussed in the Appendix A.1.

4.3.1 Frequency Domain Observer Kalman Filter Identification

Similar to time domain OKID derived in Section 4.2, the first step to perform OKID is to calculate the SMPs from the input-output signals for the Hankel matrices. The output response of the discrete time model can be written as a combination of the transient and forced response comprised of SMPs

$$y(k) = CA^k x(0) + \sum_{i=1}^{\infty} CA^{k-i} Bu(i-1) + Du(k) \tag{4.24}$$

by performing the DFT calculation on Equation (4.20), the FRF can also be expressed in the form of

$$G = \sum_{k=0}^{\infty} Y_k e^{-j\omega k d_t} \tag{4.25}$$

Therefore, taking the inverse transform of FRF acquired from Equation (4.23) yields the SMP of the system

$$Y_\tau = \frac{1}{l} \sum_{k=0}^{l-1} G(k) e^{j\frac{2\pi k}{l}\tau} \tag{4.26}$$

where $Y_0 = D$ and $Y_\tau = CA^{\tau-1}B$ are the system markov parameters. Following the same steps from Equation (4.15) and Equation (4.16) yields the identified system matrices.

4.4 Data Collection and Input Maneuver Design

System identification is the process of capturing the dynamic behavior of the vehicle embodied in measured data. The results are dominated by the measured data quality and the input maneuver design. Poorly designed or executed maneuvers lack the amplitude or frequency range and results in inputs that are incapable of exciting the specified dynamic modes. Data sampling rate is also important and needs to be addressed in order to reduce data aliasing. Aliasing occurs when the sampling rate is lower than the system Nyquest frequency, and the issue is caused by analog and digital data conversion. The result of aliasing is a false translation of power outside of the original frequency range due to distortion, which leads to poor identification models. Therefore, with advanced digital computers being used due to the high computational power, sampling rate must be above the minimal threshold to avoid signal aliasing.

4.4.1 Sampling Rate

The decision of the sampling rate of a system is driven by the Nyquist-Shannon sampling theorem. With a time interval between sample as T_s, the sampling rate

$$f = 1/T_s \tag{4.27}$$

is expressed in samples/second or Hz. The theorem establishes a sufficient condition for a sample rate that allows discrete samples of a continuous system at a frequency f to capture frequency content up to the Nyquist frequency f_N,

$$f_N = f/2 \tag{4.28}$$

However, the theoretical minimum sampling rate merely provides the lowest rate to avoid aliasing. In practice, a much higher sampling rate is necessary for good identification results. The known rule of thumb for an ideal sampling rate f is

$$f = 25 f_{max} \tag{4.29}$$

where f_{max} is the maximum frequency of interest. From previous flight tests results we have seen that most UAS vehicle rigid body modes does not exceed 1.5 HZ. Accordingly, the maximum sampling rate for DFTI is set to 100 Hz and is normally operated at 50 Hz.

For data acquisition systems acquiring data from multiple sensors, it is common that different sample speed is used for individual sensors. To unify the data sample rate two methods can be used. The first method is to interpolate the data acquired from the sensor with a lower sampling rate to a higher rate. A linear or polynomial interpolation works well for most signals. The second method is to sample the high sample rate down to the lowest sample rate. The later method was used for the DFTI system. For DFTI, the μADC has a maximum sensing sampling rate of 100 Hz while other sensor have higher sampling rates. Therefore, the whole DFTI system was set to a sampling frequency of the lowest sensor sample rate. Note that for both methods the lowest sampling rate still needs to comply with the Nyquist-Shannon theorem to avoid aliasing.

4.4.2 Data Smoothing

Data smoothing is a common procedure for system identification. By using a certain number of data points before and after the desired data, a smoothed data can be acquired. Data smoothing can be done in both time and frequency domain, and can also be used to interpolate data from different sensor with dissimilar sampling rate. Frequency domain smoothing applies Fourier analysis with a goal to separate signal and measurement noise. Detailed development of smoothing using optimal Fourier smoothing can be found in [59]. The smoothing method primary used in this thesis is a local time domain method. The idea is to fit a polynomial to the measured signal while maintaining data integrity. Common time domain methods include moving average method, local regression methods, and the Savitzky-Golay smoothing, and differentiation filter. Moving average method has the advantage of a low computational cost but suffers from lagging issues, and thus, to avoid signal distortion the moving window should also be assigned carefully. Regression methods and Savitzky-Golay smoothing on the other hand has the advantage of preserving the area and position of the peaks, such traits are important to be preserved during data smoothing for system identification. Therefore, to enhance consistency on signal smoothing, regression methods are more popular for data smoothing.

4.4.3 Input Maneuvers

Several flight test maneuvers have been discussed in [58, 60, 61] for the purpose of performing efficient excitation for system identification. The excitation methods in this thesis are commanded by the input signal generator function, sent through Clark to the Pixhawk2 auxiliary ports for execution.

Two general approaches are applied in designing inputs for system identification. The first approach requires no *a priori* knowledge of the behavior of the dynamical system. Inputs such as impulses and frequency sweeps fall in this category, where the goal is to excite all of the dy-

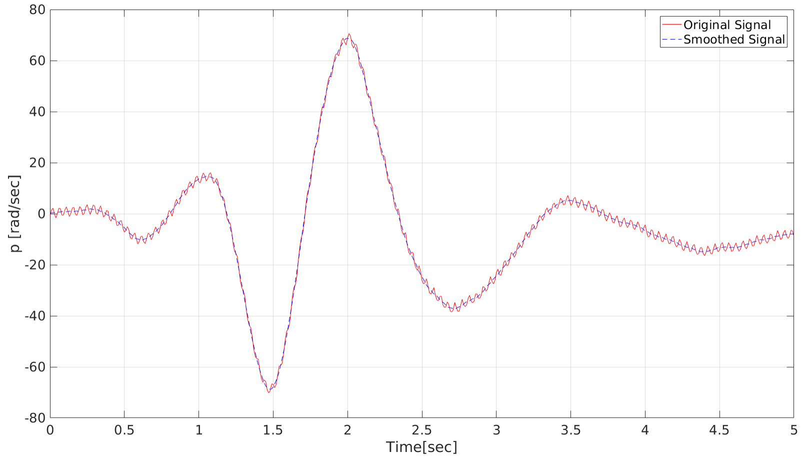


Figure 4.2: Time series comparison of smoothed and original measured data

namic modes in a large range of frequency. The second approach requires *a priori* understanding of the system and designs the input with respect to the dynamic modes. Square waves are used to construct this particular kind of excitation. To excite the dynamic modes without flexible structural modes the maneuver time length, control surface magnitude, maneuver sequence, and input correlations are specified [14]. The latter approach is used for SUAS flight testing due to the *a priori* knowledge of the natural frequencies of the structure, and doublets are used for excitation to reduce the excitation error induced by the pilot. Multiple control inputs must be coordinated to maximize data content and to ensure the responses fall between the limitations for model structure validity. In addition, it is known that in order to model the effect of a particular control surface, the surface has to have moving action. Take lateral/directional roll mode for example, the input maneuver sequence must include a aileron deflection

Input amplitudes must also be correctly addressed in order to provide quality dynamic mode excitation. Small input maneuvers may result in a low signal-to-noise ratio, but on the other hand an input too large will drive the dynamic model out of the linear range, which is a common model structure assumption for most system identification methods.

4.4.4 Piloted Maneuvers

To successfully excite the dynamic modes of the aircraft, input maneuvers are carefully designed to achieve maximum success rate. Traditionally, excitation maneuvers has been executed by trained pilot. Figure 4.3 gives an input excitation sequence example from flight test used for lat/d system identification. The excitation frequency is chosen with regard to the expected frequency of the dynamic modes; the input magnitude is selected in order to ensure the vehicle responds in the linear range.

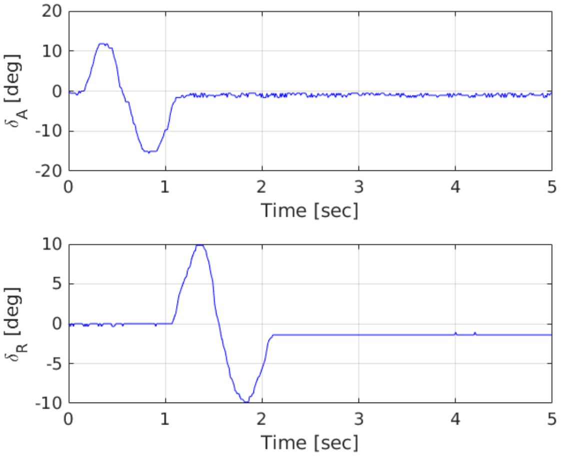


Figure 4.3: Coordinated input excitation to perturb lateral/directional modes.

From empirical experience, input excitations are performed using two separate test sequences: one for longitudinal modes and the other for lateral/directional modes. Using this method to get the most precise data, all maneuvers are performed on a 1-2-3 type count. For example, an aileron doublet is performed as a two count (or 2 second) maneuver: one count of left aileron followed by one count of right aileron. The most general doublet excitation sequences for both longitudinal and lateral/directional modes are explained in the following paragraph.

4.4.4.1 *General Longitudinal Input Sequences And Modes*

Longitudinal modes on aircrafts are hard to excite due to the nature of a mixing of fast and slow dynamics. The short period mode is the fast mode that is highly correlated to the elevator deflection while the phugoid mode is part of the slow dynamics induced by the throttle input. Both modes are second order modes but with different damping ratio. For the best response from the aircraft the following sequence is applied: down elevator, up elevator, full throttle, and finally cut throttle. This is performed as a five count maneuver: one count down elevator, one count up elevator, two counts of full throttle, and one count cut throttle. Using this excitation sequence consistently results in three periods of the phugoid mode under stick free conditions. Figure 4.4 is an actual flight test longitudinal input example that produced good identification result using OKID.

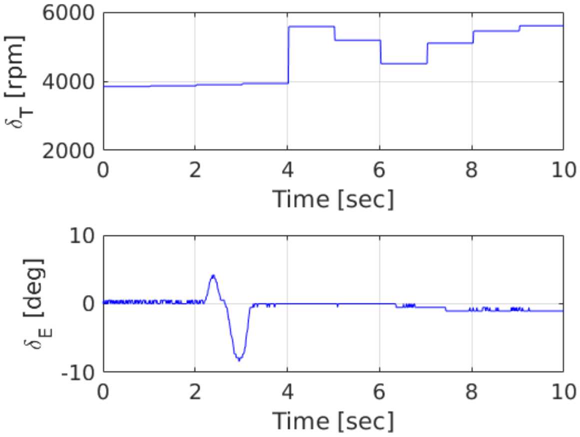


Figure 4.4: Coordinated input excitation to perturb longitudinal modes.

4.4.4.2 *General Lateral/Directional Input Sequences And Modes*

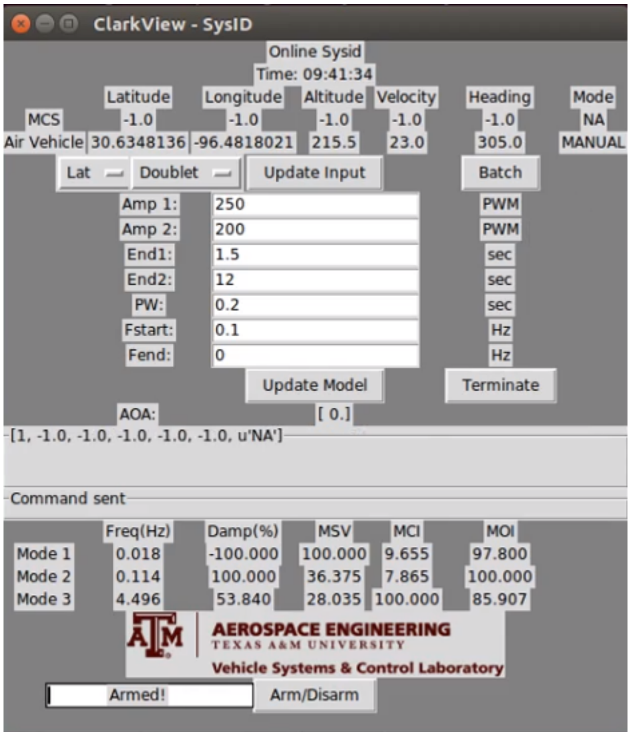
Two first order modes and one second order mode is expected for a complete lateral/directional excitation. The first order spiral mode is perturbed using a four count maneuver: right rudder, left rudder, right aileron, and left aileron. This sequence allows for at least one complete spiral revolution. The Dutch Roll mode is perturbed using a four count maneuver: right aileron, left aileron, right rudder, and left rudder. The modification of applying the rudder doublet at the end of the input sequence allows for a visible Dutch roll mode and then a complete spiral revolution. Figure shows an asynchronous aileron doublet followed by a rudder doublet.

4.4.5 Automated excitation

Modeling cost can be reduced by improving the success rate of generating dynamic models. As mentioned previously in Chapter 1, modeling efficiency can be affected by both atmospheric disturbances and human-induced command errors. While environmental disturbance can be reduced by flight testing at a calm day with minimal wind gust, excitation consistency is still a major issue that needs to be addressed. That is why an automated mode excitation method was developed. The method aims to target two issues with current human piloted inputs. One is to eliminate imprecise pilot inputs that causes unsuccessful mode excitation, and the other is to increase the repeatability of the input to enable recursive model building.

System identification input design methods can be categorized into two general approaches. The first approach requires no *a priori* knowledge of the behavior of the dynamical system. Inputs such as impulses and frequency sweeps fall in this category, where the goal is to excite all the dynamic modes in a large range of frequency. The second approach requires *a priori* understanding of the system and design the input with respect to the dynamic modes. Square waves are used to construct this particular kind of excitation. Previous flight test results have shown that the consistency and quality of piloted excitation relies heavily on training and experience [62]. Human piloted maneuvers are limited to the transmitter sensitivity and human operating rate, square waves are mostly applied due to the ease of execution. However, even with experienced pilots, the process of executing the exact assigned excitation frequency required is difficult and poor input executions may excite flexible structural modes that can cause vehicle fluttering. Therefore, an automated excitation with user assigned maneuver time length, control surface magnitude, maneuver sequence, and input correlations is needed.

The human-in-the-loop automated excitation is developed using the Clark communication system described in Section 2.2.2. All decisions are transferred through XBEEs utilizing the zigbee



Online Sysid
Time: 09:41:34

	Latitude	Longitude	Altitude	Velocity	Heading	Mode
MCS	-1.0	-1.0	-1.0	-1.0	-1.0	NA
Air Vehicle	30.6348136	-96.4818021	215.5	23.0	305.0	MANUAL

Lat
Doublet

Amp 1: 250 PWM
Amp 2: 200 PWM
End1: 1.5 sec
End2: 12 sec
PW: 0.2 sec
Fstart: 0.1 Hz
Fend: 0 Hz

AOA: [0.]

[1, -1.0, -1.0, -1.0, -1.0, -1.0, u'NA']

Command sent

	Freq(Hz)	Damp(%)	MSV	MCI	MOI
Mode 1	0.018	-100.000	100.000	9.655	97.800
Mode 2	0.114	100.000	36.375	7.865	100.000
Mode 3	4.496	53.840	28.035	100.000	85.907

Armed!

Figure 4.5: Graphical interface for input modifications

protocol. Figure 4.5 shows the operational GUI for interfacing between the GCS and the air vehicle. Several input excitation methods are implemented in the input signal generator function that are executed through the automated excitation software. Each input can be tailored for the selected servos, and additional in-flight modification choices are provided. Besides the standard doublet inputs, inputs such as the 3-2-1-1 input, sinusoid inputs, and multi-sine inputs can be applied though the automated excitation system.

4.4.5.1 3-2-1-1

One of the common input excitation applied on both manned and unmanned fixed wing aircraft is the 3-2-1-1 input. Natural frequency and damping ratio can be accurately determined when the input contains frequency in the range of where the expected natural frequencies are located. The 3-2-1-1 is a combination of three consecutive pulses. The input pulses are designed so that the second pulse signal corresponds to the dominant dynamic mode of interest, and the first and third excitation broadens the excited frequency range. Note that although this method can yield great identification results, it should also be used with caution and not applied on certain control surfaces. For example, an input not properly designed in amplitude and duration on the aileron will drive the aircraft away from the trimmed flight condition resulting in a hard left/right bank, and similarly, same situation on the elevator may perturb the vehicle to be close to stall condition. Figure 4.6 is an example of a comparison between the assigned and measured 3-2-1-1 input.

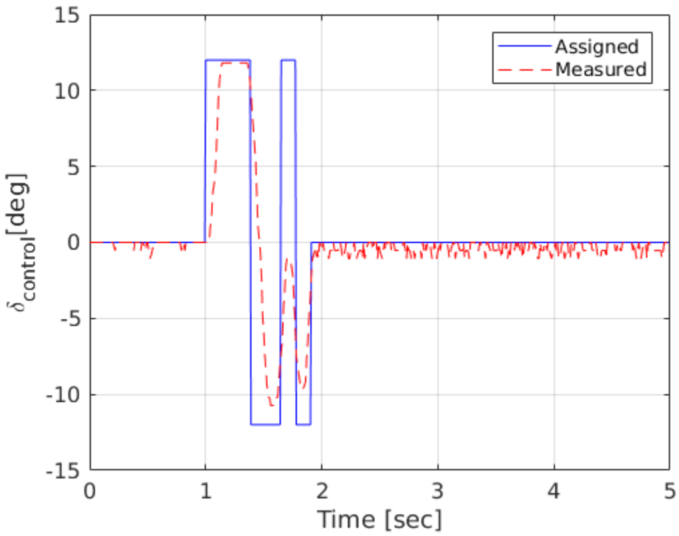


Figure 4.6: Measured and assigned 3-2-1-1 input time histories

It can be seen that although the 3-2-1-1 excitation provides a wider range of excitation frequency comparing to normal doublets, the excitations are limited by the servo response time delay and often cannot achieve the desired full excitation magnitude. Although for the group of SUAS of interest, the dynamics of the aircraft does not exceed this frequency, the unbalanced input issue does lead to the vehicle deviating from trim condition.

4.4.5.2 Sinusoid Frequency Sweeps

Sinusoid sweeps have been tested in flight tests manually and have provided promising results for offline system identification with piloted excitations. The limitation with piloted sinusoid frequency sweeps are that the transmitter as well as the human pilot can only provide excitations up to a certain frequency, as a result leaving certain parts of the power spectrum not excited. As shown in Figure 4.7, both the top and the bottom figures are displaying the measured elevator deflection angle recorded by DFTI though potentiometers. The signal in the top figure was assigned a duration of 1.5 seconds with a assigned amplitude of 180 PWM, resulting an oscillation of ± 4 degree deflection around the elevator trim point $\delta_{E_{trim}}$, a starting frequency of $f_{start} = 1$ Hz, and ending frequency of $f_{end} = 10$ Hz. The automated sweep signal successfully performed the designated excitation and returned to the trim point after the excitation ended. On the other hand, the bottom figure shows the manual sweep signal performed by a pilot. It can be seen that the sweep signal are a lot less precise and uneven around the trim point, and additionally, the exact input excite frequency is unknown.

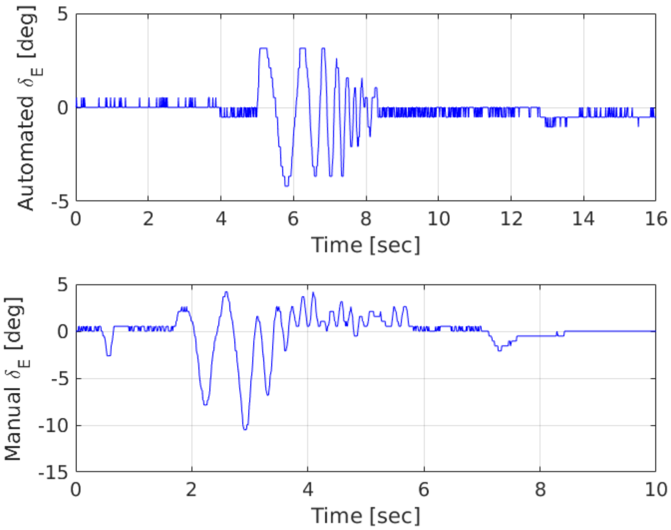


Figure 4.7: Comparison of measured manual and automated sinusoid frequency sweep

Similar to the 3-2-1-1 excitation, it can be seen from the top figure of Figure 4.7 that after 7 sec the excitation amplitude reduces due to the servo excitation limits. Therefore, besides being capable of providing a reliable and balanced input that excites the vehicle around the trimmed point, the servo limits should be taken into account to maintain the excitation amplitude.

4.4.5.3 Multi-sine

Multi-sine input is an excitation signal that consists of multiple sinusoid signals with a phase or frequency shift. Schroeder [63] have previously shown that by adjusting the phase angles of a periodic signal with a given power spectrum the peak-to-peak amplitude can be minimized. It has the advantage of tailoring the input signal to the desired excitation frequencies. Multi-sine inputs eliminates unnecessary high frequency inputs while maintaining the vehicle very close to the trim point. For a periodic input signal $u(t)$ with period T and finite bandwidth, its Fourier series is

$$u(t) = \sum_{k=1}^{M} A_k \, \cos(\frac{2\pi k \tau(t)}{T} + \phi_k), \;\; t = 0, 1, \cdots, N-1 \tag{4.30}$$

where M is the number of harmonically related frequencies and ϕ_k are the phase angles to be harmonic components to produce low Peak Factor (PF). Morelli [26] introduced the idea of using PF and Relative Peak Factor (RPF) to measure the efficiency of an input for parameter excitation purposes. Peak factor is defined by

$$PF(\boldsymbol{u}) = \frac{\frac{\max(\boldsymbol{u})-\min(\boldsymbol{u})}{2}}{\sqrt{\frac{\boldsymbol{u}^T\boldsymbol{u}}{N}}} \tag{4.31}$$

or

$$PF(\boldsymbol{u}) = \frac{\frac{\max(\boldsymbol{u})-\min(\boldsymbol{u})}{2}}{rms(\boldsymbol{u})} = \frac{\|\boldsymbol{u}_\infty\|}{\|\boldsymbol{u}_2\|} \tag{4.32}$$

where $\boldsymbol{u} = [u_0, u_1, \cdots, u_{N-1}]$ and $\|\boldsymbol{u}_\infty\| \,/\, \|\boldsymbol{u}_2\|$ is a crest factor. Crest factor is the ratio of the peak value to the rms value of the signal. Similarly, the RPF is defined by

$$RPF(\boldsymbol{u}) = \frac{\frac{\max(\boldsymbol{u})-\min(\boldsymbol{u})}{2}}{2\sqrt{2}\, rms(\boldsymbol{u})} = \frac{PF(\boldsymbol{u})}{\sqrt{2}} \tag{4.33}$$

The RPF quantifies the peak factor of $\boldsymbol{u}$ relative to the peak factor of a single sinusoid wave, which has a peak factor of $\sqrt{2}$. Low RPFs indicates that the signal is capable of exciting the system with a variety of frequencies without driving the aircraft too far away from the trim point.

To perform an uniform power spectrum applying Schroeder sweep, Equation (4.30) will have the following characteristics,

$$\begin{aligned} A_k &= \sqrt{P/M} \\ \phi_1 &= 0 \\ \phi_k &= \phi_{k-1} - \frac{\pi k^2}{N} \;\; k = 2, 3, \cdots, M \end{aligned} \tag{4.34}$$

in which P is the total desired input power.

By applying the Schroeder sweep, it enables an experimental input design to synchronously excite multiple control surfaces in a short period with the benefit of having mutually orthogonal inputs that are not correlated to each other.

Table 4.1: Comparison of different inputs

Method	**Assigned Frequency**	**Balanced Excitation**	**Pros**	**Cons**
Doublets	Yes	Yes	Easy for piloted maneuver	Low success rate
3-2-1-1	Yes	No	Wider range of frequency excited	Deviate trimmed point
Sinusoid sweep	No	Yes	Widest range of excitation frequencies	May run into structural modes Possible coupling between inputs
Multi-sine sweep	Yes	Yes	Avoids structural modes; No coupling between different control surfaces	Optional basics knowledge on vehicle dynamic frequencies

5.1 Online System Identification Procedure

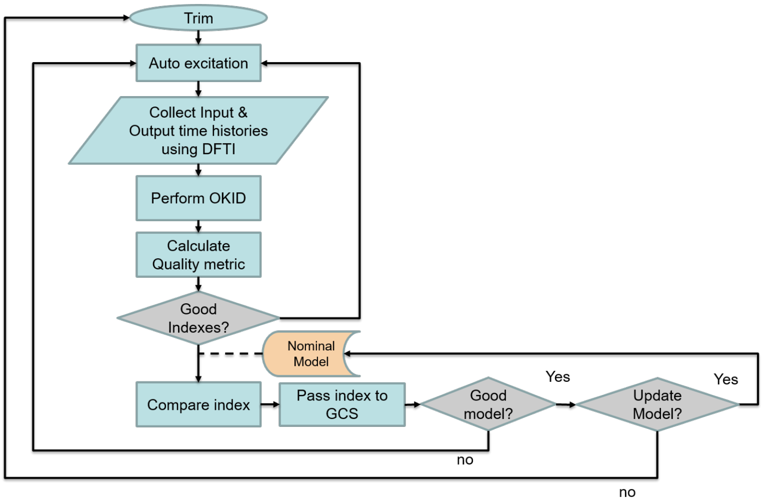


Figure 5.1: Online system identification procedure

The structure of the procedure consist of an identification loop in the onboard computer and an update loop involving interactions with the GCS operator. A UAS pilot is required for aircraft trimming, excitation initiating, as well as making sure the aircraft is in a safe condition. The human-in-the-loop model update procedure is initiated when the test pilot flips the excitation switch on the transmitter and ends when the ground control operator makes the decision for model update. Data is being transfered from the air vehicle to the GCS through XBees and communication between each sub-processors follows the UDP/IP protocol. The update procedure is shown in Fig. 5.1 for better understanding.

Once initiated, DFTI starts recording a log that records the full duration of the flight for offline data comparison. After the pilot manually trimmed the aircraft, the ground control operator updates the excitation parameters through the GUI to update the stored excitation parameters on the vehicle. Once the updated values are received, the system reads and saves the trimmed servo values before excitation. The excitation is initiated only after the pilot flips the excitation switch on the transmitter to start the automated-excitation loop. Servo commands are then sent to Pixhawk via Mavlink and the system identification interface records a separate data log for each excitation, which we consider as the current data log. The current data log records the current input excitation signal, excitation states, and the result of the identification. Results are transferred down to the GCS though Clark showing identification quality indicators introduced in Section **??** along with additional information of the difference of the current identified model and the known nominal model. Finally, the ground control operator makes the decision of model update. If the ground control operator decides to update the model, the information in the current data log will replace the log for the nominal model. Note that a nominal model can be preloaded but is not required for the update procedure. Moreover, instead of identifying each input signal, the identification process can also support batch identification with a higher computational time.

A more detailed software pseudo code for the identification process is shown in Algorithm 1. Where f_g is a set of initializing parameters for the identification process, x_t is the confidence level for the identified model, *DoExcite* is a boolean parameter that is set to **TRUE** when the Ground Control Operator (GCO) assigns a excitation command, q_{index} is a quality indexes calculated for each iteration, and $\{\hat{A}, \hat{B}, \hat{C}, \hat{D}\}$ is a set of identified system matrices. Note that the detailed method used for quantifying the integrity of newly identified model will be presented in Section 5.3.

5.2 Model Reduction

Noise and structural nonlinearity both introduces uncertainty on the rank of the Hankel Matrix in Equation (4.15) and alters the dimension of the realization matrix. In order to identify the real dynamic modes, quality indicators are used to provide information on individual mode participation. Spurious or computational modes have a low temporal and spacial consistency, and therefore, by truncating the modes below the assigned threshold yields the identified model with the proper rank. Several quality indicators have shown to be reliable for mode selection [64, 65], in this

Algorithm 1 Online System Identification

```
procedure MODEL UPDATING ALGORITHM
    Initialize parameters $f_g$ for identification        ▷ Stable/Unstable modes
    Set desired system size
    Initialize excitation characteristics
    Set update decision $U_d$ to FALSE, Confidence level $x_t == 0$
    if $DoExcite$ == TRUE then
        Perform excitation of GCO's choice
    Pass acquired data from DAQ through UDP
loop:
    while $xt \leqslant 0.5$ do        ▷ Confidence level is 1 when 100 % confident
        for $i = 0; \quad i++$ do
            Run OKID with $i$th input and state measurements
            Append $i$th data set to previous total data set
            Calculate quality indexes $q_{index}(i)$
    return $q_{index}(i)$, $\{\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i\}$
update:
    Send $q_{index}$ to GCS through CLARK, update $U_d$
    if $U_d$ ==TRUE and $q_{index}(i) >$ nominal quality index $q_n$ then
        Update nominal model $\{\hat{A}_N, \hat{B}_N, \hat{C}_N, \hat{D}_N\} \leftarrow \{\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i\}$.
        goto loop.
    else
        Remove $\{\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i\}$
        goto loop.
```

section, the methods used for both online and offline modal selection is introduced and discussed.

In order to analyze the individual identified dynamic mode, the system matrices from Equation (4.16) is transformed to modal coordinates via coordinate transformation. With the eigenvalue matrix Λ and eigenvector matrix Ψ, the system matrices in modal coordinates are

$$\begin{aligned}\Lambda &= \Psi^{-1}A\Psi \\ B_m &= \Psi^{-1}B \\ C_m &= C\Psi\end{aligned} \tag{5.1}$$

where $\Lambda \in \mathbb{R}^{nm\times nm}$ is a matrix with eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ as the diagonal terms, $B_m \in \mathbb{R}^{nm\times r}$ is the modal participation factor, nm is the number of modes, and $C_m \in \mathbb{R}^{m\times nm}$ is the mode shape. With system matrices in modal coordinates, Mode Singular Values (MSV), Modal Controllability Index (MCI), and Modal Observability Index (MOI) can be calculated via

$$\begin{aligned}MCI &= 100\cdot\ |B_{\mathrm{m}}|\max|B_{\mathrm{m}}| \\ MOI &= 100\cdot\ |C_{\mathrm{m}}|\max|C_{\mathrm{m}}| \\ MSV &= 100\cdot\frac{\dfrac{\sqrt{|B_{\mathrm{m}}|\cdot|C_{\mathrm{m}}|}}{|1-|\lambda||}}{\max\dfrac{\sqrt{|B_{\mathrm{m}}|\cdot|C_{\mathrm{m}}|}}{|1-|\lambda||}}\end{aligned} \tag{5.2}$$

With $Y_{mi} = \begin{bmatrix} c_{mi} & c_{mi}\lambda_i b_{mi} & \cdots & c_{mi}{\lambda_i}^{l-1}b_{mi}\end{bmatrix}$ and

$$B_m = \begin{bmatrix} b_{m1} \\ b_{m2} \\ \vdots \\ b_{mi}\end{bmatrix} \quad C_m = \begin{bmatrix} c_{m1} & c_{m2} & \cdots & c_{mi}\end{bmatrix} \tag{5.3}$$

for $i = 1, \cdots, n$. Also, for MSV the following relationship must be established

$$\|Y_{mi}\| \leq \sqrt{\|c_{mi}\| \left(1 + |\lambda_i| + \left|\lambda_i^{\,2}\right| + \cdots + \left|\lambda_i^{\,l-1}\right|\right) \|b_{mi}\|} \tag{5.4}$$

These indicators have the advantage of requiring simple computational steps while providing valuable modal information. MSV is an indication of the contribution of each mode to the identified model pulse response history, and therefore, is the primary identification quality indicator that is calculated on-board and transmitted to the GCS to be used to determine the identification success for online system identification.

Although by analyzing the frequency, damping ratio, and quality indicators provide sufficient modal information for online modal selection, quality indicator such as MSV is known to be not applicable for unstable modes. This is because the denominator of MSV in Equation (5.2) becomes negative, resulting in a imaginary value. For aircraft system identification with certain modes such as the lateral spiral mode known to be prone to be unstable, it is also important to calculate other quality indicators to ensure the integrity of the identified modes. Therefore, for offline data analysis additional quality indicators that involve more calculation cost are performed.

The Hankel matrix in Equation (4.14) can also be transformed into modal coordinates, recall that previously the more general form was presented. Taking k as 1 to form the $H(0)$ matrix, Equation (4.14) can be expressed in the form of

$$\begin{aligned} H(0) &= \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_\beta \\ Y_2 & Y_3 & \cdots & Y_{\beta+1} \\ \vdots & \vdots & \ddots & \vdots \\ Y_\alpha & Y_{\alpha+1} & \cdots & Y_{\alpha+\beta-1} \end{bmatrix} \\ &= \left[P\Sigma^{1/2}\Psi\right]\left[\Psi^{-1}\Sigma^{1/2}Q^T\right] \\ &= \widetilde{P_m}\widetilde{Q_m} \end{aligned} \tag{5.5}$$

With no noise, this equation can be also expressed in the form of

$$
\begin{aligned}
H(0) &= \begin{bmatrix} C_m B_m & C_m \Lambda B_m & \cdots & C_m \Lambda^{\beta-1} B_m \\ C_m \Lambda B_m & C_m \Lambda^2 B_m & \cdots & C_m \Lambda^{\beta} B_m \\ \vdots & \vdots & \ddots & \vdots \\ C_m \Lambda^{\alpha-1} B_m & C_m \Lambda^{\alpha} B_m & \cdots & C_m \Lambda^{\alpha+\beta-2} B_m \end{bmatrix} \\
&= \begin{bmatrix} C_m \\ C_m \Lambda \\ \vdots \\ C_m \Lambda^{\alpha-1} \end{bmatrix} \begin{bmatrix} B_m & \Lambda B_m & \cdots & \Lambda^{\beta-1} B_m \end{bmatrix} \\
&= \overline{P}_m \overline{Q}_m
\end{aligned}
\tag{5.6}
$$

Where $\overline{P}_m$ is the identified modal observability matrix and $\overline{Q}_m$ is the identified modal controllability matrix.

With the results calculated, a more comprehensive analysis of the modal behavior of the identified model can be accomplished. The Consistent Mode Indicator is well known for its ability to capture accurate both spatial and temporal consistency of the identified results, for each mode i, CMI is a multiplication of the Extended Modal Amplitude Coherence (EMAC) and the wighted Modal Phase Collinearity (wMPC)

$$CMI_i = EMAC \cdot wMPC_i \tag{5.7}$$

in which the temporal consistency is quantified by EMAC and the spatial consistency is captured through wMPC. It has been shown that both the temporal and spatial conditions needs to be satisfied to ensure the accuracy of the identified result [66].

5.2.1 Extended Modal Amplitude Coherence

The EMAC indicator is computed for each r inputs and m outputs for every mode, it quantifies the temporal consistency of the results of the identification. EMAC is an extension of the calculation of MAC indicators by adding in a weighting function. There are two kinds of modal amplitude coherence indicators, Input Modal Amplitude Coherence (IMAC), which calculates the cosine of the angle between $\tilde{q}_{mi}$ and $\bar{q}_{mi}$, and Output Amplitude Coherence (OMAC), which is calculated similarly but using $\tilde{p}_{mi}$ and $\bar{p}_{mi}$ instead. IMAC is calculated based on the comparing the experimental data with the identified model

$$IMAC_i = \frac{|\tilde{q}_{mi}\bar{q}^*_{mi}|}{\sqrt{\tilde{q}_{mi}\tilde{q}^*_{mi}}\sqrt{\bar{q}_{mi}\bar{q}^*_{mi}}} \tag{5.8}$$

and similarly for OMAC,

$$OMAC_i = \frac{|\tilde{p}_{mi}\bar{p}^*_{mi}|}{\sqrt{\tilde{p}_{mi}\tilde{p}^*_{mi}}\sqrt{\bar{p}_{mi}\bar{p}^*_{mi}}} \tag{5.9}$$

where the superscript $*$ represents a vector transpose and conjugate. From IMAC and OMAC a total MAC can be formed by

$$TMAC = IMAC \cdot OMAC \tag{5.10}$$

The EMAC extends the total MAC calculation by adding in a weighting term W_ij, for each mode i, output response j, and initial condition k

$$\begin{aligned} EMAC^I_{ij} &= IMAC \cdot W_{ij} \\ EMAC^O_{ik} &= OMAC \cdot W_{ik} \end{aligned} \tag{5.11}$$

note that the superscript I and O stands for input and output respectfully. The weighting term is calculated using the following equation,

$$W_{ij} = 1 - (|P_{ij}|/(\pi/4)) \tag{5.12}$$

where

$$P_{ij} = \left\{ \begin{array}{ll} = \dfrac{\phi_{ij}}{\bar{\phi}_{ij}} & \forall \ |P_{ij}| \leq \pi/4 \\ = 0 \quad otherwise & \end{array} \right\} \tag{5.13}$$

The ϕ_{ij} term is substituted with b_{mi} or c_{mi} depending on the calculation of input or output EMAC. Finally, the total EMAC is acquired similarly to Equation (5.10) by multiplying the $EMAC^I_{ij}$ $EMAC^O_{ik}$ at each $j-k$th input-output pair for the ith mode

$$EMAC_{ijk} = EMAC^I_{ij} \cdot EMAC^O_{ik} \tag{5.14}$$

5.2.2 Weighted Modal Phase Collinearity

The MPC quantifies the spatial consistency of the identified results.

$$\begin{aligned} S_{xx} &= (c^r_{mi})^T \cdot (c^r_{mi}) \\ S_{yy} &= (c^i_{mi})^T \cdot (c^i_{mi}) \\ S_{xy} &= (c^r_{mi})^T \cdot (c^i_{mi}) \end{aligned} \tag{5.15}$$

with

$$\eta = \frac{S_{yy} - S_{xx}}{2S_{xy}} \tag{5.16}$$

the eigenvlaues are

$$\lambda_{1,2} = \frac{S_{yy} - S_{xx}}{2} \pm S_{xy}\sqrt{\eta^2 + 1} \tag{5.17}$$

and finally the MPC for mode i is defined as

$$MPC_i = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2 \tag{5.18}$$

5.3 Nominal Model Selection

The question of when to update an identified model is a difficult. Updating a model clearly is necessary when the system performance is degrading to a level which is not acceptable. So, the key is to define a performance index. There is no need to update the identified model when the performance is reasonable based on the performance index. To transmit from a model to another model is not an easy task for updating controller designs because the identified model and the updated model are not necessary in the same coordinate systems. The control force generated by the updated model may be considerably different from the original identified model at the transition moment. It results in a rough transition yielding a sharp system response that may not be acceptable.

A simple but yet efficient method of comparing two different identified models is by analyzing the tracking between two signals. The Theil Inequality Coefficient (TIC) is a normalized metric that quantifies the identified response in terms of residuals. Calculating the TIC for each output variables using Equation (5.19) yields a total TIC of the identified model.

$$\mathcal{T}_i = \frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[\hat{y}_i(k) - y_i(k)\right]^2}}{\sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[\hat{y}_i(k)\right]^2} + \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[y_i(k)\right]^2}} \quad i = 1, 2, \cdots, N \tag{5.19}$$

Where $\hat{y}$ is the estimated output from the identified model, and y is the measured output for each input i. The TIC has a value between 0 to 1, and a decent fit on a specific output should have a $\mathcal{T}_i < 0.3$ [67]. Therefore, in order to determine if an identified model is better than the current nominal model the process shown in Figure 5.2. Where m is the number of output measurements made and the dual subscripts for TIC values have two separate meanings, the first character indicates the model used to calculate the TIC value, and the second character points out the input set that is applied on the model of interest. For example, TIC_{jN} represents the TIC value calculated from the jth identified model by applying the input data set that was used to excite the nominal model.

The n step nominal update procedure is as follows, any answer of **NO** will result in a suggestion of not updating the nominal model:

1. Check model stability of the newly identified model from the jth experiment

2. Check the quality indexes

3. Check if the total TIC value of the jth identified model using the jth input data set TIC$_{jj}$ has an average TIC$_{avg}$ below 0.3
4. Calculate the total TIC value of the jth identified model using the Nth input data set TIC$_{jN}$ for model verification
5. Compare TIC$_{jN}$ with previously saved TIC$_{NN}$
6. Calculate the total TIC value of the Nth identified model using the jth input data set TIC$_{Nj}$
7. Compare the final TIC values by adding the TIC

The online system suggests the GCS whether to update the model based on TIC values.

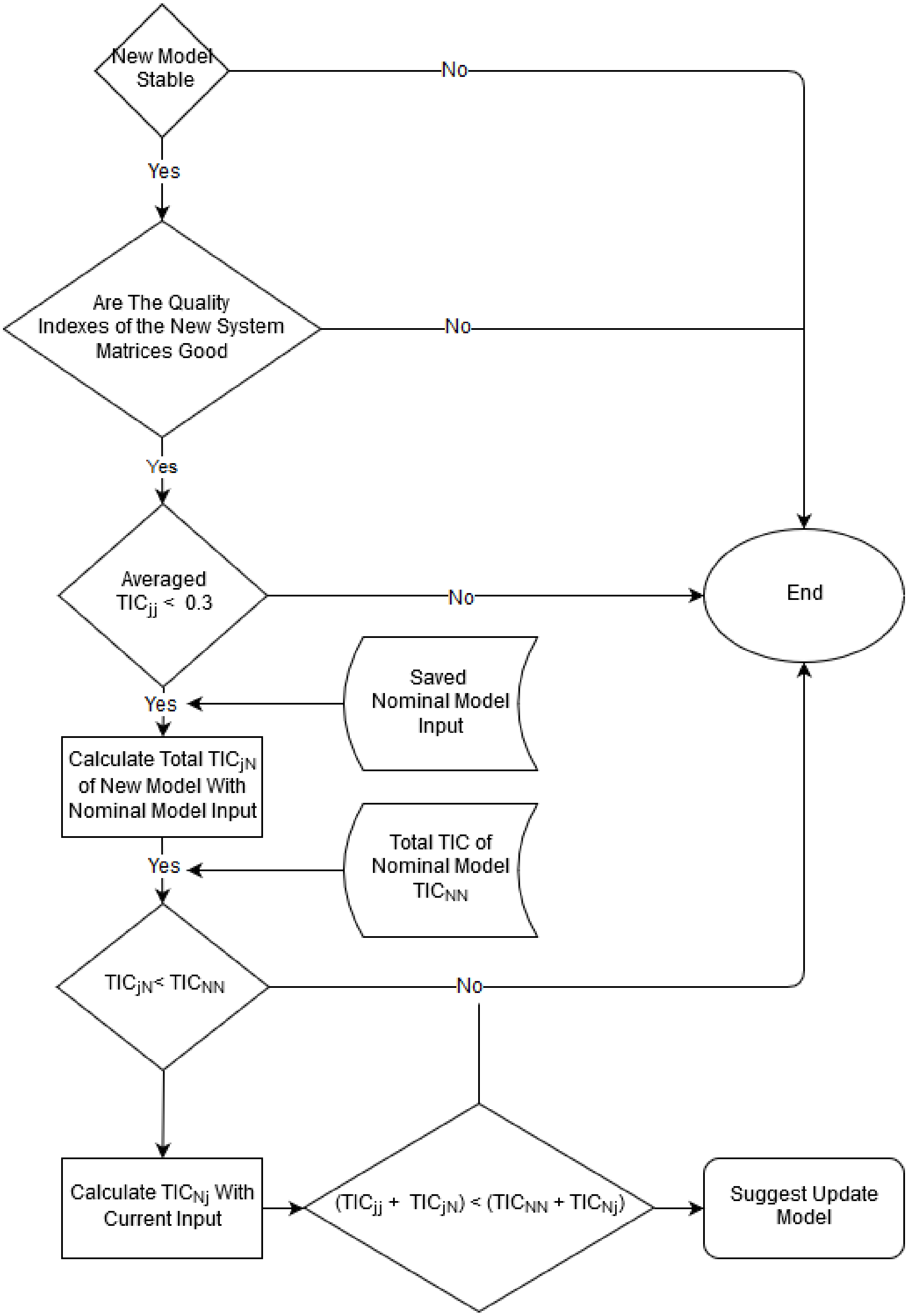


Figure 5.2: Procedure of model update suggestion by comparing TIC values.

6. TESTING AND RESULTS*

Flight tests were performed at Texas A&M University RELLIS Campus in College Station, TX. All tests were conducted during early morning to minimize the effect of wind aloft. To identify the vehicle model at a trimmed point, test inputs were applied after the aircraft is manually trimmed. Offline system identification was monitored using Mission Planner via a 915 Hz radio, and for online system identification, data was transmitted using XBees. Both piloted and automated excitations are injected during flight. Due to the inherent nature of OKID, identifying a local linear time invariant model, the inputs are preferred not to excite too much nonlinearity of the desired plant. Note that all the flight results presented are trimmed by the same UAS pilot. Additionally, offline identifications were analyzed using Matlab, and online identifications were performed in the Python environment. Note that part of the data collected were previously used in previous publication.

6.1 Offline System Identification Results With Human Inputs

The flight test results for offline system identification in this subsection were conducted with wind speed around 2 mph gusting to 5 mph prevailing from North Northeast (NNE) of the flight path. The input maneuvers were generated by the human pilot on the ground via the transmitter. Flight data was downloaded to a hard drive and analysis were conducted using the SOCIT package in MATLAB®post flight.

6.1.1 Longitudinal

Figure 6.2 shows the identified longitudinal model in the dashed red line and flight test data in the solid blue. For SUAS with fairly rigid structures the sine sweep excitation can also be successfully used for system identification using data acquired by the instrumentation system.

---

*Part of the data sets used for online identification results were reused with permission from "Online Near Real-Time System Identification on a Fixed-Wing Small Unmanned Air Vehicle" by Han Hsun Lu, Cameron Rogers, Vinicius G Goecks, John Valasek, in *2018 AIAA Atmospheric Flight Mechanics Conference, AIAA SciTech Forum*, copyright 2018 by Han Hsun Lu.

Table 6.1 shows the identified modes and characteristics using OKID.

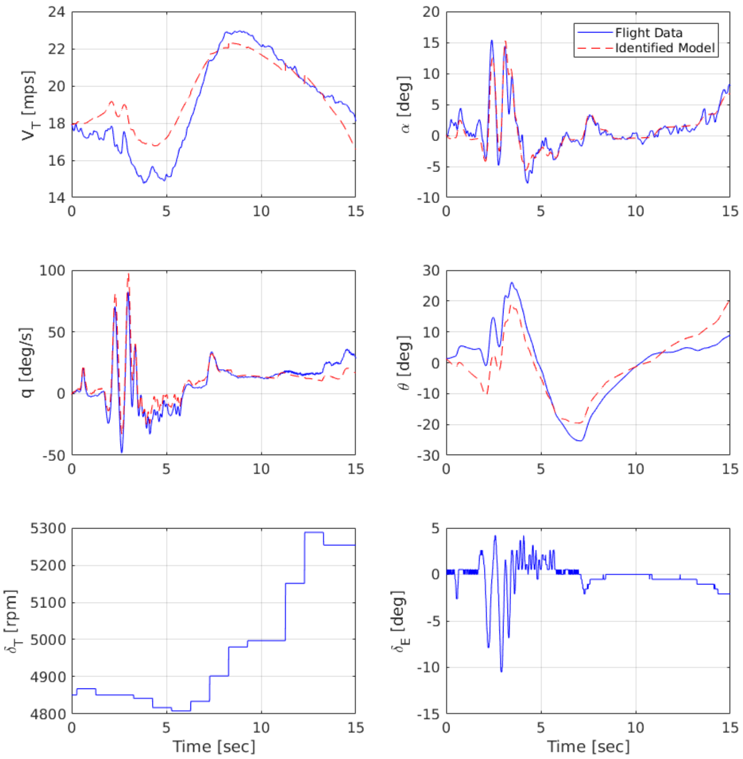


Figure 6.1: Longitudinal identification with frequency sweep excitation.

Table 6.1: Super Cub longitudinal dynamic modes.

Mode	**Phugoid**	**Short Period**
Eigenvalue	$-0.03 \pm \mathrm{j}0.09$	$-6.09 \pm \mathrm{j}2.63$
Damping Ratio	0.33	0.92
Natural Freq. (rad/s)	0.09	6.6
MSV (%)	100.0	15.1
MCI (%)	79.3	100.0
MOI (%)	64.8	100.0
CMI (%)	92.5	8.0
EMAC (%)	94.0	27.1
MPC (%)	98.4	29.4

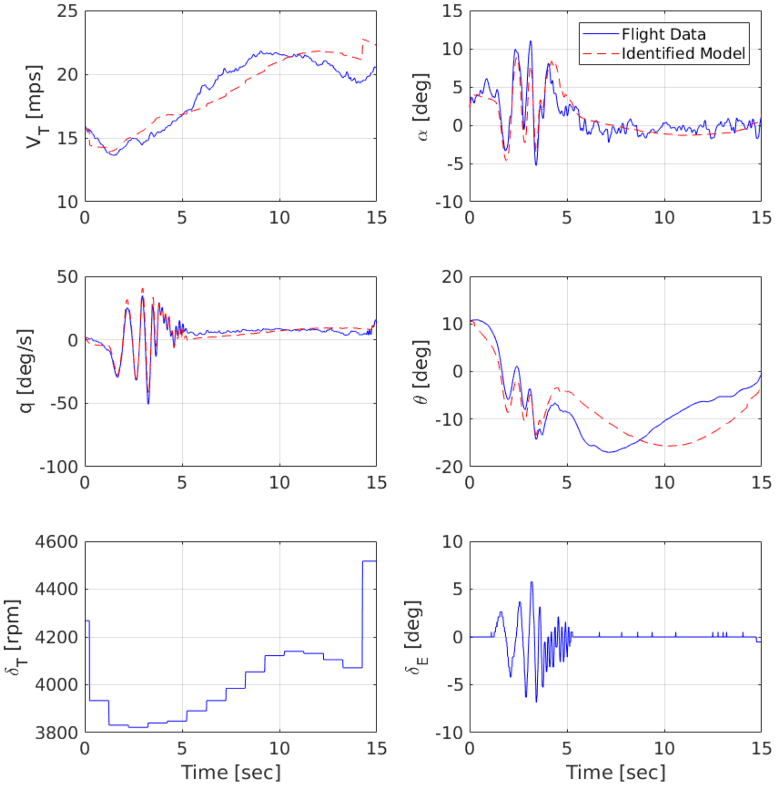


Figure 6.2: Longitudinal model verification with alternate excitation sets

6.1.2 Lateral/Directional

The same method is implemented on the lateral axis using doublet and sine sweep excitations. Figure 6.3 shows the results from a lat/d maneuver set. The identified states are shown as dashed red while the actual flight data is solid blue. The identified model tracks all four states and system frequencies very well. Note that the discrepancy in sideslip angle after two seconds is caused by the ±20° measuring limit imposed by the μADC.

A filter is added using local regression weighted linear least squares and a second degree polynomial to filter out the measurement noise for the state measurements. Filtered state measurements are then used for system identification calculations.

The identified modes and characteristics using OKID are shown in Table 6.2.

Table 6.2: Super Cub lateral/directional dynamic modes.

Mode	**Spiral**	**Roll**	**Dutch Roll**
Eigenvalue	-0.038	-0.54	$-2.73 \pm \text{j}2.64$
Damping Ratio	—	—	0.71
Natural Frequency (rad/s)	—	—	3.76
MSV (%)	79.2	100.0	81.3
MCI (%)	9.4	79.4	100
MOI (%)	83.6	100.0	89.6
CMI (%)	0.0	0.0	18.1
EMAC (%)	0.0	0.0.0	50.7
MPC (%)	0.0	0.0	35.7

To verify the integrity of the identified model, the identified model is simulated with another set of inputs. These simulation results are displayed in Figure 6.4. The identified model shows good consistency with the measured state measurements, and it can be concluded that the identified model is reasonably close to the true vehicle model. The results show that despite experiencing signal clipping, the identified method can still successfully identify the system dynamics without errors.

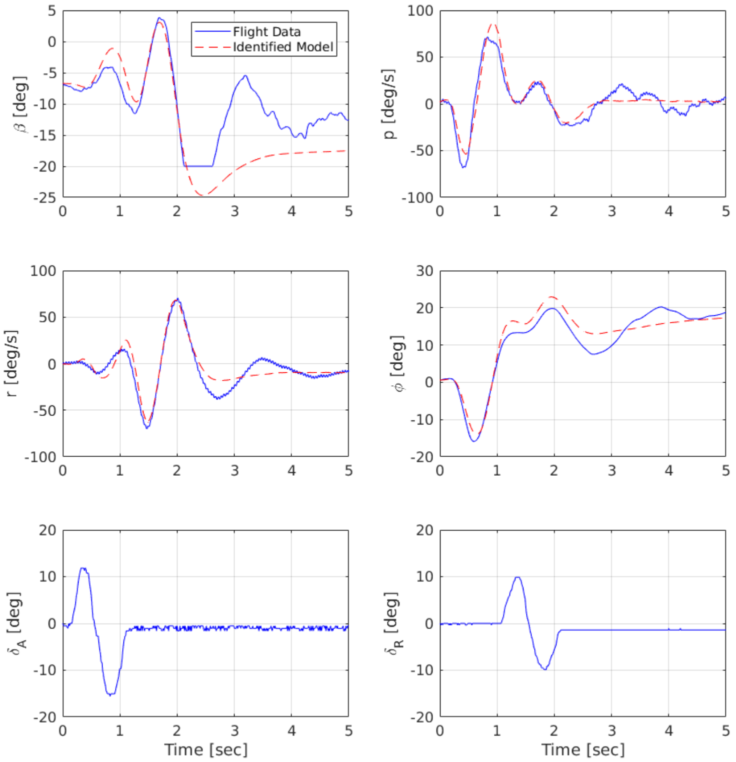


Figure 6.3: Comparison between flight data and identified lateral/directional model.

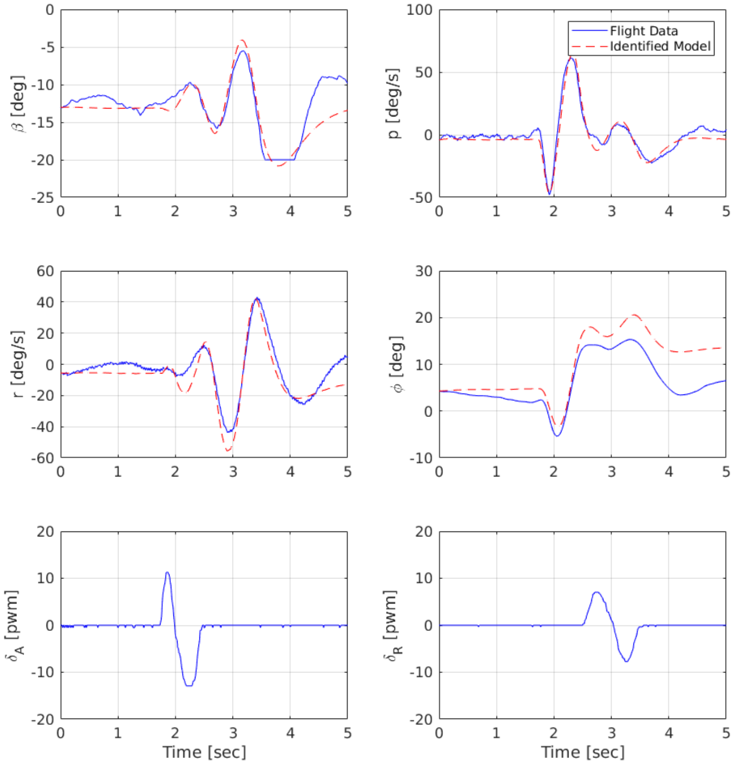


Figure 6.4: Lat/D model verification with alternate excitation sets

6.1.3 Summary on Piloted Results

Despite showing good tracking on both lateral/directional and longitudinal system identification, piloted maneuvers lacks consistency even with experienced pilots. Analyzing data from 11 different flight days with 59 excitation maneuvers combining both lateral/directional and longitudinal maneuvers, piloted maneuvers averaged a 24 percent success rate in identifying a linear model with the right rank and modes. Within these models, merely 6 flight maneuvers yielded an identified model that tracks the output response extremely well with a $\text{TIC}_{avg} < 0.3$. Figure 6.5 shows an example of the identified results by plotting the identified eigenvalues of the lateral/directional model from single flight day. The red and green markers represent the preferred identified models with a low TIC value and the blue markers indicate other data sets with similar flight conditions that had a poor TIC value. The red marker is the data set that is identified as the nominal identified model with the lowest average TIC value.

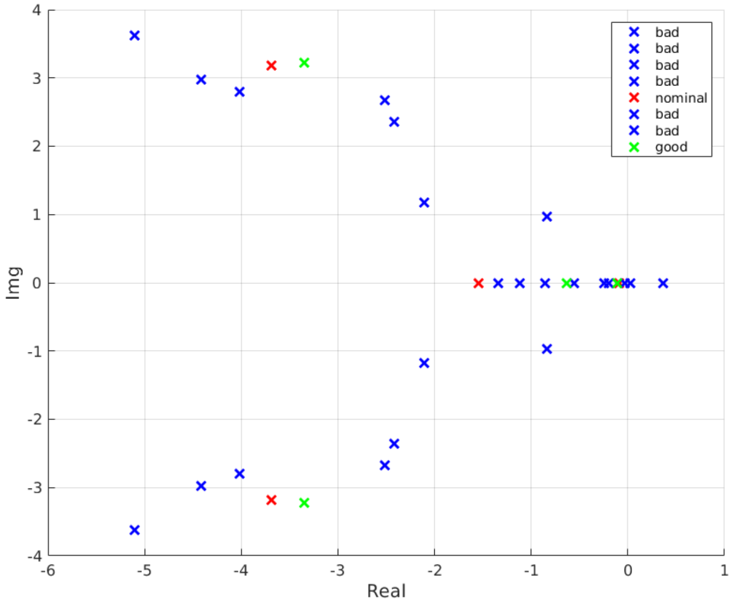


Figure 6.5: Accumulated identified eigenvalues of the multiple piloted and automated excitation results

The results show that even with the correct rank and number of modes, the identified eigenvalues range across a large region. Analyzing the acquired data, several issues may result in the changes. The first obvious reason is the variations between different flight maneuvers, that is the duration and magnitude of the input change between each piloted maneuver. It is known that inputs with large magnitudes drive the system to deviate the linear range. For most of the identified models with poor tracking consistency, variations between piloted inputs are the main reason a poor identification.

The other reason regarding model variations is the difference between each trimmed straight and level flight. Before each flight maneuver, the pilot manually trims the vehicle to a wings level flight. Due to both human operational differences and sudden wind gusts, the angle-of-attack α and the body axis roll angle ϕ was seen to variate slightly between each flight path.

6.2 Data Analysis With Automated Input

Automated inputs detailed in Section 4.4.5 were applied during each flight maneuver using the online identification procedure from Section 5.1. The input functions were preloaded onboard the vehicle and called when the GCS operator presses the "Update Input" button on the GUI. Figure 6.6 presents an example of the flight path from a lateral/directional input maneuver overlaid in Google Earth. Note that the flight path log file used is recorded onboard the Pixhawk log file.

The red section indicates the location where the pilot enabled the excitation on the transmitter and both blue and orange sections displays the flight path while the vehicle is in excitation mode for two separate flight paths.

The automated input function not only allows the user to specify a desired input excitation but also improves the repeatability of the excitations. Figure 6.7 compares the measured input of two excitations with identical settings. The results show that the two signals have a averaged TIC value of 0.007 and can be viewed as identical.

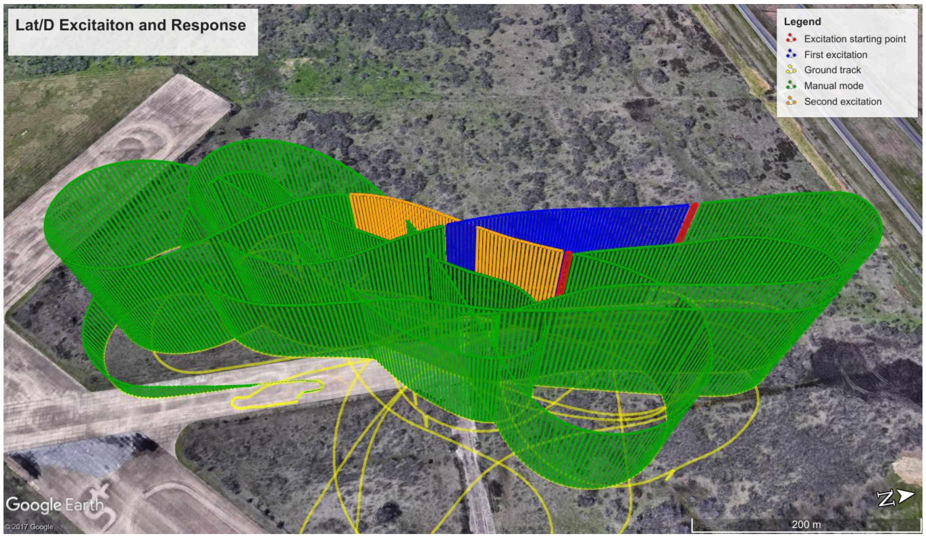


Figure 6.6: Flight path of a lateral/directional path using the developed automated excitation method.

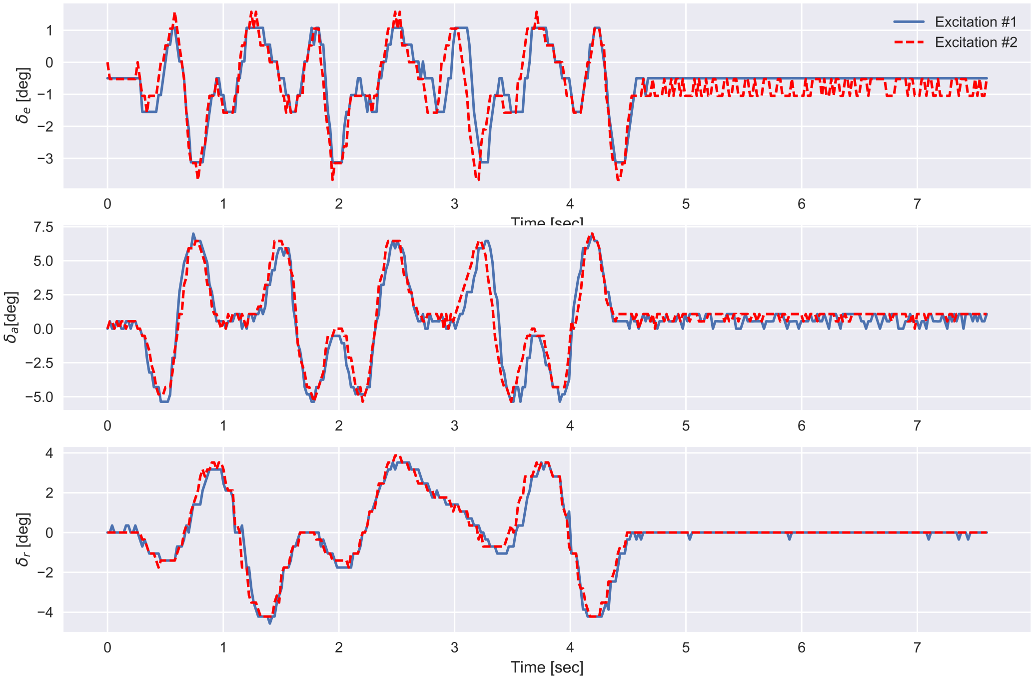


Figure 6.7: Comparison of two excitation signals with identical settings

6.2.1 Offline Frequency Domain OKID Results

In this section frequency domain OKID is applied to the data collected from automated applying multi-sine sweeps to multiple control surfaces. Automated input enables periodic excitations that are required for frequency domain methods. Data is analyzed offline in Matlab using SOCIT. The main goal is to apply frequency domain OKID to get similar results as analyzing the time domain data. The FRF is acquired using Equation (4.21) to Equation (4.23).

For lateral/directional system identification, the same multi-sine input used for time domain OKID was applied. Figure 6.8 shows the Input signal. A multi-sine input with the same configuration as listed in Table 6.4 was applied on the aileron and the rudder.

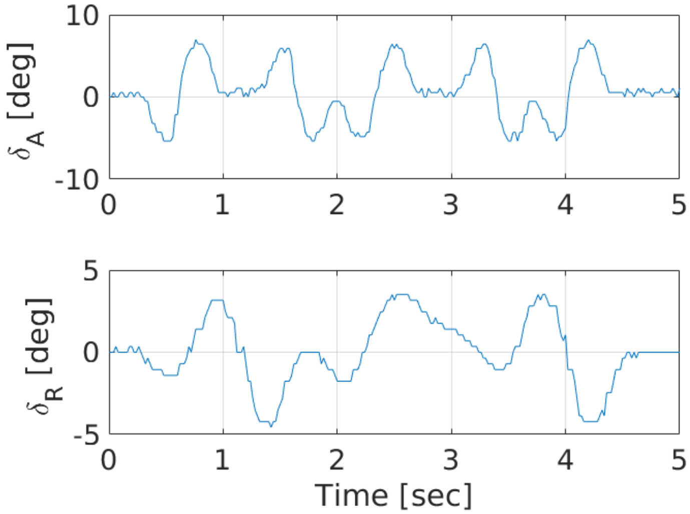


Figure 6.8: Input used for lateral/directional frequency domain system identification

The FRF of the input-output signal from this excitation was calculated and is plotted in Figure 6.9.

The identified results using frequency domain OKID is shown in the following table:

The measured and identified output response is shown in Figure 6.10.

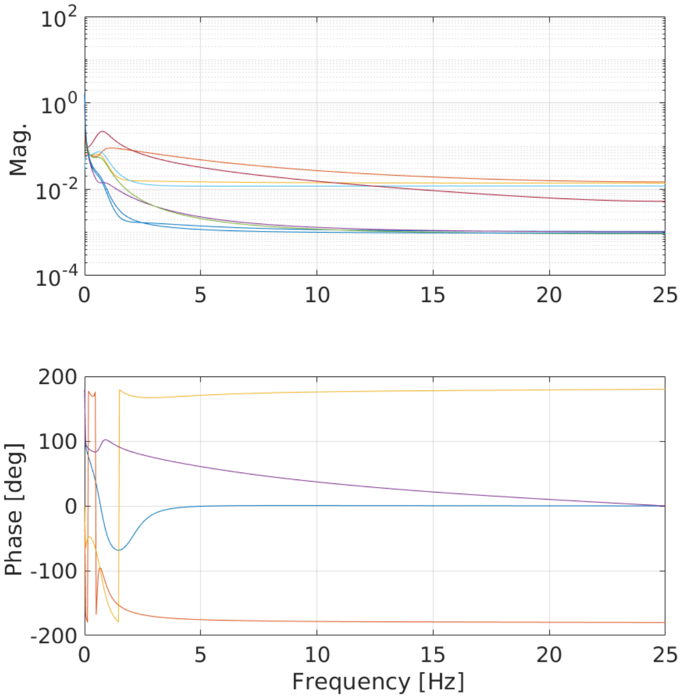


Figure 6.9: Frequency response function of a lateral/directional input-output data set

Table 6.3: Super Cub lateral/directional dynamic modes using frequency domain OKID.

Mode	**Spiral**	**Roll**	**Dutch Roll**
Eigenvalue	-0.04	-1.83	$-2.16 \pm j4.46$
Damping Ratio	—	—	0.44
Natural Frequency (rad/s)	—	—	4.96
MSV (%)	100.0	2.7	20.4
MCI (%)	56.7	28.4	100.0
MOI (%)	80.6	43.5	100.0
CMI (%)	0.0	0.0	0.0
EMAC (%)	0.0	0.0	0.0
MPC (%)	0.0	0.0	45.8

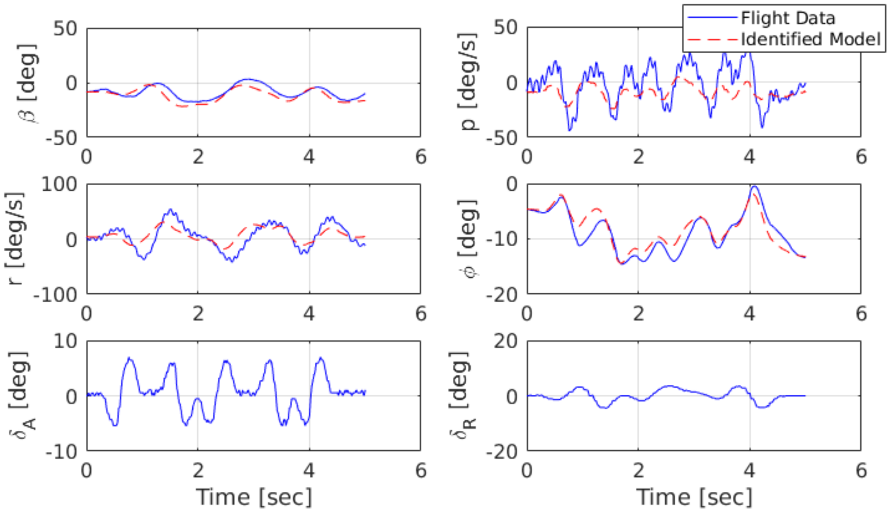


Figure 6.10: Identified lateral/directional output response

6.2.2 Online Full Lat/Lon Identification Results

One of the objectives of this thesis is to analyze the different excitation methods for effective online near real-time system identification. Full lateral/longitudinal identification has been in particular of interest due to the possibility of exciting all the vehicle dynamic modes though a short period of excitation. In general, short excitation time is preferred over a long excitation because the plane tends to deviate further from the trim condition as the excitation duration prolongs. In addition, although lateral/directional and longitudinal input excitations have shown to be effective on identifying the appropriate dynamic modes separately, the main assumption is that the lateral directional modes and longitudinal modes are fully uncoupled, which means that no cross-axis terms exist. However, it is also known that as AOA of the aircraft increases, the dynamic coupling between lateral/directional and longitudinal modes can become more significant [68]. Moreover, for fully nonlinear aircraft models lateral/directional inputs affect longitudinal states. Figure 6.11 is an example of how the uncoupled assumption differs from the coupled response from a single large rudder input.

For SUAS system identification related research it is especially important that the coupling effects are not totally ignored. Trimming from a large distance away is not a trivial task, and even

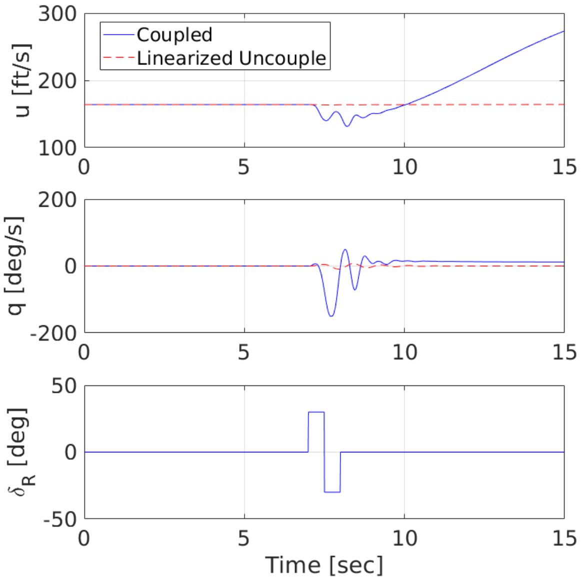


Figure 6.11: Comparison of linear and nonlinear coupling response of a large rudder input.

with an experienced remote pilot it is difficult to trim the UAS perfectly at 0 degrees AOA without a first person view.

In this subsection, a full lateral/directional and longitudinal excitation combination is applied to acquire a full dynamic model. With the goal to excite the dynamic modes in a short time period effectively, several input excitation combinations were tested. In Section 4.4.5, it has been pointed out that multi-sine inputs has the advantage of having a low RPF, keeping the vehicle close to the trimmed flight condition, and is preferred for multi-surface excitation. Table 6.4 displays the input details for each multi-sine input and its RPF. Note that the A_k is the amplitude commanded in terms of PWM. All the excitation inputs and analysis results shown in this subsection is directly collected via flight test with no post additional processing. The input and output sets used are from the log files and the identification code is the exact same code onboard the vehicle.

It is preferred to also apply a low frequency multi-sine input on the throttle due to the benefits mentioned in previous sections. However, experiments have shown that the time delay caused by the ESC response time restricts the excitation efficiency. Moreover, the asymmetric nonlinear change on ramping throttle up and down creates results in a large RPF. Therefore, applying expe-

Table 6.4: Input parameters for multi-sine excitation.

Input	A_k **(PWM)**	**k**	RPF
δ_a	40	$3, 6, 9, 12$	1.51
δ_r	40	$5, 10, 15, 20$	1.21
δ_e	40	$7, 14, 21, 28$	1.52

rience from the piloted maneuvers for longitudinal system identification, a step-like input similar to the longitudinal input in Figure 4.4 was applied to the throttle. Figure 6.12 is the measured combined input from DFTI.

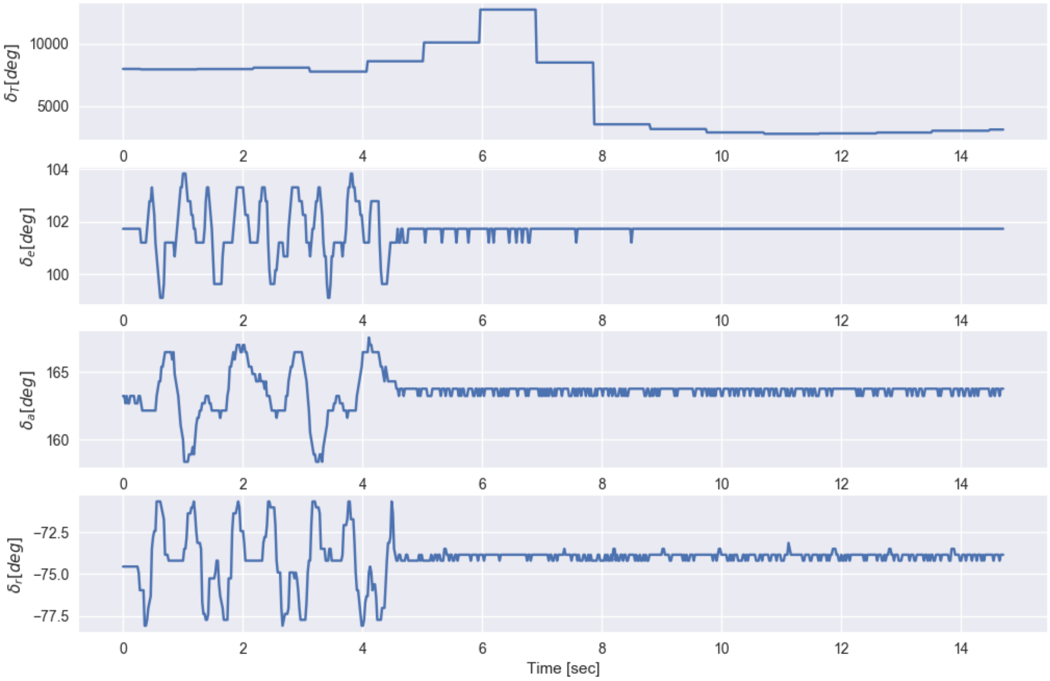


Figure 6.12: Combined multi-sine excitation applied on aileron, rudder, and elevator followed by a step-like throttle input sequence.

The identified and measured output results are shown in Figure 6.5. Similar to previous results, the dashed red signal represents the identified signal while the blue signal is the original measured data. The results yielded an averaged TIC value $T_{avg} = 0.21$ indicating very good tracking on between the estimated and measured responses and the identified details can be seen in Table 6.5. Additionally, the identified system matrix can be found in Appendix A.2.

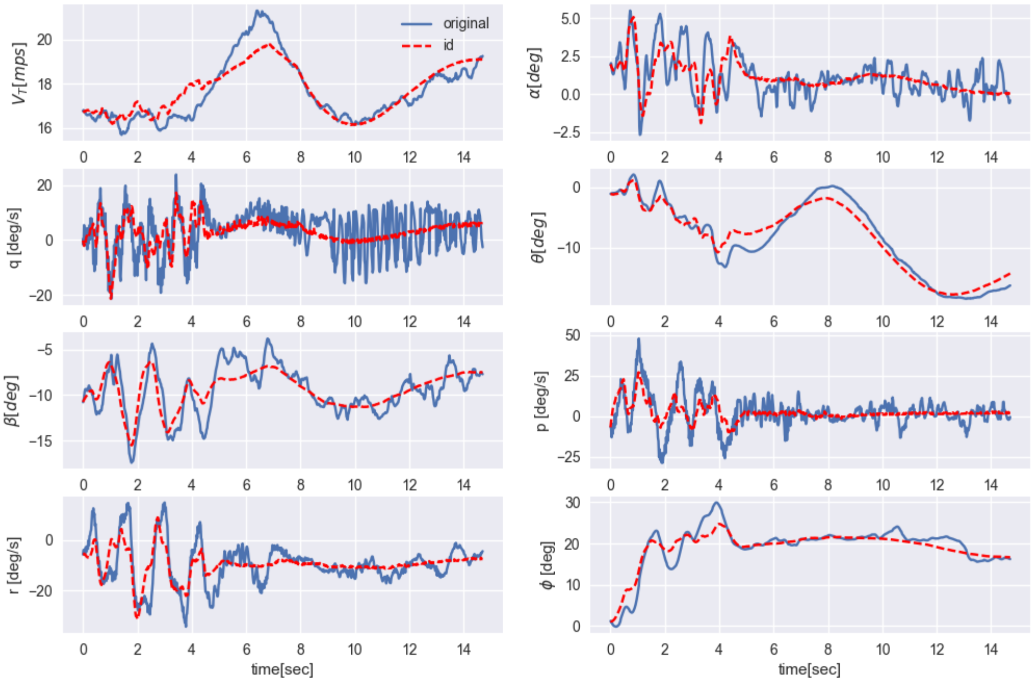


Figure 6.13: Full lateral/directional and longitudinal identification with the combined input excitation set.

Other inputs sequences that were tested includes moving the throttle input sequence forward before the multi-sine input started as shown in Figure 6.14.

The results from this input showed reasonable identification quality but the success rate of identifying a model with an average TIC value below 0.3 was low. In some of the identified data sets, the slow phugoid mode fails to develop before the lateral/directional inputs initiate.

Table 6.5: Super Cub full Lat/Lon dynamic modes.

Mode	**Phugoid**	**Short Period**	**Roll**	**Spiral**	**Dutch Roll**
Eigenvalue	-0.19 ±j0.48	-4.88 ±j3.23	-1.72	-6.31	-1.5 ±3.7
Damping Ratio	0.36	0.83	—	—	0.29
Natural Freq. (rad/s)	0.5	5.84	—	—	3.96
Time Const. (sec)	—	—	1.19	0.03	—
MSV (%)	100.0	31.7	36.5	42.1	28.8
MCI (%)	35.7	46.7	100.0	28.7	13.3
MOI (%)	49.6	92.9	73.4	100.0	85.4

Accordingly, throttle input is preferred to be introduced after lateral/directional inputs are excited.

Another aspect of interest is the amount of response time needed for a successful identification. The amount of time allowing the vehicle response to develop was changed for this experiment. To illustrate, the identified results were closely monitored while the pilot gradually reduces the free response time of each path. Quality indexes, TIC value, as well as identification success was analyzed and compared.

Figure 6.15 and Figure 6.17 shows the input and output signals of a short response excitation maneuver. It can be seen that the pilot took back control of the vehicle right after the excitation was completed. Figure shows the identified results. Information includes including identified frequency, damping ratio, MSV, MCI, MOI, TIC value, and plotted figure of the eigenvalues are shown.

6.2.3 Post Flight Data Analysis

There are several differences between post flight data analysis and onboard data analysis. Although post flight analysis is conducted using the exact same program used for onboard analysis along with data log files stored during flight. The main differences are the capability to manually change the starting time of identification, the option to apply a different signal filtering, and the ability to modify extensive parameters. In this subsection, the differences mentioned above are discussed.

In order to address the effect of the starting point of the identification process, it is important

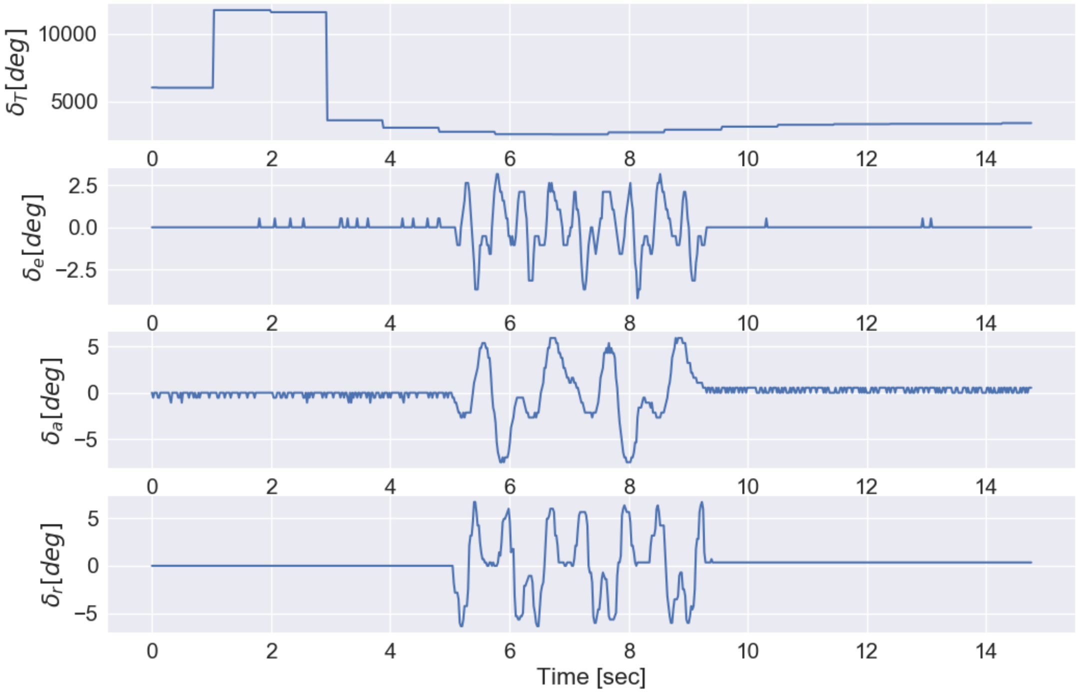


Figure 6.14: A step-like throttle input followed by a combined multi-sine excitation applied on aileron, rudder, and elevator sequence.

to understand the underlying delay time in the system. The starting time variates within a 0.8 sec due to both the time delay caused by the multiplexer along with the general system delay. The delay caused by the multiplexer is the difference between software communication and hardware channel change. Once the pilot initiates the excitation on the transmitter, Clark reads the channel change signal and passes a "Logging start" signal to the system identification subsystem. At the same time the output channels on the Pixhawk are switched by the hardware multiplexer. The difference between the two process may result in a time delay and sometimes poses a problem for real-time identification system if the pilot was still applying inputs. Figure 6.18 shows how this can possibly effect the identified results. The red solid line indicates the starting point of the automated excitation, which is the ideal identification starting time. It can be seen that some aileron input was still being applied before the excitation was initiated. Therefore, instead of a fixed pre-specified

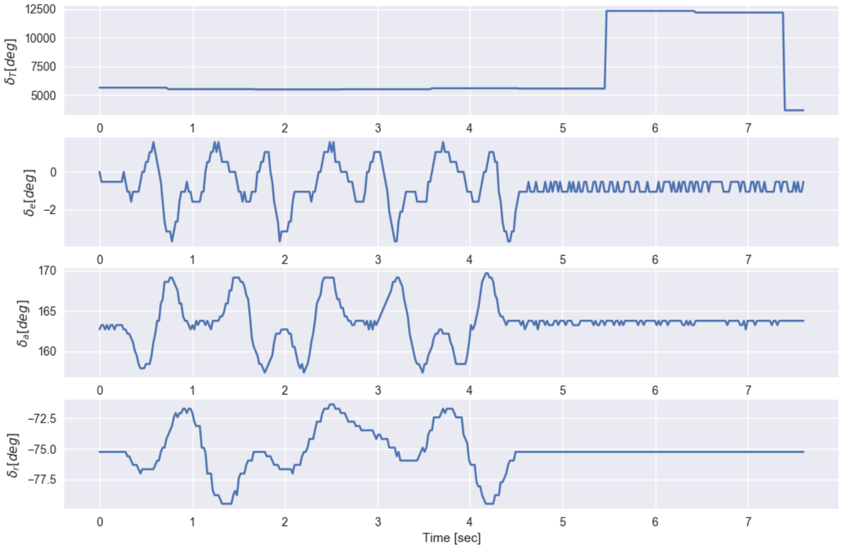


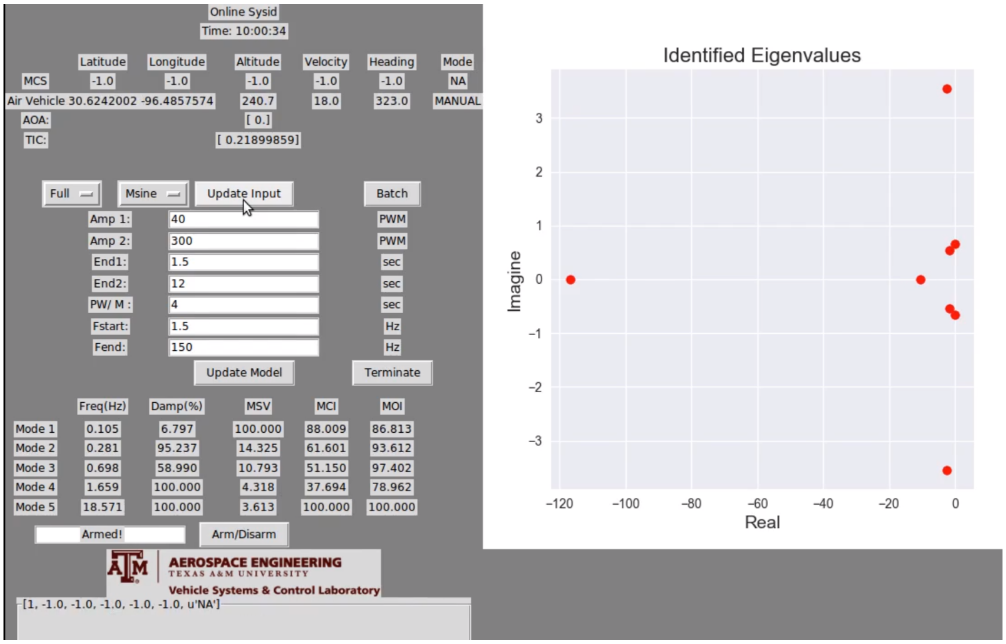
Figure 6.15: Combined input excitation sequence with short response time.

starting point, the starting point of identification can be manually chosen to eliminate this effect.

The second difference is the ability to apply different signal filtering via signal post-processing. This discrepancy is minor as the measurement noise have shown to not have a significant effect on the identification results, and therefore, the same filtering method is applied on the online identification process.

Although most identification parameters can be modified through the communication GUI, the ideal number of certain parameters may vary with each flight path. Such parameters include the identified the number of observer Markov parameters p and the system order n. For online identification, p is set to a fixed value 10, and n is set to the desired system order. In general, p is chosen from experience and is required to follow certain criteria. That is, a system with system order n and m outputs, the minimum number of OMP required to be identified is

$$p_{min} = int(\frac{n}{m}) \tag{6.1}$$


Online Sysid

Time: 10:00:34

	Latitude	Longitude	Altitude	Velocity	Heading	Mode
MCS	-1.0	-1.0	-1.0	-1.0	-1.0	NA
Air Vehicle	30.6242002	-96.4857574	240.7	18.0	323.0	MANUAL
AOA:			[0.]			
TIC:			[0.21899859]			

Amp 1: 40 PWM
Amp 2: 300 PWM
End1: 1.5 sec
End2: 12 sec
PW/ M : 4 sec
Fstart: 1.5 Hz
Fend: 150 Hz

	Freq(Hz)	Damp(%)	MSV	MCI	MOI
Mode 1	0.105	6.797	100.000	88.009	86.813
Mode 2	0.281	95.237	14.325	61.601	93.612
Mode 3	0.698	58.990	10.793	51.150	97.402
Mode 4	1.659	100.000	4.318	37.694	78.962
Mode 5	18.571	100.000	3.613	100.000	100.000

Armed!

```
[1, -1.0, -1.0, -1.0, -1.0, -1.0, u'NA']
```

Figure 6.16: Screen capture of the identification GUI with identified results plotted in near real time.

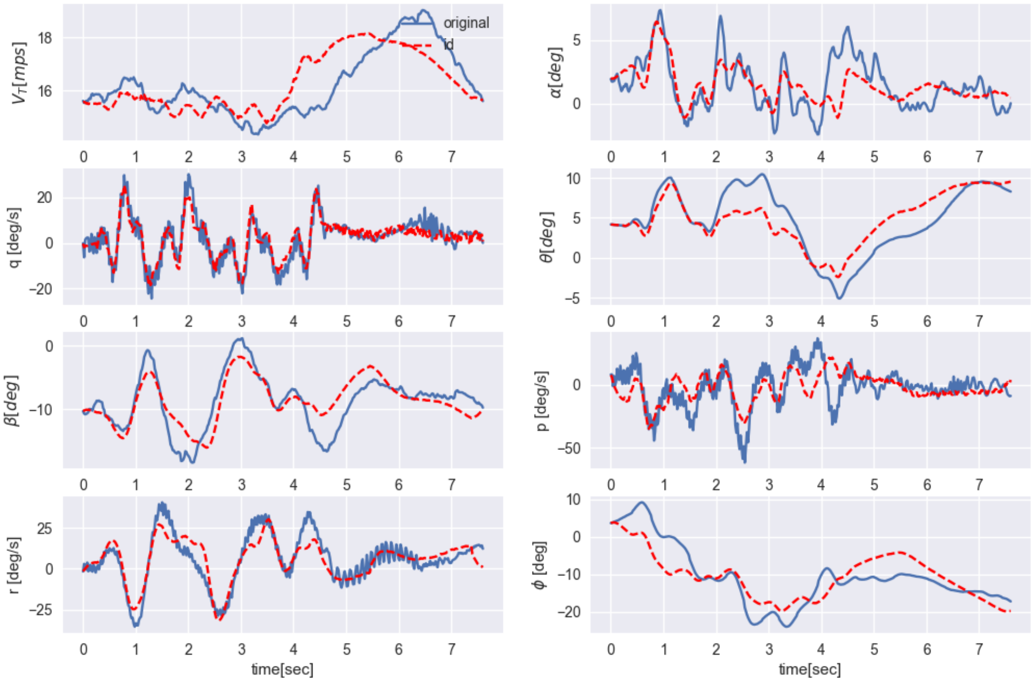


Figure 6.17: Identified results from combined input excitation sequence.

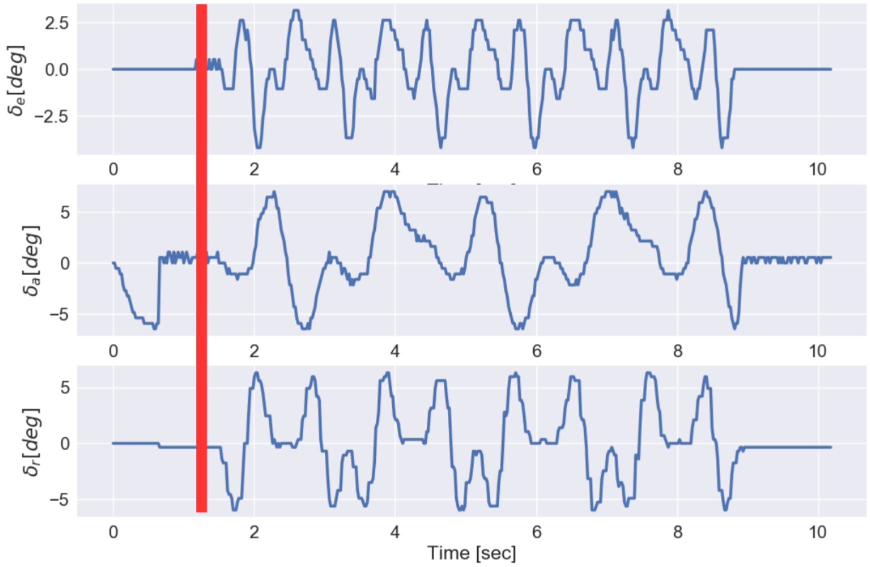


Figure 6.18: Identification start time during flight test compared to the ideal starting time. The red line indicates the decision made by analyzing the data post flight.

and the maximum order of the system that can be recovered is mp. Take a system with a system order of 5 with 8 measured outputs as an example. The p_{min} would be 1, and the maximum order of the system that can be realized will be 8. In theory, a large enough p should be chosen to be able to realize a system with an unknown order. However, a larger number does not guarantee yielding a better identified result. Using the same flight data, Figure 6.19 illustrates an example of an offline flight analysis by iterating different p values to find the ideal value. In this case, the online identification program had a pre-assigned value of $p = 10$, and the TIC$_{avg}$ values with different p is shown in Table 6.6. From the table it is shown that the larger p value does not correlate to a small TIC$_{avg}$. For this specific case, $p = 1$ would have been the preferred value assigned. This iterative process is currently not implemented on the flight software, but may be easily add into the system with simple modifications. However, it is worthy to note that the processing time will increase significantly on the current ARM micro-controller chip.

Table 6.6: TIC_{avg} values with different p values.

p	$\textbf{TIC}_{avg}$
1	0.2715
2	0.3241
3	0.3395
5	0.3655
7	0.3375
10	0.3555

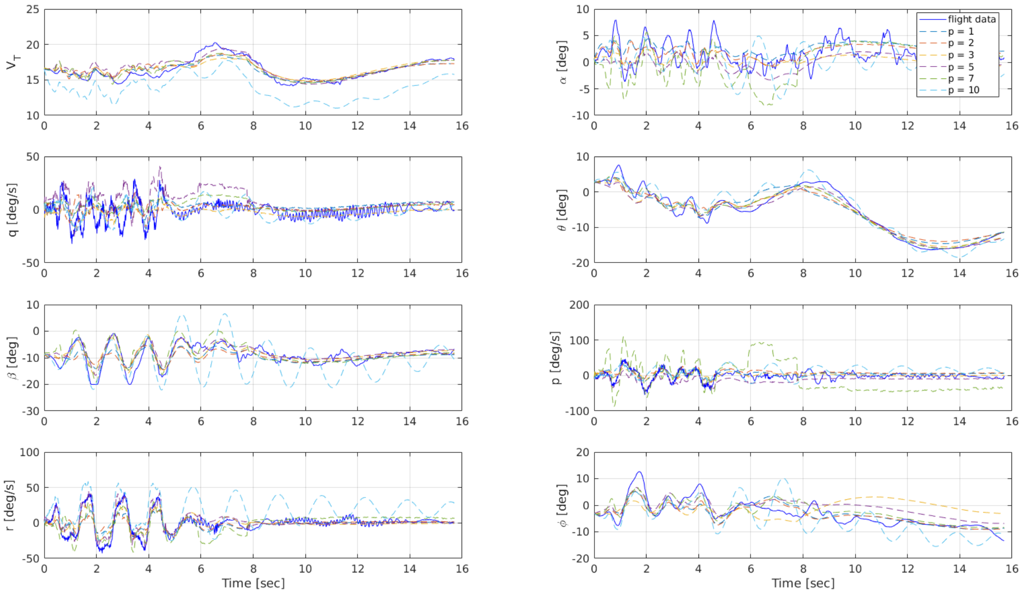


Figure 6.19: Output responses with different p compared to flight data.

6.2.4 Summary for Online Identification with Automated Inputs

Automated input excitation have shown to be the superior over piloted excitation maneuvers both in accuracy and repeatability. It is important for online near real-time system identification and can also be utilized to perform different excitation sequences for more efficient excitations. In terms of the quality of identified results, excluding flights in the developmental stage, the proposed combined lateral/directional and longitudinal excitation sequence recorded a 7 out of 21 success rate of identifying a system model with the correct dynamic modes and a $\text{TIC}_{avg} < 0.3$. Increasing the success rate by 23 percent.

7. CONCLUSIONS AND RECOMMENDATIONS

This chapter concludes the research presented in previous chapters and further on make recommendations for future advancements.

7.1 Conclusions

1. From the identified results, the design of a flight test instrumentation capable of accurately measuring aircraft state and control time histories at 100 Hz demonstrated the capability to support online near real-time system identification.

2. An automated method of applying excitation inputs was shown to be capable of applying various excitations of the users choice. The method yielded a repeatable and precise excitation with 99 % of similarity and identical power spectrum.

3. An input combination was introduced for a combined lateral/directional and longitudinal system identification on a Small Unmanned Air Systems. The full surface excitation was shown to be capable of exciting all the vehicle dynamic modes effectively in a single excitation.

4. The human-in-the-loop system identification procedure along with the developed system have shown to be capable of performing near real-time system identification using Commercial-Off-The-Shelf products. The developed system was shown to improve the identification results with a averaged TIC value smaller than 0.3 by 23 %.

5. Observer Kalman Filter Identification yielded accurate identified results during flight tests and is a feasible candidate for online near real-time system identification on a Small Unmanned Air Systems. Additionally, the results of the identified model that transferred to the Ground Control Station and shown to provide sufficient information for the ground control operator to make the model update decision.

7.2 Recommendations

- The current system is configured to apply sensed AOA α and sideslip β for system identification on medium to large sized UAV that are in the SUAS category. These measurements require a 5 hole probe with a data process system to provide accurate measurements. For SUAS with payloads less than 1 lbs or fixed wing Miniature Aerial Vehicles (MAV) that are not capable of carrying this system, an alternative is to estimate a derived angle-of-attack and sideslip angle directly from the lift coefficient C_L and sideforce coefficient C_Y as in Eq. (7.1):

$$\begin{aligned}\hat{\alpha} &= \frac{-\left(C_{L1} + C_{Lq}\frac{q\bar{c}}{2V_T} + C_{L\delta_E}\delta_E\right)\frac{\bar{q}S}{W} - n_z}{C_{L\alpha}\frac{\bar{q}S}{W} - n_x} \\ \hat{\beta} &= \frac{n_y\frac{W}{\bar{q}S} - C_{Yp}\frac{\gamma b}{2V_T} - C_{Y\delta_A}\delta_A - C_{Y\delta_R}\delta_R}{C_{Y\beta}}\end{aligned} \tag{7.1}$$

  Valasek et al. showed that using Monte Carlo experiments that the derived AOA and sideslip method provided accurate approximation even with high sensor noise [69]. A comparison between the results from the current system using measured AOA and a derived AOA method in flight test would be an interesting topic for future research.

- The latest version of the system still requires a human pilot to manually trim the aircraft before initiating the automated excitation. This requires an experienced pilot and allows very little tolerance on days with strong wind gusts. A reliable stabilize auto-trim feature could reduce the probability of identifying a model at a different orientation resulting in discrepancies between models.

- The process of providing a high quality LTI model has been thoroughly investigated in this thesis, however, the nominal model update procedure is still very preliminary, and also requires a lot of user experience. A more complex method of making the model update decision is to analyze the performance index. First, a performance index needs to be appointed according to the system control objective. Second, the identification system should be tightly integrated with the control subsystem. By analyzing the performance index constantly during flight, the model should only be updated when the system performance is degrading to an unacceptable level.

- Several wind gust identification models and research have been conducted in both the manned and unmanned research area, incorporating in knowledge of the wind gusts would increase the knowledge of the disturbances and could be a factor to increase the identification success rate.
- The near real-time identification framework enables the vehicle to continuously update a linear time invariant model that can be utilized in control design. The hardware settings are capable of adding in control design functions that enable implementation of control techniques such as dynamic inversion, model reference predictive control, and adaptive control to the system.
- The discrepancy between the assigned control input and the actual measured control input was noted but the effect on the identified results can be further analyzed. Items that can be analyzed include the reliability of the control implementation on the Pixhawk autopilots, the execution time lag on between each command signal, and the execution amplitude difference.

REFERENCES

[1] L. Meier, D. Honegger, and M. Pollefeys, “Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 6235–6240, IEEE, 2015.

[2] R. Mancuso, O. D. Dantsker, M. Caccamo, and M. S. Selig, “A low-power architecture for high frequency sensor acquisition in many-DOF UAVs,” *2014 ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2014*, pp. 103–114, 2014.

[3] D. M. Bushnell, “Scaling: wind tunnel to flight,” *The Annual Review of Fluid Dynamics*, vol. 38, pp. 111–121, 2006.

[4] J. S. Berndt, “Jsbsim: An open source flight dynamics model,” in *in C++. AIAA*, Citeseer, 2004.

[5] “SimGen - Computational Aerodynamics Software,” 2017. Web, accessed 30 August 2017.

[6] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” in *Field and Service Robotics*, 2017.

[7] “ArduPlane SITL - Ardupilot Dev Team,” 2016. Web, accessed 30 August 2017.

[8] A. Babushkin, “Jmavsim - pixhawk flight controller hardware project.” Web, asccessed 30 August 2017.

[9] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, pp. 2149–2154, IEEE.

[10] R. E. Kalman and R. S. . Bucy, “New results in linear filtering and prediction theory,” *Journal of Basic Engineering*, vol. 83, pp. 95–108, 1961.

[11] M. Hoshiya and E. Saito, “Structural identification by extended kalman filter,” *Journal of Engineering Mechanics*, vol. 110, pp. 1757–1770, 1984.

[12] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, 2000.

[13] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*. CRC press, 2011.

[14] E. A. Morelli, "System Identification Program for AirCraft (SIDPAC)," in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, (Monterrey, CA), Aug. 2002.

[15] D. J. Linse and R. F. Stengel, "Identification of aerodynamic coefficients using computational neural networks," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 6, pp. 1018–1025, 1993.

[16] V. Puttige and S. Anavatti, "Real-time neural network based online identification technique for a uav platform," in *2006 International Conference on Computation Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06)*, pp. 92–92, Nov. 2006.

[17] J.-N. Juang and R. S. Pappa, "An eigensystem realization algorithm for modal parameter identification and modal reduction," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 620–627, 1985.

[18] J.-N. Juang, M. Phan, L. G. Horta, and R. W. Longman, "Identification of observer/Kalman filter Markov parameters," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 2, pp. 320–329, 1993.

[19] M. B. Tischler and M. G. Cauffman, "Frequency-Response Method for Rotorcraft System Identification: Flight Applications to BO-105 Coupled Rotor/Fuselage Dynamics," *Journal of the American Helicopter Society*, vol. 37, no. 3, pp. 3–17, 1992.

[20] J.-N. Juang, *Applied System Identification*. Prentice Hall, 1994.

[21] J.-N. Juang and M. Phan, "Identification of System, Observer, and Controller from Closed-loop Experimental Data," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 1, pp. 91–96, 1994.

[22] P. G. Hamel and R. V. Jategaonkar, “Evolution of flight vehicle system identification,” *Journal of Aircraft*, vol. 33, no. 1, pp. 9–28, 1996.

[23] D. Chinarro, “System identification techniques,” in *System Engineering Applied to Fuenmayor Karst Aquifer (San Julián de Banzo, Huesca) and Collins Glacier (King George Island, Antarctica)*, pp. 11–51, Springer, 2014.

[24] J.-N. Juang and M. Q. Phan, *Identification and control of mechanical systems*. Cambridge University Press, 2001.

[25] R. Mehra, “Optimal inputs for linear system identification,” *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 192–200, 1974.

[26] E. A. Morelli, “Multiple input design for real-time parameter estimation in the frequency domain,” *IFAC Proceedings Volumes*, vol. 36, no. 16, pp. 639–644, 2003.

[27] E. A. Morelli, “Flight test maneuvers for efficient aerodynamic modeling,” *Journal of aircraft*, vol. 49, no. 6, pp. 1857–1867, 2012.

[28] M. Majji, *System identification: time varying and nonlinear methods*. Texas A&M University, 2009.

[29] S. Chen, S. Billings, and P. Grant, “Non-linear system identification using neural networks,” *International journal of control*, vol. 51, no. 6, pp. 1191–1214, 1990.

[30] F. Ahmed-Zaid, P. Ioannou, and Polycarpou, “Identification and control of aircraft dynamics using radial basis function networks,” in *Control Applications, 1993., Second IEEE Conference on*, 1993.

[31] P. Singla and J. L. Junkins, *Multi-resolution methods for modeling and control of dynamical systems*. CRC Press, 2008.

[32] S. Ross and J. A. Bagnell, “Agnostic system identification for model-based reinforcement learning,” *arXiv preprint arXiv:1203.1007*, 2012.

[33] A. H. McCoy, "Flight testing and real-time system identification analysis of a uh-60a black hawk helicopter with an instrumented external sling load," 1998.

[34] J. Cooper, "On-line version of the eigensystem realization algorithm using data correlations," *Journal of guidance, control, and dynamics*, vol. 20, no. 1, pp. 137–142, 1997.

[35] J. Valasek and W. Chen, "Observer/kalman filter identification for online system identification of aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 26, pp. 347–353, 2003.

[36] V. Klein and E. A. Morelli, *Aircraft System Identification: Theory and Practice*. American Institute of Aeronautics and Astronautics Reston, Va, USA, 2006.

[37] J. A. Grauer, "Real-time parameter estimation using output error," in *AIAA Atmospheric Flight Mechanics Conference*, p. 2556, 2014.

[38] V. R. Puttige and S. G. Anavatti, "Real-time system identification of unmanned aerial vehicles: A multi-network approach.," *JCP*, vol. 3, no. 7, pp. 31–38, 2008.

[39] W. DeBusk, E. Johnson, and G. Chowdhary, "Real-time system identification of a small multi-engine aircraft," in *AIAA Atmospheric Flight Mechanics Conference*, p. 5935, 2009.

[40] J. M. Brandon and E. A. Morelli, "Nonlinear aerodynamic modeling from flight data using advanced piloted maneuvers and fuzzy logic," 2012.

[41] E. A. Morelli, K. Cunningham, and M. A. Hill, "Global aerodynamic modeling for stall/upset recovery training using efficient piloted flight test techniques," in *AIAA Modeling and Simulation Technologies (MST) Conference*, p. 4976, 2013.

[42] E. A. Morelli, "Real-time global nonlinear aerodynamic modeling for learn-to-fly," in *AIAA Atmospheric Flight Mechanics Conference*, p. 2010, 2016.

[43] F. Arthurs, J. Valasek, and M. D. Zeigler, "Precision onboard small sensor system for unmanned air vehicle testing and control," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, (San Diego, CA), 2016.

[44] H.-H. Lu, J. Harris, V. G. Goecks, E. Bowden, and J. Valasek, "Flight test instrumentation system for small uas system identification," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pp. 1696–1705, IEEE, 2017.

[45] T. Woodbury, J. Valasek, and F. Arthurs, "Flight test results of observer/kalman filter identification of the pegasus unmanned vehicle," AIAA Paper 2015-1481, (Kissimmee, FL), AIAA Atmospheric Flight Mechanics Conference, Jan. 2015.

[46] J. Harris, J. Henrickson, F. Arthurs, and J. Valasek, "Aircraft System Identification using Artificial Neural Networks with Flight Test Data," in *Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS'16)*, (Arlington, VA), IEEE, June 2016.

[47] N. C. Rogers, Cameron and J. Valasek, "Heterogeneous multi-vehicle modular control framework with payload integration," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, IEEE, 2017.

[48] J.-N. Juang and M. Q. Phan, "System/observer/controller identification toolbox (socit)," tech. rep., NASA Technical Memorandum 107566, Feb 1992.

[49] J. Harris, V. G. Goecks, H.-H. Lu, and J. Valasek, "VSCL Developmental Flight Test Instrumentation," May 2017.

[50] G. D. Hanson and R. F. Stengel, "Effects of displacement and rate saturation on the control of statically unstable aircraft," *Journal of Guidance, Control, and Dynamics(ISSN 0731-5090)*, vol. 7, pp. 197–205, 1984.

[51] "MANUAL Mode - AruPilot Dev Team," 2016. Web, accessed 23 February 2017.

[52] D. T. McRuer, D. Graham, and I. Ashkenas, *Aircraft Dynamics and Automatic Control*. Princeton Legacy Library, Princeton, NJ: Princeton University Press, 2014.

[53] J. Roskam, *Airplane Flight Dynamics & Automatic Flight Controls: Part I*. DARCorporation, Jan. 2001.

[54] D. K. Schmidt, *Modern Flight Dynamics*. New York, NY: McGraw-Hill, 1 ed., 2010.

[55] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. Hoboken, NJ: John Wiley & Sons, Inc., 3 ed., 2016.

[56] D. F. Enns *et al.*, "Application of Multivariable Control Theory to Aircraft Control Laws," Tech. Rep. WL-TR-96-3099, Flight Dynamics Directorate, USAF Wright Laboratory, Wright-Patterson AFB, OH, May 1996.

[57] W. Durham, K. A. Bordignon, and R. Beck, *Aircraft Control Allocation*. Chichester, UK: John Wiley & Sons, Ltd., 2017.

[58] K. W. Iliff, "Parameter estimation for flight vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 12, no. 5, pp. 609–622, 1989.

[59] E. A. Morelli, "Estimating noise characteristics from flight test data using optimal fourier smoothing," *Journal of Aircraft*, vol. 32, no. 4, pp. 689–695, 1995.

[60] E. A. M. Jared A. Grauer and D. G. Murri, "Flight test techniques for quantifying pitch rate and angle of attack rate dependencies," in *AIAA Atmospheric Flight Mechanics Conference*, 2017.

[61] R. E. Maine and K. W. Iliff, "Maximum likelihood estimation of translational acceleration derivatives from flight data," *Journal of Aircraft*, vol. 16, no. 10, pp. 674–679, 1979.

[62] T. Woodbury, J. Valasek, and F. Arthurs, "Flight test results of observer/kalman filter identification of the pegasus unmanned vehicle," AIAA Paper 2015-1481, (Kissimmee, FL), AIAA Atmospheric Flight Mechanics Conference, Jan. 2015.

[63] M. Schroeder, "Synthesis of low-peak-factor signals and binary sequences with low autocorrelation (corresp.)," *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 85–89, 1970.

[64] R. W. Longman, M. Bergmann, and J.-N. Juang, "Variance and bias confidence criteria for era modal parameter identification.[eigensystem realization algorithm]," 1988.

[65] J.-N. Juang and R. S. Pappa, “An eigensystem realization algorithm for modal parameter identification and model reduction,” *Journal of Guidance*, vol. 8, no. 5, pp. 620–627, 1985.

[66] R. S. Pappa, K. B. Elliott, and A. Schenk, “Consistent-mode indicator for the eigensystem realization algorithm,” *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 5, pp. 852–858, 1993.

[67] R. Jategaonkar, *Flight vehicle system identification: a time domain methodology*, vol. 216. AIAA, Reston, VA, USA, 2006.

[68] R. F. Stengel and P. W. Berry, “Stability and control of maneuvering high-performance aircraft,” *Journal of Aircraft*, vol. 14, no. 8, pp. 787–794, 1977.

[69] J. Valasek, J. Harris, S. Pruchnicki, M. McCrink, J. Gregory, and D. G. Sizoo, “Characterization of derived angle-of-attack and sideslip angle algorithms using monte carlo and piloted simulation,” in *AIAA Atmospheric Flight Mechanics Conference*, p. 4059, 2017.

APPENDIX A

DATA LEAKAGE AND IDENTIFIED SYSTEM MATRICES

A.1 Data Leakage

Data leakage occurs when the input signal for forming the frequency response function is not periodic. Leakage of data results in loss of data information at the specific frequency where leakage is observed. The leakage problem can be shown in the following equation

$$\begin{aligned} e(k) &= \frac{1}{l}\sum_{i=0}^{l-1} u(i)e^{-j(\frac{2\pi k}{l})i} - \frac{1}{l}\sum_{i=-\mu}^{l-1-\mu} u(i)e^{-j(\frac{2\pi k}{l})i} \\ &= \frac{1}{l}\sum_{i=l-\mu}^{l-1} u(i)e^{-j(\frac{2\pi k}{l})i} - \frac{1}{l}\sum_{i=-\mu}^{-1} u(i)e^{-j(\frac{2\pi k}{l})i} \\ &= \frac{1}{l}\sum_{\tau=-\mu}^{-1} \left[u(\tau+l) - u(\tau)\right] e^{-j(\frac{2\pi k}{l})i} \end{aligned} \tag{A.1}$$

where $u(\tau + l) - u(\tau) = 0$ when the input is perfectly periodic.

A.2 Identified System Matrices

The identified individual lateral/directional and longitudinal model for the Hangar-9 1/4 Scale PA-18 Super Cub is Equation (A.2) and Equation (A.3). The combined full lateral/directional and longitudinal model is Equation (A.4).

Angular states are in radians, angular rates in radians per second, velocity is in meters per second, and controls in degrees. The lateral/directional model is trimmed at $\beta_1 = -6.75°$, $p_1 = 2.2\,°/\mathrm{s}$, $r_1 = -0.62\,°/\mathrm{s}$, and $\phi_1 = -0.59°$, where a ‘1’ subscript indicates a trim value.

$$\begin{Bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{Bmatrix} = \begin{bmatrix} 0.07918 & -0.1425 & -0.8387 & -0.414 \\ 4.81 & -7.098 & -3.568 & -2.693 \\ 3.444 & 4.548 & -1.98 & -0.8893 \\ -0.04679 & 0.9998 & -0.03553 & -0.02902 \end{bmatrix} \begin{Bmatrix} \beta \\ p \\ r \\ \phi \end{Bmatrix} + \begin{bmatrix} -0.002815 & 0.01296 \\ -0.666 & -0.2216 \\ 0.2464 & -0.5871 \\ -0.01386 & -0.005222 \end{bmatrix} \begin{Bmatrix} \delta_{\mathrm{A}} \\ \delta_{\mathrm{R}} \end{Bmatrix} \tag{A.2}$$

Equation (A.3) is trimmed at $V_{\mathrm{T}_1} = 17.96\,\mathrm{m/s}$, $\alpha_1 = 0.09°$, $q_1 = -0.74\,°/\mathrm{s}$, and $\theta_1 = 1.31°$.

Note that in Equation (A.3) the true airspeed V_{T} is substituted for body-axis x velocity u in Equation (4.6).

$$\begin{Bmatrix} \dot{V}_{\mathrm{T}} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{Bmatrix} = \begin{bmatrix} -0.4541 & -2.628 & 1.806 & -7.129 \\ -0.0851 & -2.468 & 1.788 & 0.1256 \\ 0.2701 & -5.163 & -7.527 & 1.255 \\ 0 & -0.2657 & 0.9126 & 0.3046 \end{bmatrix} \begin{Bmatrix} V_{\mathrm{T}} \\ \alpha \\ q \\ \theta \end{Bmatrix} + \begin{bmatrix} 0 & 0.02639 \\ 0 & 0.05085 \\ -0.0001826 & -1.417 \\ 0 & -0.01782 \end{bmatrix} \begin{Bmatrix} \delta_{\mathrm{T}} \\ \delta_{\mathrm{E}} \end{Bmatrix} \tag{A.3}$$

$$\begin{bmatrix} \dot{V}_{\mathrm{T}} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0.35 & 21.42 & 1.213 & -10.43 & -13.85 & 3.731 & 3.668 & 5.348 \\ -0.078 & -1.892 & 1.769 & 0.092 & 0.783 & -0.672 & 0.016 & 0.549 \\ 0.242 & -16.74 & -8.09 & -0.506 & -5.946 & 1.921 & 0.507 & -2.755 \\ 0.017 & 0.601 & 0.609 & -0.282 & -0.129 & 0.413 & -0.29 & 0.401 \\ -0.013 & 0.506 & -0.068 & -0.239 & 0.376 & 0.083 & -1.16 & -0.308 \\ -0.395 & 0.157 & 4.778 & 2.027 & 3.84 & -7.848 & -2.151 & -5.991 \\ -0.197 & -3.526 & -4.581 & -1.451 & 14.5 & 1.071 & -3.975 & 0.625 \\ 0.0006 & 0.051 & 0.079 & -0.214 & -0.034 & 1.131 & 0.317 & 0.182 \end{bmatrix} \left\{ \begin{matrix} V_{\mathrm{T}} \\ \alpha \\ q \\ \theta \\ \beta \\ p \\ r \\ \phi \end{matrix} \right\} \tag{A.4}$$

$$+ \begin{bmatrix} 0.0002 & 19.83 & -10.92 & 2.486 \\ -1.068e-05 & -1.827 & 0.919 & -0.4767 \\ 2.956e-05 & -6.455 & 3.168 & -1.71 \\ 9.704e-06 & -2.666 & 1.631 & -0.07735 \\ 1.468e-05 & 1.559 & -0.4864 & 1.037 \\ 0.0001101 & 23.23 & -15 & -0.1386 \\ 5.53e-05 & -11.29 & 0.7215 & -14.03 \\ 8.609e-06 & 0.5379 & -0.5072 & -0.4112 \end{bmatrix} \left\{ \begin{matrix} \delta_{\mathrm{T}} \\ \delta_{\mathrm{E}} \\ \delta_{\mathrm{A}} \\ \delta_{\mathrm{R}} \end{matrix} \right\}$$

Equation (A.4) is trimmed at $V_{\mathrm{T}_1} = 16.8\,\mathrm{m/s}$, $\alpha_1 = 0.74°$, $q_1 = -2.77\,°/\mathrm{s}$, $\theta_1 = -12.2°$, $\beta_1 = -9.9°$, $p_1 = 6.16\,°/\mathrm{s}$, $r_1 = -2.28\,°/\mathrm{s}$, and $\phi_1 = -1.07°$. The aircraft mass is $20.5lbs$, and the C.G. location is approximately 1.6 ft measured from the tip of the propeller shaft.

APPENDIX B

SYSTEM DESIGN CONCEPTS

B.1 Concept of Operations

The system identification system shall be able to perform near real-time system identification using onboard calculations only. The system should be able to perform both lateral and longitudinal identification and provide real-time feedback to the ground control user on the ground control station. For comparison, the system shall be able to log sensed data, identified results, and identified nominal models for off-line analysis.

B.2 Design Requirements

To achieve the research objectives and overcome the research challenges, certain requirements should be met for the data acquisition subsystem and system identification subsystem:

1. The system shall be capable of logging aircraft states at a high rate.
    1.1. The system should be capable of logging aircraft states and control effector positions at rates no less than 100 Hz.
2. The system shall be capable of directly measure control surface deflections.
3. The system should have a reliable excitation method capable of exciting all the dynamic modes.
    3.1. The auto excitation should be commanded by the ground control operator.
    3.2. The ground control operator should be able to terminate an excitation
    3.3. The pilot should still have control authority when excitation is executed
    3.4. The system shall be able to perform doublet, 3-2-1-1, and sine sweep excitations on multiple control surfaces
4. The system shall be able to identify local linear models during flight.
    4.1. The system should be capable of updating the identified local models every specified time period.
        4.1.1. The ground control operator should be able to assign and modify the specified update time.

    4.2. The identification algorithm should check the stability of the identified system
    4.3. The identification algorithm shall calculate identification quality indexes
        4.3.1. Quality indexes should provide clear coherence percentage of the identified model

5. The on-board system should be able to pass the identified parameters through telemetry radios to the ground control station.
6. The update procedure shall be a human-in-the-loop system with the ground control operator as the final decision maker.
    6.1. The ground control operator should be provided with updated quality indexes compared with previous model
    6.2. The ground control operator should be able to decide whether to update the identified model on the aircraft through a GUI interface.
    6.3. The system should be able to switch back to a nominal model through the telemetry radio when the ground control operator observe an issue with the new identified model
        6.3.1. If a faulty identified model was chosen and updated, the system should be able to self check and restore the previous nominal model
7. The system should be modular for extensive capabilities
    7.1. The system shall be coded in Python
    7.2. The system should use COTS components as much as possible
    7.3. The system should have a modular software architecture and be highly extensible
    7.4. The system should work with any sensor as long as the data is provided in the a similar format