DESIGN AND EVALUATION OF UAV GROUND CONTROL STATION BY CONSIDERING HUMAN FACTORS STANDARDS AND GUIDELINES

by

Taiwo Amida

Bachelor of Engineering, Shizuoka Institute of Science and Technology (2015)

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the program of

Aerospace Engineering

Toronto, Ontario, Canada, 2018

©Taiwo Amida, 2018

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

DESIGN AND EVALUATION OF UAV GROUND CONTROL STATION BY CONSIDERING HUMAN FACTORS STANDARDS AND GUIDELINES

Taiwo Amida

Master of Applied Science, Aerospace Engineering, Ryerson University, Toronto (2018)

ABSTRACT

The majority of Unmanned Aerial Vehicle (UAV) accidents can be directly related to human error. For this reason, standards and guidelines focusing on human factors have been published by various organizations such as Transport Canada, FAA, EASA, NASA and military agencies. The objective of this thesis is to present a methodology for designing a Ground Control Station (GCS) using available standards and guidelines for human factors. During the design process, a detailed analysis was performed using human factors methods to ensure all requirements were met; each phase of the design follows the guidelines presented in the compiled human factors standards and guidelines. The GCS interface was developed using advanced programming techniques and commercial off-the-shelf software. Moreover, an operator workload evaluation was carried out using NASA task load index for validation of design methodology. It was found that the applied methodology not only improved the pilot workload, but also ensured that all user and stakeholders' requirements are met.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my advisor, Dr. Joon Chung, for all his guidance, support, and assistance throughout my master's degree program, for providing me with the opportunity to have industrial experience while simultaneously completing my research, and for making me think outside my self-imposed boundaries, over these two years. I would also like to thank Stephane Roy, Eric Simon and the Presagis Team for their time and patience throughout my internship period at CAE (Presagis). This research would have been impossible without their help. To my friend and colleague, Jeffrey Haber, thank you for your inspiration and mentoring during my first months at the MIMS laboratory. Michel Patenaude and Alexander Genio, thank you for your support throughout this project. Lastly, I would like to thank my family for their prayers and financial assistance throughout my education. Most importantly, Nas: Nosiru Amida. Thank you for believing in me during my education in Japan.

TABLE OF CONTENTS

AUTHOR'S DECI	LARATION	ii
ABSTRACT		iii
ACKNOWLEDGE	EMENTS	iv
LIST OF TABLES		x
LIST OF FIGURE	S	xi
LIST OF ACRONY	YMS	xv
CHAPTER 1: IN	NTRODUCTION	1
1.1. Unm	anned Aerial Vehicles (UAV)	1
1.1.1.	Definition of UAV	1
1.1.2.	Classification of UAV	2
1.1.3.	History of UAV	6
1.1.4.	History of Flight Modelling and Simulation for Training	
1.1.5.	Visual System and Terrain Modelling for Simulation	14
1.1.6.	UAV Flight Simulation	
1.2. Hum	an Factors Methods	16
1.2.1.	Human Factors	
1.2.2.	Human-Centered Design	
1.2.3.	Hierarchical Task Analysis Method (HTA)	19
1.3. Tand	lem Rotor UAV Overview	

1.4. R	esearch Tools	23
1.4.	1. Presagis HeliSIM	23
1.4.	2. Presagis VAPS-XT	23
1.4.	.3. Lightweight Communications and Marshalling (LCM)	24
1.5. R	yerson Reconfigurable GCS	24
1.6. R	esearch Objectives and Constraints	26
CHAPTER 2:	USER REQUIREMENTS	28
2.1. U	ser Domain Description	29
2.2. U	ser Goals, Tasks, and actions	33
2.3. T	andem Rotor UAV User Goals, Task, and actions	34
2.4. H	lierarchical Task Analysis (HTA) of Tandem Rotor UAV	36
2.5. Ir	iitial Design Requirements	48
2.6. H	uman Factors Standards and guidelines	50
CHAPTER 3:	PRELIMINARY DESIGN AND SET-UP	52
3.1. P	reliminary Design	52
3.2. D	Design Iterations	53
3.3. S	et-Up	55
CHAPTER 4:	GCS CONFIGURATION	59
4.1.	GCS Screen Configuration Design	59
4.2.	Design of Quick Swap Gesture	61

4.2.1.	Idle State	
4.2.2.	Idle State to Dragging State Transition	
4.2.3.	Dragging State	
4.2.4.	Dragging State-Throwing State Transition	
4.2.5.	Throwing State	
4.2.6.	Double Tap Feature	
4.3. Q	Quick Swap Gesture Integration	
CHAPTER 5: G	GCS INSTRUMENT PANELS DESIGN	
5.1. P	rimary Instrument Panels	
5.1.1.	Primary Flight Display (PFD)	
5.1.2.	Mission	
5.1.3.	Map	
5.1.4.	3D Heading Display	
5.1.5.	Connection Status	
5.1.6.	Automatic Flight Control System (AFCS)	
5.1.7.	Simple and Detailed Engine Information	
5.1.8.	Message Alerts	
5.2. S	econdary Instrument Panels	
5.2.1.	Obstacle Visualization	
5.2.2.	Message Log	

	5.2.3	3. Radar	96
	5.2.4	4. Camera	97
СНАРТЕН	R 6:	DESIGN EVALUATION AND RESULTS	99
(6.1.	GCS Requirements Reevaluation	99
(6.2.	Evaluation Test Set-Up	100
(6.3.	NASA TLX Results	103
(6.4.	Compliance Rating Results	111
CHAPTER	R 7:	CONCLUSION AND FUTURE WORKS	113
,	7.1.	Conclusion	113
,	7.2.	Future Works	114
APPENDI	IX		116
	APPEN	NDIX A - Compiled Human Factors Design Standards and Guidelines.	116
	APPEN	NDIX B - Compliance Rating Results	142
	APPEN	NDIX C - Participants Consent Form	145
	APPEN	NDIX D - In-Person Recruitment Transcript	148
	APPEN	NDIX E - Mail Recruitment Transcript	149
	APPEN	NDIX F - Participants Worksheet	150
	APPEN	NDIX G - Additional Questionnaire	152
	APPEN	NDIX H - User Requirements and Grouping	153
1	APPEN	NDIX I - Drag Limits Definition Vap-XT Code	156

	APPENDIX J - Panels Swap VAPS-XT Code	157
	APPENDIX K - Demo Mission XML File	159
REFERE	ENCES	160

LIST OF TABLES

Table 1 - DARPA design requirements for MAV[4]	
Table 2 - Classification of UAV (EUROUVS) [3]	5
Table 3 - User Characteristics	
Table 4 - Variables Definitions for Quick Swap Gesture	
Table 5 - Data Objects for Quick Swap Gesture Integration	
Table 6 - Waypoint and Loiter Data Identifier	
Table 7 - Experiment Participants Profile	101
Table 8 - Sources of Load Results	
Table 9 - Magnitude of Load Results	

LIST OF FIGURES

Figure 1 - MQ-1 Predator (Left) [5] and The Global Hawk (Right) [6]	4
Figure 2 - Elmer Sperry [8]	6
Figure 3 - Aerial Torpedo on the Rails[9]	7
Figure 4 - Kattering Bug[8]	8
Figure 5 - The Oeminchen No.2 [12]	9
Figure 6 - The Convertawings Model-A Quadcopter [14]	10
Figure 7 - The Curtis VZ-7 [16]	11
Figure 8 - Early flight simulator [19]	12
Figure 9 - The Link Trainer [18]	13
Figure 10 - Redifon Visual Terrain System [19]	15
Figure 11 - Human-Centered Design Plan for GCS Design	18
Figure 12 - Hierarchical Task Analysis Process Flowchart[29]	20
Figure 13 - Boeing CH-47 Chinook [31]	21
Figure 14 - Simulator Cockpit designed by Gulfstream using VAPS-XT [35]	
Figure 15 - Ryerson MIMS Laboratory	25
Figure 16 - Reconfigurable Feature of Ryerson GCS [37]	
Figure 17 - User Domain Model of Tandem Rotor UAV	32
Figure 18 - Relationship between Goal, Tasks, and actions	33
Figure 19 - User Goals for Tandem Rotor UAV	35
Figure 20 - Goal and Sub-goals of Tandem Rotor UAV	36
Figure 21 - HTA Result for Executing Manual Take-off	38
Figure 22 - HTA Result for Executing Manual Landing	

Figure 23 - HTA Result for Executing Automatic Take-off	40
Figure 24 - HTA Result for Executing Automatic Landing	41
Figure 25 - HTA result for Executing Automatic Obstacle Avoidance	42
Figure 26 - HTA Result for UAV Status	43
Figure 27 - HTA Result for Camera Tracking	44
Figure 28 - HTA Result for RADAR Control	45
Figure 29 - HTA Result for Waypoints Creation	46
Figure 30 - HTA result for Executing Flight Plan	47
Figure 31 - Venom GCS Developed by RAAS [49]	52
Figure 32 - First Design Iteration [51]	54
Figure 33 - Second Design Iteration [52]	55
Figure 34 - Final Design [52]	55
Figure 35 - Pilot Workstation [52]	56
Figure 36 - Payload Workstation [52]	58
Figure 37 - CH Products USB Desktop Joystick	58
Figure 38 - Start Screen [52]	60
Figure 39 - Dual Screen Configuration [52]	60
Figure 40 - Primary and Secondary Screen [52]	60
Figure 41 - Quick Swapping the PFD and Map [52]	61
Figure 42 - Final Position After Panels Have Been Swapped [52]	61
Figure 43 - Lock Icon [52]	61
Figure 44 - State Chart Diagram for Quick Swap Gesture [52]	64
Figure 45 - Relationship between the dragLimits, ExtentGroup and the container	67

Figure 46 -	Maximized Primary Panels During Automated flight	68
Figure 47 -	Primary Flight Display [52]	71
Figure 48 -	Mission Panel [52]	73
Figure 49 -	Pop-Up Bar for Loading Mission Files [52]	74
Figure 50 -	File Name Input Screen [52]	74
Figure 51 -	2D Map View [52]	77
Figure 52 -	3D Map View [52]	78
Figure 53 -	Waypoint/Loiter menu [52]	78
Figure 54 -	Touch Gesture for Changing Map Scale [52]	79
Figure 55 -	3D heading Display Panel [52]	80
Figure 56 -	Connection Status Panel [52]	81
Figure 57 -	Automatic Flight Control System Panel [52]	82
Figure 58 -	Engine Value Behavior Procedure	85
Figure 59 -	Detailed Engine Information [52]	85
Figure 60 -	Simple Engine Information [52]	86
Figure 61 -	Message Alert Panel [52]	86
Figure 62 -	Process for Displaying Sensor Information	88
Figure 63 -	GCS C++ Plugin Procedure for Displaying Sensor Points	90
Figure 64 -	Data Point X, Y and Z coordinates [58]	91
Figure 65 -	LiDAR 2 Points [52]	91
Figure 66 -	LiDAR 1 and LEDDAR 1 Points [52]	92
Figure 67 -	Log Messages Creation Procedure	94
Figure 68 -	Message Log Panel [52]	96

Figure 69 -	RADAR Instrument Panel [52]	. 97
Figure 70 -	Camera Panel [52]	. 98
Figure 71 -	GCS Complete Set-up [52]	102
Figure 72 -	Average Source of Load	106
Figure 73 -	Average Source of Workload for Inexperienced Pilots	106
Figure 74 -	Average Source of Workload for experienced Pilots	107
Figure 75 -	Average Magnitude of Load	108
Figure 76 -	Average Magnitude of Load for Inexperienced Pilots	108
Figure 77 –	Average Magnitude of Load for experienced Pilots	109
Figure 78 -	Average Adjusted rating	110
Figure 79 -	Overall Workload Rating	111
Figure 80 -	Normalized Overall Workload Rating	111
Figure 81 -	Compliance Rating Results	112

LIST OF ACRONYMS

Attitude Direction Indicator
Automatic Flight Control System
Application Programming Interface
Arcade
Auto Throttle System
Computer Image Generation
Computer in Control
Commercial-Off-The-Shelf
Close Range
Defense Advanced Research Agency
Document Object Model
Endurance
European Association of Unmanned Vehicle System
Federal Aviation Administration
Flight Management System
Field of View
Ground Control Station
Global Positioning System
Ground Speed
High Altitude Long Endurance
Human Centered Design
Human Factors Design Standards
Human Machine Interface

HTA	Hierarchical Task Analysis
IAS	Indicated Airspeed
IMU	Inertia Measurement Unit
ISAR	Inverse Synthetic Aperture RADAR
ISO	International Standards Organization
LCM	Lightweight Communications and Marshalling
LEDDAR	Light Emitting Diode Detection and Ranging
LiDAR	Light Detection and Ranging
MALE	Medium Altitude Long Endurance
MAV	Micro Air Vehicle
MIL-HDBK-759C	Handbook of Human Engineering Design Guidelines
MIMS	Mixed-Reality Immersive Motion Simulation
MR	Medium Range
MTI	Moving Target Indicator
MTOW	Maximum Take-Off Weight
M&S	Modelling and Simulation
NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization
PFD	Primary Flight Display
PIC	Pilot in Control
RADAR	Radio Detecting and Ranging
RBGM	Real Beam Ground Map
RC	Radio Controlled
RPM	Rotations per Minute
SA	Situational Awareness

SpotSAR	Synthetic Aperture RADAR			
SR	Short Range			
StripSAR	Strip-Mapping Synthetic Aperture RADAR			
TAS	rue Airspeed			
TLX	Task Load Index			
UAS	nmanned Aerial System			
UAV	Unmanned Aerial Vehicle			
UDOFT	niversal Digital Operational Flight Trainer			
UDP	User Datagram Protocol			
UI	User Interface			
USAF	United States Air Force			
VTOL	Vertical Take-Off and Landing			
WS	Wind Speed			
WX	Weather			
XML	Extensible Markup Language			

CHAPTER 1: INTRODUCTION

1.1. UNMANNED AERIAL VEHICLES (UAV)

1.1.1. Definition of UAV

Unmanned Aerial Vehicles (UAVs) are powered, fixed or rotary-wing aerial vehicles with no onboard human pilot. They can be autonomous, semi-autonomous, remotely controlled or have a combination of these capabilities. They can be used for a wide variety of civilian and military applications such as weather monitoring, military, traffic control, transportation, search and rescue, and communication. Also, UAVs can be equipped with various types of sensors and payloads, making them acceptable for performing specific tasks which are typically considered dangerous for human-piloted aircraft [1].

The term "unmanned" is usually ambiguous and falsely considered to mean that the aircraft is flown with no human pilot. However, the difference between UAVs and manned aircraft is only limited to the location of the pilot while controlling the aircraft. A pilot of the manned aircraft is usually located onboard while controlling the aircraft. However, a UAV pilot controls the aircraft remotely using a simple radio controller or an advanced ground control station (GCS) linked to the aircraft via suitable communication links.

The terms UAV, Unmanned Aerial System (UAS), and drones are sometimes used interchangeably to refer to the same thing. However, to avoid any form of ambiguity, efforts will be made to clarify the difference between these terms by defining each one.

From its name, it can be inferred that the term "UAS" refers to a system. By applying common sense, it can be concluded that the UAS is not a single element or device but a system

comprised of different elements. In his book, Reg Austin stated that: "The system [UAS] includes a number of sub-systems which include the aircraft (often referred to as a UAV or unmanned air vehicle), its payloads, the control station(s) (and, often, other remote stations), aircraft launch and recovery sub-systems where applicable, support subsystems, communication subsystems, transport sub-systems, etc." [2]

UAV, on the other hand, is a standalone fixed or rotary-wing aircraft that is part of the UAS. The degree of intelligence and automation of a single UAV varies. Some are equipped with the capability of taking corrective actions when there is an unplanned error in the system while others are programmed to return to the station in a similar case.

Finally, the most commonly used term—"drones"—are unmanned fixed or rotary-wing aircraft flown utilizing pre-programmed mission and course with negligible intelligence [2]. The major difference between drones and UAVs is only limited to the level of intelligence and the complexity of the mission they can carry out. Also, UAVs can communicate and transfer an enormous amount of data and results in real time with the control station. However, the drones will be required to complete the mission before the data it has obtained can be retrieved [2].

Based on these definitions, it can be understood that the highly intelligent category of unmanned aircraft and system is mainly considered in this thesis. Thus, the term "drone" will rarely be used. The terms UAV and UAS will be used more throughout this research.

1.1.2. Classification of UAV

Numerous organizations and authors have proposed various methods of classifying UAVs. However, the classification used throughout this thesis was suggested by the European Association of Unmanned Vehicles System (EUROUVS). The EUROUVS classified UAV based on parameters such as flight altitude, endurance, speed, maximum take-off weight (MTOW) and size. It must be stated that EUROUVS did not create these classifications for certification purposes, but rather for the compilation of a universal UAV catalog [3].

The first category is the micro- and mini-UAVs. Micro-Air Vehicles (MAV) are small in size, and can only be flown within 300 meters [3]. They are mostly used for photography and recording purposes. For better clarification of this group, limitations were set for designers of the MAV by the Defense Advanced Research Projects Agency (DARPA) [4]. Table 1 shows the requirements for the MAV design.

Specifications	Requirements			
Size	<15.24 cm Maximum dimension			
Weight	~100 g Objective gross takeoff weights (GTOW)			
Range	1 to 10 km Operational range)			
Endurance	60 min Loiter time on station			
Altitude	<150 m Operational ceiling			
Speed	15 m/s Maximum flight speed			
Payload	20 g Mission dependent			
Cost	\$1500 Maximum cost			

Table 1 - DARPA design requirements for MAV[4]

On the other hand, the mini-UAVs have a weight restriction of about 30 kilograms, and the altitude at which they can be flown is limited to between 150 and 300 meters. Mini UAVs have double endurance when compared with MAVs (about 2 hours). Also, they are much more popular, especially the rotary wing types with vertical take-off and landing (VTOL) capability [3].

The next category is the tactical UAVs which are known for possessing large platforms flying at a maximum altitude of 8,000 meters. In contrast to civil/commercial applications of the micro and mini UAVs, tactical UAVs are used for serious businesses, such as military missions

and rescue operations. This category can further be subdivided into close range, short range, medium range, long range, endurance and medium altitude long endurance (MALE) UAVs [3]. The most popular in this group is the MQ-1 Predator (shown in Figure 1) designed by the United States (U.S.) General Atomic Aeronautical System. The Predator is a MALE UAV that can operate for about 40 hours nonstop at a maximum range of about 3,704 kilometers. Furthermore, it can be equipped with various payloads such as precision missiles and an infrared tracking device.



Figure 1 - MQ-1 Predator (Left) [5] and The Global Hawk (Right) [6] The final category of UAV is the strategic UAVs. These UAVs are well known for their high altitude, long endurance, and long range performance. For this reason, they are mostly known as high altitude long endurance (HALE) UAVs. Their MTOW could vary from 2,500 kilograms to 12,000 kilograms and can reach a maximum altitude of about 20,000 meters. The GCS of the HALE UAVs is usually complex and highly detailed compared to other categories of UAVs. A perfect example of a military HALE UAV is the Global Hawk (Figure 1), which has an endurance of about 35 hours and can reach a maximum altitude of about 18,000 meters [3, 7].

Table 2 shown below presents the various categories of UAV listed above with examples and missions they carry out.

UAV	Category (Acronym)	MTO W	Maximum Flight	Endurance (hours)	Data Link Range	Missions	Systems
	× • /	(kg)	Altitude(m)	、 <i>、</i>	(Km)		·
Micro /Mini UAVs	Micro (MAV)	0.10	250	1	<10	Scouting, National Broadcasting Company (NBC) Sampling, Surveillance inside buildings	Black Widow, Microstar, Microbat, Fancopter, QuattroCopter, Mosquito, Hornet, Mite
	Mini	<30	150-300	<2	<10	Film and broadcast industries, agriculture, pollution measurement, surveillance inside building, communication relay Electronic Warfare (EW)	Mikado, Aladin, Tracker, DragonEye, Raven, Pointer II, Carolo C40/P50, Skorpio, R-Max and R-50, RoboCopter, YH-300SL
Tactical UAVs	Close Range (CR)	150	3000	2-4	10-30	Reconnaissance, Surveillance, and Target Acquisition (RSTA), mine detector, search & rescue, EW	Observer I, Phantom, Copter 4, Mikado, Robocopter 300, Pointer, Camcopter, Aerial and Agriculture RMax
	Short Range (SR)	200	3000	3-6	30-70	Bomb damage assessment (BDA), RSTA, EW, mine detection	Scorpio 6/30, Luna, SilverFox, EyeView, Firebird, R-Max Agri/Photo, Hornet, Raven, Phantom, GoldenEye 100, Flyrt, Neptune
	Medium Range (MR)	150- 500	3000-5000	6-10	70-200	BDA, RSTA, EW, mine detection, NBC sampling	Hunter B, Mucke, Aerostar, Sniper, Falco, Armor X7, Smart UAV, UCAR, Eagle Eye+, Alice, Extender, Shadow 200/400
	Long Range (LR)	-	5000	6-13	200-500	RSTA, BDA, communications relay	Hunter, Vigilante 502
	Endurance (EN)	500- 1500	5000-8000	12-24	>500	BDA, RSTA, EW, communication relay, NBC sampling	Aerosonde, Vulture II Exp, Shadow 600, Searcher II, Hermes 45OS/45OT/700
	Medium Altitude, Long Endurance (MALE)	1000- 1500	5000-8000	24-48	>500	BDA, RSTA, EW weapons delivery, communications relay, NBC sampling	Skyforce, Hermes 1500, Heron TP, MQ-1 Predator, Predator-IT, Eagle ¹ / ₂ , Darkstar, E- Hunter, Dominator
Strategic UAVs	High Altitude, Long Endurance (HALE)	2500- 12500	15000-20000	24-48	>2000	BDA, RSTA, EW, communications relay, boost phase intercept launch vehicle, airport security	Global Hawk, Raptor, Condor, Theseus, Helios, Predator B/C, Libellule, EuroHawk, Mercator, SensorCraft, Global Observer, Pathfinder Plus

Table 2 - Classification of UAV (EUROUVS) [3]

1.1.3. History of UAV

Most people imagined UAV to be the fascinating technology that was invented in recent years. Truly, they are fascinating, and the majority of the technologies such as real-time imaging, search and rescue missions were all discovered over the past few years; however, many will be shocked when they learn that the development of UAV itself began about eight years after the Wright brothers attempted their first flight. During the WWI era, in 1911, Elmer Sperry, the inventor of the gyroscope, had become keenly interested in autonomous control using radio control and gyroscope. During a three-month period after receiving financial support from the Navy in 1913, he oversaw 58 flight tests of aircraft flying with only a specialized gyroscope at the controls [8]. The historical image of Elmer Sperry is shown in Figure 2 below.



Figure 2 - Elmer Sperry [8]

About five years after Elmer started this project, the genius behind the development of Norden bombsight of WWII, Carl Norden, and Glenn Curtiss joined Elmer's team and developed the concept of aerial torpedo [8]. The aerial torpedo was well funded, and several efforts were made to incorporate radio devices in the aerial torpedo. However, after several attempts, they realized that the radio technology was not matured enough for use in the aerial torpedo. Subsequent research and tests focused more on maintaining course and measuring the distance to targets [8]. Figure 3 below shows the aerial torpedo with the weight of about 950 pounds running on 90 Hp Curtiss OX-5 engine. It possesses a wingspan of about 22 inches and can attain a top speed of 70 MPH [9]



Figure 3 - Aerial Torpedo on the Rails[9]

Even though the aerial torpedo never witnessed wartime, the idea paved the way for several inventions. In 1917, Charles Kettering designed an unmanned flying bomb named Kettering bug (as shown in Figure 4 below) that could precisely hit a target from mid-range. The design was brought to life with the help of reputable companies, such as Ford Motors, that acted as consultants for the bug's engine design.



Figure 4 - Kattering Bug[8]

After WWI ended, all tests and experiments with the unmanned aircraft and flying bombs came to a halt [9]. However, it did not prevent enthusiastic researchers from designing radiocontrolled aircraft. The military also conducted more tests on the bug and started sponsoring similar projects which led to the development of the Curtiss F-5L aircraft which became the first unmanned aircraft to be flown successfully through all phases of flight.

Fast forward to our modern era research: simple radio control has shifted to advanced satellite control for advanced UAVs, which includes features such as weather monitoring, live surveillance, and stealth. However, over the past five years, model micro unmanned air vehicles¹ and simple drones such as the DJI Phantom drones have become extremely popular with a rapidly growing market, which are projected to triple in the next two or three years [8]. Note that these projections are made despite regulatory agencies still debating regulations to set.

¹ Model UAV refers to UAVs that can only be flown within the sight of the operator

As for rotary-wing UAVs, their idea has existed since the 1920's. Interestingly, some working models were built, and flight tested; however, for the most part, rotary-wing UAV designs stayed on paper. The reason was attributed mostly to complexity, weight, and the interactions between multiple rotor blades in proximity to each other [10], [11]. Credit for the first working aircraft with four rotors goes to Etienne Oemichen in 1923, at the time a young engineer with the Peugeot motor car company [12]. In fact, the aptly named Oemichen No. 2 (shown in Figure 5 below) broke helicopter world records at the time and had over 1,000 successful test flights [13]. It is important to note that the rotors of the Oemichen No. 2 were all driven by the same engine, thus they cannot be compared to modern-day UAVs with multiple propellers. It was impressive nonetheless.



Figure 5 - The Oeminchen No.2 [12]

Over the years, the helicopter evolved to remain a one-rotor aircraft. However, some more fully-sized multi-rotor aircraft flew before it. In 1956, the Convertawings Model A Quadcopter, designed by Dr. George de Bothezat and Ivan Jerome, was the first multi-rotor aircraft to have control by varying the throttles of individual rotors. The aircraft was a continuation of the Oeminchen No. 2, as Bothezat had some contribution to it [14]. A picture of the Convertawings Model A Quadcopter can be seen in Figure 6 below.



Figure 6 - The Convertawings Model-A Quadcopter [14]

In the end, the Model-A did not develop further as the pilot was required to control each rotor himself; one can imagine how monumental and terrifying this task would be for even a seasoned pilot. Interestingly, the Curtis aircraft company worked on a quadcopter for the U.S. Army as well, the Curtis Wright VZ-7 (see Figure 7 below). Despite it overcoming control issues associated with multi rotors and being easy to fly, it did not meet United States Air Force (USAF) standards and had its program canceled [15].

The reasons for all these multi -rotor aircraft not getting past the prototype stage are many. Aerodynamic effects, control issues, and noise (in particular for the Air Force) are just some of the problems one could think about running into when designing a quad copter platform. The other reason is simply performance competition from fixed wing aircraft for benchmarks such as speed, payload capacity, range, endurance, and so forth.



Figure 7 - The Curtis VZ-7 [16]

However, no one could have imagined how far aircraft would get into our daily lives a century ago. Aside from air travel not being a luxury anymore, the technology era is on the eve of having drones deliver packages right to one's doorstep [17], and it is the multi-rotor concept that is making it all happen. The quadcopter went from seeming a failure to a critical platform for innovation in a quickly growing market. If anything, the history of the multi-rotor is a standing monument to how a crazy idea can evolve to fuel big changes.

1.1.4. History of Flight Modelling and Simulation for Training

Over the past years, the control of aircraft has been advancing to the point that it has become impossible to fly them successfully without an adequate amount of training. Flight simulation has been one of the major technologies used to train a pilot before flying the real aircraft. A flight simulator is a machine that behaves like a real aircraft by simulating the flight motion, aircraft controls and instruments, and the terrain on which the flight takes place. The main reason that the aviation industries have focused on this technology is that it reduces the cost of training by allowing the trainee to master their flying skills without flying the actual aircraft. Also, it guarantees the safety of the pilot while simultaneously giving them the chance to improve their decision-making ability in an emergency situation without risking their life [18].

The very first synthetic flight simulator was developed in 1910 (Figure 8). It consisted of two half-sectioned barrels constructed in a way that looked similar to an aircraft. The structure can be adjusted manually to simulate pitch and roll, and the pilot was required to line up the reference bar with the horizon [19].



Figure 8 - Early flight simulator [19]

After about a decade, engineers started exploring the creation of a simulator using machines with mechanical and electrical actuators linked to the controls. In 1917, for the first time, a pivoted fuselage that simulates the aircraft pitch, roll and yaw was developed. The flight motion of the device was generated using compressed air motors which felt like the real aircraft. This device was later improved to become electrical driven, and the most successful version was the Link trainer, as shown in Figure 9 below. The Link trainer utilized an electrically driven suction pump in a fixed base that extends to the control valves and can be controlled using a stick and rudder, while another mechanically driven motor generates motion to simulate attitude

disturbance that occurs during flight. It was later equipped with a radio signal to support blind flying training by simulating the behavior of radio navigation.

However, despite all the time inventors and engineers have put into developing a reliable flight simulator for training, the aviation industry never accepted it as a real substitute for actual flight. They had to wait for the gradual growth of the field [19].



Figure 9 - The Link Trainer [18]

At the start of WWII, the need to train a large number of pilots rejuvenated the research on flight simulation. Fortunately for the inventors, the analog computers were invented during this time, too, which made it possible for engineers to integrate flight equations of motion into the simulator. In 1943, after the war, Curtiss-Wright constructed the first full flight simulator for the Boeing 377 Stratocruiser. The achievement stirred a response from Link Company, and they also developed their full simulator for the C-11 Jet in 1949. After some time, the large analog simulators reached their limit, and it became evident that achieving a higher fidelity and reliability, as demanded, is impossible. However, after the introduction of the second-generation digital computers, shifting to digital flight simulation became inevitable. The University of Pennsylvania became the first set to take advantage of the new trend by developing a universal digital operational flight trainer (UDOFT). Other companies followed the same step, and from 1970 onward, the digital computer had improved to the extent that they could be used to achieve a high level of fidelity and reliability [19].

1.1.5. Visual System and Terrain Modelling for Simulation

The concept of providing the pilot with out-the-window feel when flying the simulator has been proposed and developed since the time of the Link trainer. In the early 1950s, the light projection method became popular in helicopter simulators. During this period, the Link could produce a high quality visual for their simulator by using film and an anamorphic optical system called Vamp. However, the Vamp was crippled with the inability to fly outside the area present in the film strip. A series of improvements took place after the development of the closed-circuit television system, leading to the first color system invented by Redifon in 1962 [19]. It was developed using a large rigid model that a camera moves over to simulate realistic terrain in the view of the simulator pilot (Figure 9). At that time, this system was so reliable that it remained in service for about a decade until the first computer image generation (CGI) systems were invented by the General Electric Company (USA). The growth in this new technology was so rapid that engineers were able to develop a 3d terrain simulation using it [19].



Figure 10 - Redifon Visual Terrain System [19]

1.1.6. UAV Flight Simulation

In recent years, the world of flight simulation has grown past its application in the pilot training of manned aircraft. It is now being used to train UAS pilots to help improve their decision-making and situation awareness during the UAV flight. The UAV simulator, called Multi-UAV, developed by the American Air Force and the Institute of Scientific Research, is a good example of how simulation is being used as a mission trainer for pilots. The MultiUAV it built using the MATLAB/SIMULINK technology, and it is capable of simulating multiple UAV at the same time. Other examples include the Unmanned Aerial Vehicle Research Test Bed (UAVRTB) by the Canadian Armed Forces, UAV Simulator by Ness Tech and Israel Aircraft Industry, and Virtual UAV developed by 5DT [20].

1.2. HUMAN FACTORS METHODS

1.2.1. Human Factors

The term human factors (or ergonomics) can be defined as multidisciplinary scientific fields that deal with understanding the interaction between humans and all other elements present in a system by applying various principles and methods to design to improve system safety, human well-being and overall system performance [21]. Human well-being has not always been the priority, especially in aviation. During World War I, for example, a high number of casualties among pilots was recorded. Combats did not cause the majority of these deaths; rather. human factors issues in the design of the airplane in general was the cause. Issues such as physiological stress on the pilots was a major problem, and precise design of the equipment to ensure mission effectiveness and safety were completely missing [22]. However, during World War II, despite the incredible advancement in technology, greater emphasis was put on studying human capabilities and limitations. Numerous guidelines for the design of the display, controls, environmental systems, equipment, and communication systems were developed, which brought about the start of human factors as part of design requirements of any system.

After the war era, human factors became increasingly popular in military and commercial aviation. This occurred after the industry realized that most accidents happen due to human negligence and error rather than mechanical failure [23]. It was recorded that the majority of all manned aircraft accident causes can be traced back to human factors issues. Similarly, the same problems were also being encountered in the realm of unmanned aircraft. About 67% of UAV accidents can also be directly or indirectly related to human factors issues [24]. Moreover, about

16

24% of these accidents can be attributed to specific human factors errors in the ground control station and interface design [25].

1.2.2. Human-Centered Design

Human-centered design is an approach to system design and development that is concerned with making an interactive system more usable by applying the human factors/ergonomics and usability knowledge and techniques [26]. According to the ISO 9241-210:2010 standard [26], to successfully carry out a human-centered design and development, four major human-centered design activities must be considered. First, the context of use must be well understood, and the user domain must be specified. This process includes consulting domain specialists and primary users for a better understanding of the design context. Second, since the major strength of the human-centered design is the involvement of the users in the design process [27], a well-detailed plan for specifying the user requirements must be produced. The third step is concerned with the application of specialized human factors techniques to produce an iterative design solution that meets both the user and the stakeholders' requirements. Finally, proper evaluation of the design must be carried out to validate the whole process.

Following the steps stated above, a clear plan was developed for achieving a humancentered design for the tandem rotor UAV. Figure 11 below shows a human-centered design plan that was created for the GCS Design

17



Figure 11 - Human-Centered Design Plan for GCS Design

As shown in the illustration above, to compile the final design requirements, a detailed analysis of the primary users' requirements must be performed. After the completion of the analysis, a proper description of the user characteristics and domain must be well understood. The second step is the initial design requirements, which involve stakeholders' design requirements and considerations. The final step involves considering all available human factors standards and guidelines for applicable requirements. When the final design requirements are completed, an iterative design of the GCS will be performed using the commercial off-the-shelf (COTS) software. Evaluation of the design procedure proposed will be the last step of the Human-Centered design.

1.2.3. Hierarchical Task Analysis Method (HTA)

The HTA is regarded as the most popular task analysis method [28]. This is because at the initial stage of most human factors analysis method, it is usually required for simplification of analysis [29]. It has been applied to numerous domains, ranging from civil aviation to process control and power generation. When compared to other task analysis methods, the major advantages of HTA include the ease at which it can be implemented and the minimal training it requires. Using the HTA, the level of available detail in any analysis can be varied depending on the purpose [29].

To properly carry out HTA, task under analysis must be defined. This involves stating the purpose of the task analysis. After the completion of the task definition, the data collection process must take place. Depending on the level of detailed required, the data used for HTA analysis can be collected through interviews, questionnaire or observations. The task identified can be broken down into goals and sub-goals which comprise the actions that must be taken in hierarchical order to complete the task. Moreover, finally, the plan and sequence that must be followed to complete the goals and actions must be defined.

Figure 12 below shows the processes that must be followed to carry out a HTA analysis successfully.

19


Figure 12 - Hierarchical Task Analysis Process Flowchart[29]

1.3. TANDEM ROTOR UAV OVERVIEW

Tandem rotor helicopters possess two propellers in both ends of the aircraft. They are popularly used to carry out military missions due to certain advantages they have over single rotor helicopters. Examples of these advantages include the low horsepower required in the shaft for the aircraft to hover, and the elimination of the tail rotor [30]. Boeing CH-47 Chinook, shown in Figure 13 is one of the most popular tandem rotor helicopter.



Figure 13 - Boeing CH-47 Chinook [31]

In terms of appearance and configuration, the tandem rotor UAV is very similar to the CH-47 tandem helicopter. The system of the UAV comprises of six main subsystems; five of these subsystems are on board while the remaining one is on the ground. The onboard subsystems include the autopilot system for autonomous control of the UAV, the front facing camera for image capturing, processing and continuous object tracking purposes, the RADAR for long-range obstacle profiling and detection, and the LiDAR and LEDDAR for manual and autonomous obstacle avoidance. The subsystem on the ground is the ground control station connected to the UAV via a data link for monitoring and managing the UAV flight from the base.

The main function of the autopilot system is to monitor the flight properties of the UAV (pitch, roll, yaw, speed, and position). It is also capable of receiving the radio control (RC) signal for switching between the control modes when required, and reading predefined flight path files sent by the pilot via the GCS. The Micro-Pilot autopilot was chosen because it is reliable, meets all the autopilot requirements, and is suitable for rotary wing UAV. The autonomous flight of the

Micro-Pilot is achieved using the Global Positioning System (GPS) receiver and inertia measurement unit (IMU) [32].

The basic functions of the RADAR are detection and ranging of objects from the point the RADAR is placed [33]. The RADAR detects and locates objects by transmitting an electromagnetic signal and measuring the distance it takes for the transmitted signal to echo back. The RADAR system has not been integrated into the tandem rotor UAV; however, it has been included in the simulation mode for testing.

The four LEDDAR sensors cover 360° (90° each) within a 20 m range to detect obstacles that could potentially hit the propeller. The LEDDAR M16 was chosen for this purpose. The M16 transmits 16 laser beams with a total field of view ranging from 9° to 95° for accurate obstacle detection and ranging [34].

The function of the two LiDAR sensors attached to the UAV is to detect objects within the 100 meters range to help the aircraft perform a successful obstacle avoidance. The LiDAR has a 360° horizontal field of view and a 30° vertical field of view. Its rotation speed ranges from 5 to 20 rotations per second, transmitting approximately 300,000 points per second. The first LiDAR is located 50 cm below the UAV body frame's center of gravity reference, and it rotates 360° in the direction the UAV is facing. Similarly, the LiDAR 2 is located 50 cm below the CG. However, it rotates in a sideways direction. Using the left-hand rule, the LiDAR 1 rotates in the x-direction while the LiDAR 2 rotates in the y-direction.

1.4. RESEARCH TOOLS

1.4.1. Presagis HeliSIM

The Presagis HeliSIM is a standalone simulation software for rotary wing aircraft. Using the software, the user can define many parameters to describe the performance of aircraft in a simulation environment. Aerodynamic parameters, weight balance, fuel tank location, the center of mass and moment of inertia can be defined. Individual subsystems can also be defined here with details of how the main system powers them. The control laws of various flight instruments and autopilot can also be defined. These laws are usually defined with graphs generated by HeliSIM user interface, or can be imported directly from MATLAB Simulink

In this research, for simulation and testing purposes, the flight dynamics and engine parameters of the tandem rotor UAV was defined using HeliSIM. As a result of this, parameters could be passed to the GCS for human-in-loop simulation.

1.4.2. Presagis VAPS-XT

VAP-XT is a Presagis software for developing a C++ object-oriented graphical HMI which can be applied to data display, simulation, and embedded system developments. The code generator implemented into VAPS-XT can be used to link the application with C++ for complex development. It can also be used to generate a standalone executable that can run on numerous operating systems. For this reason, companies have used the software to develop a cockpit display for aircraft and simulators. Figure 14 below shows the cockpit developed by Gulfstream aerospace using VAPS-XT [35].

VAPS-XT was used in this research to transform all the requirements into a real GCS display.



Figure 14 - Simulator Cockpit designed by Gulfstream using VAPS-XT [35]

1.4.3. Lightweight Communications and Marshalling (LCM)

LCM is a message passing and marshalling tool for real-time systems where high-speed connection and low latency is critical. It simplifies complex low latency message passing systems into tiny bits for better efficiency by providing a model for sending and receiving messages with automatic marshaling and unmarshaling generation of codes compatible with different platforms and programming languages [36].

LCM was used in this research for receiving the flight parameters from the real tandem rotor UAV. In short, flight parameters are received from HeliSIM when the GCS is in simulation mode; however, when in real-time mode, all the parameters are sent and received using LCM.

1.5. RYERSON RECONFIGURABLE GCS

The Ryerson reconfigurable GCS was designed at Ryerson Mixed Immersive Motion Simulation (MIMS) Laboratory (shown in Figure 15). The MIMS laboratory is supervised by Dr. Joon Chung, professor in Aerospace Engineering department at Ryerson University. The research conducted in the lab includes, Human-Machine interface (HMI), flight data analysis, simulation and Multidisciplinary Design Optimization (MDO), further details about the lab are available at http://www.ryerson.ca/~j3chung/.



Figure 15 - Ryerson MIMS Laboratory

The reconfigurable GCS was designed with the beta version of Presagis VAPS-XT 4.1 and FlightSIM 13. The main feature of this GCS is the feature that allows operator to reconfigure all the 12 instrument panels present in the GCS. All the panels can be scaled and moved to any location on the single display screen. It also comes with tab creation feature for saving configured instruments panels. The instrument panels are grouped into six different groups which include: Control panels, navigation panels, engine panels and payload panels. Control panels contain the instrument panels that are related to the flight control of the UAV, navigation panels for gesture controlled and non-gesture controlled maps, Engine panel for the engine parameters, while the payload contains the payload panels. As seen in Figure 16 below, by selecting each group, instrument panels that are needed are dragged to the main screen, the position of all panels in the main screen can be saved as a tab.



Figure 16 - Reconfigurable Feature of Ryerson GCS [37]

1.6. RESEARCH OBJECTIVES AND CONSTRAINTS

The objective of the research is to design and evaluate a tandem rotor UAV GCS using available human factors standards and guidelines. The study considered all the applicable human factors standards and guidelines to ensure optimum human factors integration. Since the purpose is to improve the concept of the existing re-configurable GCS, in the evaluation stage, the results will be compared to the reconfigurable GCS designed at Ryerson University's Mixed-Reality Immersive Motion Simulation (MIMS) laboratory.

The research is part of Presagis Canada Inc.'s project of creating a GCS for Roy Aircraft & Avionics Simulation Inc. (RAAS) for a tandem rotor UAV. For this reason, specific information regarding the tandem rotor UAV will not be disclosed during this study. Presagis Canada Inc. is a global provider of software for the development of modeling, simulation, visualization, and embedded display applications. The whole work was performed while the researcher was working as an intern at Presagis Canada Inc. Therefore, the final product presented in this thesis belongs to Presagis Canada Inc.

CHAPTER 2: USER REQUIREMENTS

User data-gathering is a crucial first step for designers and human factors practitioners. It is important because it sets the pathway for redesigning an existing system or developing an entirely new one from scratch. It gives the designer an in-depth view of the user - who they are, how they expect to perform, constraints, and the qualitative and quantitative usability level they expect in the system. Usually, data regarding user characteristics are gathered by conducting a direct or indirect questioning session with a specialist with abundant knowledge of the user group, the stake holders, or the primary group of users that are currently using the existing or closely related system. This questioning session can be in the form of a questionnaire, interview or observation, depending on the type of information and detail needed to be collected.

The direct interview method was used as the user data-gathering technique during this research. In this phase, the specialist behind the design of the existing tandem rotor UAV ground control station and some of the primary users and stake holders of the current system were consulted. By interviewing these groups, we could generate a clear picture of who the real users are and the expectations that must be met during the use of the system.

The first set of information that was collected focused on the characterization of the users that are relevant to using the ground control station. These characteristics include age group, pilot/flying experience, physical limitations and more. The expected primary users of the tandem rotor UAV ground control station are adults within the age group of 18-50, with flying experience ranging from low to medium. The form of flying experience referred to could be on fixed or rotary wings aircraft, UAV, flight simulators or flight games. Below is a well-detailed table showing the expected characteristics of the tandem rotor UAV ground control station users.

Table 3 -	User Chai	racteristics
-----------	-----------	--------------

User Characteristics	Description	
Age	18 to 50	
Sex	Both male and female can use the system	
Physical limitations	The user of the system may be a fully abled body or may possess	
	minor physical limitations such as hearing. The height of the user	
	is irrelevant.	
Education Background	Users may have minimal to a high level of education. However,	
	the ability to understand basic aviation terms may be required.	
Flying experience	Having a relevant flying experience either as a fixed or rotary	
	wing aircraft pilot, UAV, flight simulators or games may be an	
	asset. However, the willingness to learn is the most important.	
Motivation and attitude	The attitude and motivation of the user are directly related to its	
	learnability and usability. Users could be highly motivated if they	
	can use the system to carry out the tasks easily without having to	
	waste much time understanding it.	

2.1. USER DOMAIN DESCRIPTION

After data regarding characteristics of the users were gathered, the next step, domain description, was defined. Domain refers to the area of expertise and specialized knowledge needed to perform tasks and accomplish goals. The process of gathering information about the domain is known as domain analysis and to gather accurate information, it is mandatory that an interview with the domain specialist be conducted. Clear and detailed information regarding the domain being investigated was obtained through a direct interview process [38]. The interview conducted was with the founder of Roy Aircraft and Avionics Simulation (RAAS). He is the designer and domain specialist behind the existing tandem rotor UAV ground control station. With his many years of experience in the field of GCS and autonomous flight control design, all the essential knowledge needed by the user to use the GCS and safely control the UAV were obtained.

After the interview session with the domain specialist involved in the project was completed, the list of specialized knowledge the user of the system needs to know was put together. The first in this category is the manual and automatic flight control system. The manual flight control refers to the state of flight where the pilot is completely in control of the system/aircraft. When the aircraft is in this state, the pilot must pay full attention to controlling the aircraft using the throttle control and the joystick. The throttle and joystick must be adjusted reasonably to enable the aircraft to maintain a stable flight and fly safely. On the other hand, automatic flight control system refers to a state of the aircraft where the automatic throttle system (ATS), automatic flight control system (AFCS) or the flight management system (FMS) is engaged. The term "automatic" may be confused with "fully autonomous system" that requires no input or monitoring to complete a task. However, the pilot is required to monitor the state of the aircraft in case there is any failure in the system that may be critical to the safety of the aircraft. Also, the pilot must be aware of the techniques needed to generate a flight plan data that can be transferred to the UAV and activated using the FMS. These techniques include setting waypoints, loiter points which are position data stored in longitude and latitude format that can be used to define the location the FMS will take the aircraft when engaged.

Secondly, for pilots to carry out surveillance tasks, a camera will be attached to the tandem rotor UAV which will be accessible and controllable by the pilot controlling the UAV with the GCS. The inputs recorded will be sent to the UAV at high frequency via data streaming technology. For this reason, the user is expected to have deep knowledge of how the camera and camera control system work. The difference between the aircraft direction and the camera direction must be well understood. This need to emphasize this fact must be higher, especially when the pilot has had years of experience flying the manned aircraft. Such pilot can easily

mistake the aircraft out-the-window view with the camera view. This difference and some other essential knowledge of camera zooming, object tracking and more are significant and must be clarified.

The knowledge of how the avoidance system works is also crucial. Unlike the manned aircraft, since the UAV does not possess the out-the-window view, it relies solely on its avoidance system. The avoidance system uses the information acquired by the proximity sensors when an obstacle is being detected to generate the necessary maneuver needed to avoid the obstacle. The process is not as simple as it seems because each sensor has its orientation and detection range. For this reason, the user must have a clear understanding of the core logic behind the LiDAR and LEDDAR sensors and how they work together in the system. Understanding the basic idea behind the object detection process of the LiDAR or LEDDAR sensor cannot be regarded as sufficient, but also a clear understanding of the level of danger the detected obstacle poses to the safety of the UAV is also important.

The basic understanding of the aircraft status and connection management is also necessary. In this case, the UAV is regarded as an entirely different system. On a manned aircraft, the pilot is not only provided with the out-the-window view but is also present in the aircraft being flown. Therefore, it can be felt by the pilot when the aircraft starts descending rapidly, or when an obstacle can be seen on the runway on which the aircraft is about to land. However, it is not the case with UAVs. All the information such as the position of the aircraft, the avoidance system, camera information and more must be streamed to the GCS via datalink for the pilot to see. The GPS data and signal strength of the datalink and each payload are also necessary. For this reason, the pilot must have a clear understanding of what it means when the connection between the UAV and the GCS is low or entirely lost.

Since the GCS of the tandem rotor UAV is equipped with RADAR, the core understanding of how the RADAR technology works is also crucial. RADAR is an acronym meaning radio detection and ranging. It detects objects by transmitting radio signals from a narrow-beam antenna which scans the horizon and times the rate at which the signal travels back. Each detected object, either moving or stagnant, is displayed at its correct range and bearing, making it easy for the pilot to interpret the state of the object of interest [39].

Finally, the user must possess the ability to interpret the engine information displayed on the GCS. The tandem rotor UAV is equipped with a powerful 4-cylinder 100 hp electronic fuel injection engine [40]. Thus, the ability to read and interpret the engine consumable resources such as the fuel, oil, battery level and more, is mandatory to the safety of the aircraft and the mission success.

Figure 17 below shows the well-detailed flowchart to explain further the user domain model explained above.



Figure 17 - User Domain Model of Tandem Rotor UAV

2.2. USER GOALS, TASKS, AND ACTIONS

The main reason behind user requirements gathering is to describe and analyze the goals, tasks, and actions the user must be able to perform when using the system. However, before these terms can be described, understanding their differences is a mandatory first step that must be addressed. The goal is the overall result needed to be achieved. The specific sequence and details of how a goal should be reached are not stated in the goal definition. However, it must be described at a high level of abstraction showing what must be achieved [38]. A task, on the other hand, is a structured set of activities carried out in a specific sequence to achieve a goal [38]. Furthermore, to successfully perform a task, the user must physically interact with the system by carrying out a set of actions. Actions are individual operations needed to complete a task [38]. In summary, to complete a goal, tasks must be carried out. Moreover, to accomplish tasks, actions are needed to be performed. The Figure 18 below shows the relationship between goal, tasks, and actions as explained above.



Figure 18 - Relationship between Goal, Tasks, and actions

2.3. TANDEM ROTOR UAV USER GOALS, TASK, AND ACTIONS

First, before specific goals, tasks and actions related to tandem rotor UAV can be addressed, a precise definition of a safety critical system must be emphasized. A safety critical system is one that takes the safety of humans and the environment as the most important goal. Therefore, the system must be designed in a way that every user action that could affect the safety of the crew, environment or the aircraft itself, must be achievable without any confusion or ambiguity. Examples can be seen in a nuclear power station, air traffic system and aircraft cockpit design. In a modern aircraft cockpit, the pilot must be able to distinguish between when the autopilot is engaged and when it has been disengaged. Any ambiguity in the representation of these two states could bring about a disastrous outcome.

The tandem rotor UAV is a safety critical system. Even though it is an unmanned aircraft and no human pilot is present on board, the safety of the environment and the UAV is paramount. For this reason, it is paramount to have a clear design to ensure that the pilot does not, at any point, put the safety of the environment or the UAV in danger. Having this in mind, the need to categorize every user action into primary and secondary action groups is crucial. The primary action group refers to the actions that are critical to the safety of flight. An obvious example of this will be the speed control in modern aircraft. The speed control and indicator are very critical actions because if the pilot fails to know the rate at which the aircraft is flying, it is almost certain that a disaster is going to occur. On the other hand, the secondary action group will refer to the actions that are explicitly related to the payload control. An example of this is the camera control. The representation of the camera orientation, for instance, does not have any effect on the state of the aircraft/UAV. However, the pilot could be distracted from carrying out the important primary actions when it is poorly designed.

In this research, carrying out a continuous target surveillance mission while keeping the UAV and the environment safe is the top priority of the user controlling the tandem rotor UAV via the GCS. Continuous target surveillance mission refers to a mission where the pilot is expected to search and follow a ground-based stationary or moving target. The target, in this case, could be a building, vehicle or human. While carrying out this mission, the pilot must also avoid anything that could put the environment or the UAV in potential danger.

Carry Out Continuous	Keep the Aircraft and
Target Surveillance	the environment safe

Figure 19 - User Goals for Tandem Rotor UAV

The above representation of the user goals of the UAV (Figure 19 above) made it appear as if the users have two different goals to achieve. However, it must be noted that these goals were considered as one because they are both dependent and cannot be achieved individually.

As explained in the previous section, the user goal was broken down into sub-goals, which were further analyzed using the hierarchical task analysis (HTA). Figure 20 below shows the initial breakdown of the main goal into sub-goals.



Figure 20 - Goal and Sub-goals of Tandem Rotor UAV

2.4. HIERARCHICAL TASK ANALYSIS (HTA) OF TANDEM ROTOR UAV

As explained in the previous section, the HTA representation used in this project followed the concept presented in Huddlestone's conference publication [41]. Usually, in HTA, every hierarchical level must be numbered starting from zero. The overall goal usually retains the level zero; however, for simplification, each sub-goal was assigned the level zero value in the analysis.

Figure 21 to Figure 30 below show the HTA results for each sub-goal defined above. Figure 17 focuses on the task and actions the pilot must accomplish to carry out a manual takeoff successfully. The actions that must be performed to ensure that the throttle position is at zero include checking the current throttle position and moving it back to zero. Similarly, to validate the condition of the aircraft, engine, link and obstacle error must be checked. Moreover, finally, as seen in the figure, to complete the take-off, seven actions must be accomplished. To transform these actions into design requirements, the following process must be followed:

- The final action nodes of each HTA analysis must be recorded. For example, 1.1 Check Current Throttle Position (Figure 21)
- A user requirement must be created based on each final action node. For example, using the same action node stated in above example. As a requirement, the user must be able to see the throttle position on the GCS.

• The requirements must be categorized into the primary and secondary groups. The above requirement will be considered as a primary requirement because the pilot must always be aware of the throttle position of the aircraft.

The requirements and the grouping result can be seen in APPENDIX H



Figure 21 - HTA Result for Executing Manual Take-off



Figure 22 - HTA Result for Executing Manual Landing



Figure 23 - HTA Result for Executing Automatic Take-off



Figure 24 - HTA Result for Executing Automatic Landing



Figure 25 - HTA result for Executing Automatic Obstacle Avoidance



Figure 26 - HTA Result for UAV Status



Figure 27 - HTA Result for Camera Tracking



Figure 28 - HTA Result for RADAR Control



Figure 29 - HTA Result for Waypoints Creation



Figure 30 - HTA result for Executing Flight Plan

2.5. INITIAL DESIGN REQUIREMENTS

Just as much as the user requirements data gathering, the initial design requirements data is also crucial. The process of gathering this specific type of data is important because stakeholders' instructions regarding the design are considered, which include: constraints and budget set for the project. For instance, the user domain analysis and user requirements data may require something that is way above the budget of the project. The initial design requirements reform this requirement to make it more realistic for the designer to achieve and stay within the constraints and budget of the project. In summary, the data acquired during this process can be considered as a much more realistic version of the user design requirements.

During this project, the data collection for the design requirements was carried out using a questionnaire. All the companies involved were given a questionnaire regarding the GCS design, and they all gave their opinion on what must be presented in the interface. Questions regarding the task performance of the pilot, UAV specifications, automation, design interface, situation awareness, controls, communication, collision avoidance and separation assurance were asked. Most pilots, ranging from medium to highly experienced, are familiar with the current aviation technology that is available now. These experiences are based on an individual pilot's experience during flight training or solo practice with simulator games. However, modern-day UAV are equipped with payloads and GCS interface designs that are entirely new to most pilots. Some of these new additions also require pilots to perform tasks that are unique to the GCS and UAV alone. Therefore, in the questionnaire presented to the stakeholders, the first set of questions asked is concerned with the pilot tasks that are unique to the UAV, or the tasks that will present unique challenges to pilots [42]. Firstly, it was noted by the stakeholders that high priority should be placed on the flight management of the UAV when there is a nearby proximity object. It was emphasized that detected objects must be displayed on the GCS visualization interface for the pilot to see and interact with. This includes the LiDAR and LEDDAR points visualization as noted in the previous section. By using the visualization interface, the information displayed on the GCS must be sufficient for the pilot to react properly to the proximity objects by either commanding a manual or automatic avoidance maneuvers. This task was considered as the most challenging from the point of view of a pilot, and designing a GCS that makes the task easier to perform must be regarded as one of the main design goals that must be achieved. Furthermore, emphases were made regarding automatic flight by the stakeholders. It was pointed out that instances at which the automatic obstacles avoidance commands are enabled, the pilot must be able to supervise and verify the status of the avoidance system and the sequence the aircraft is following to execute automatic maneuvers.

Secondly, to enable optimum task performance, the question regarding information the GCS must provide to the pilot in control was put forward [42]; the subject of avoidance system was also emphasized in this case. Additionally, it was mentioned that the GCS must display the current operational mode of the UAV to the pilot, as well as data showing the current state of the aircraft such as position, velocity, altitude and communication status with the onboard platform. References to the operational flow of the Micropilot HORIZON software were mentioned as well. This includes: the computer in control mode (CIC), arcade mode (ARC) and pilot in control mode (PIC) [43]. The GCS is expected to have a CIC mode where a flight path file will be provided, defining the waypoints to be flown. Similarly, in ARC and PIC modes, high-level

position and velocity commands or low-level commands where inner-loop attitude commands can be provided (feeding command to the mixing function directly).

Regarding the UAV limitations, the request of the stakeholders included a GCS interface that provides information to enable the pilot to use the system safely without exceeding the boundaries of the aircraft.

2.6. HUMAN FACTORS STANDARDS AND GUIDELINES

For standards and guidelines to be adequately applied, information available to the public was compiled and set as part of the design requirements. This project considered specific publications such as the Human Factors Design Standard (HFDS), handbook for Human Engineering Design Guidelines (MIL-HDBK-759C), Department of Defense Design Criteria Standard (MIL-STD-1472F), Standard Interfaces of UAV Control System for NATO UAV Interoperability (STANAG 4586 Edition 2) and the Human-Centered Design for Interactive System (ISO 9241-210:2010).

Human Factors Design Standard (HFDS) is a well-detailed human factors document compiled by the Federal Aviation Administration (FAA) for designing COTS subsystems, nondevelopmental items or development systems. It was originally intended as a guidelines document; however, since it has been transformed into a standard document, it can now be considered as part of the requirements when integrating human factors into any design [44]. The MIL-HDBK-759C, on the other hand, is a guidelines document on human engineering design for military systems, equipment, and facilities. Since the system being designed is a safety critical system carrying out a surveillance mission just like most military aircraft do, some parts of the military system design concepts can be applied to the system [45]. The MIL-STD-1472F offers

to the designer the general human engineering criteria for design and development of the military system, equipment, and facilities. The document aims to help achieve the required performance of the operator and reliability of the system while minimizing the skill and personnel training time [46]. The STANAG 4586 is a more technical standard regarding the design, defining the architectures of the data element and the message formats. This document will mainly be used as a guideline when programming some parts of the GCS such as the mission control and map representation [47]. Other guidelines used include the human factors guidelines for unmanned aircraft system GCS journal article and report published by NASA UAS researcher, Alan Hobbs [39, 45].

CHAPTER 3: PRELIMINARY DESIGN AND SET-UP

3.1. PRELIMINARY DESIGN

During the preliminary design stage, the previous GCS developed by Roy Aircraft Avionics Simulations (RAAS) was studied. The RAAS GCS, called Venom, was developed using IData 3.0. IData is a powerful Human Machine Interface (HMI) development application with certifiable libraries and Aeronautical Radio, Incorporated (ARINC) 661 capabilities. As shown in Figure 31 below, the Venom GCS has four instrument panels, all divided in a single screen equally. The map and the PFD can be maximized by clicking any corner of both panels; however, the camera and engine/waypoint panel do not have this feature. The main problem identified with this GCS is the fact that it does not show enough information. Also, the map and the PFD have no constraints when being maximized, blocking information that is critical to the safety of the flight. Moreover, it is quite difficult for users to understand some features of the display.



Figure 31 - Venom GCS Developed by RAAS [49]

After the Venom GCS was closely studied, the need to incorporate the latest technology into the new GCS became obvious. Contrary to the old display monitors that are available in the market a few years back, the technology behind the display screen has now improved to a very high level. There are now desktop monitors capable of ten touch inputs in the market. Therefore, integrating a touch-capable display will provide more room for diverse development ideas, especially in the design of the map instrument panel. Also, the need to develop a GCS that is carefully designed and logically constrained following standards and guidelines for human factors is necessary.

After a long session of brainstorming, the first concept that was thought of was the implementation of the Luciadlightspeed engine to develop the map panel. Luciadlightspeed engine is a geospatial application that allows the user to develop advanced Command and Control (C2) products with location intelligence [50]. Therefore, it provides the foundation for developing an advanced map panel with the capability of toggling between 2D and 3D, creating waypoints and loiters.

The second idea was the mission panel. Since the UAV is equipped with obstacle avoidance sensors, the need to have an advanced mission panel that can load flight path from file was necessary. Also, the idea behind the AFCS panel buttons was also developed conceptually in this stage.

3.2. DESIGN ITERATIONS

After the brainstorming session, the first design was developed using VAPS-XT 4.1. In this design, a blue theme was selected and all the design was made on a single display screen. The reason behind this is that the avoidance system panel was still under development and the

concept of dual screen has not been proposed by the stakeholders. As shown in Figure 32 below, the first design utilizes the side-by-side view concept, and the quick swipe gesture described in section 4.2 was already implemented.



Figure 32 - First Design Iteration [51]

The second and the final design iteration was designed using VAPS-XT 4.2. Both designs are similar and the only difference between the two is that the second design iteration has a simpler detailed engine panel and no message log panel. Figure 34 shows the second and final design iterations.



Figure 33 - Second Design Iteration [52]



Figure 34 - Final Design [52]

3.3. SET-UP

After the finalization of the GCS design, the construction of the pilot and payload controller workstations began. In any GCS workstation, there are two control options available. The first case is when the pilot controls both the aircraft and the payload. This is considered to be very stressful on pilots; therefore, recent configurations have endeavored to split the workload between multiple pilots. The second option is when the pilot controls the aircraft and monitors the payload. The control of the payloads is assigned to another operator called payload controller. In most cases, the two operators can switch controls during missions that require long hours of work. However, that feature has not been implemented in this design yet.

Both the pilot and payload workstations are in compliance with the compiled human factors standards and guidelines (I: Exhibit 5.1.2.10), optimum vertical and horizontal field of view. The workstations were designed to eliminate the head movement of the operator. According to the compiled document, for eye rotation only, the maximum angle of view is about 70 degrees, 35 degrees in both directions from the center. However, the optimum view of the operator is within 30 degrees from the center (15 degrees left and 15 degrees right). In order to
design a workstation that keeps the view of the operator within the optimum view range, the distance from the screen to the operator was calculated using the values shown above.

As shown in Figure 35 below, the pilot station is equipped with a dual display, two joysticks, keyboard, and mouse. The first joystick is for controlling the collective of the UAV, while the other is for controlling the pitch and roll axes. Both diamond 4 button joysticks were purchased from the Scorpion Technologies online store [53]. The collective joystick can only move on the lateral axis (up and down). The lowest point represents the zero-collective position, while the highest point represents the max collective position (100%). The joystick for controlling the pitch, roll and heading axes can be moved in all directions. Although the joystick always returns to the neutral position, the proportional–integral–derivative controller (PID controller) has been tuned to always maintain the last control position rather than returning the UAV to the neutral position too.



Figure 35 - Pilot Workstation [52]

Similarly, as shown in Figure 36 below, the payload station is equipped with a dual display, camera control device, keyboard and mouse. Compared to the pilot station, only the camera control device is different; other components have exactly the same specifications. The camera control device is manufactured by CH Products [54]. It can be rotated 180 degrees in both positive and negative directions. In addition to this rotation movement, it can also be moved in all directions (x and y axes). However, just like the joystick for controlling the pitch, roll and heading, it returns to its neutral position when released. The rotation movement of the device is used to control the zoom factor of the camera, while the directional movement is for controlling the bearing and elevation angle. The directional movement can either be in the left or right direction for changing the azimuth angle of the camera, or forwards or backwards for changing the elevation angle. As seen in Figure 37 below, the device also comes with six buttons numbered from 1 to 6. By pressing the number 1 button, the camera will try to track the object in view. To change between the camera view modes, by pressing the number 2 button, the camera switches display mode between Electro-optical display (EO) with the infra-red (IR) off, to EO IR-B (black hot) on and EO IR-W (white hot). The IR-B and IR-W modes can be used for missions carried out at night where the pilot does not have a clear visual of the surrounding objects.



Figure 36 - Payload Workstation [52]



Figure 37 - CH Products USB Desktop Joystick

CHAPTER 4: GCS CONFIGURATION

4.1. GCS SCREEN CONFIGURATION DESIGN

As part of the requirements defined in the initial design requirements section, a dual screen configuration was chosen by the stakeholders of the system. For this reason, the need to decide the best way all the available information can be displayed to the pilot was required. After numerous iterations, the configuration shown in Figure 39 below was chosen. As seen in Figure 40 below, the screen on the left side of the pilot is considered as the primary screen. This is where all the primary panels are placed, while the screen on the right is the secondary screen. At start up, the GCS prompts the user to select the source mode. When "Simulation Mode" is selected, the GCS sets HeliSIM as the data source. However, when "LCM Mode" is selected, the GCS sets the tandem rotor UAV as the data source via LCM. Also, the panels in the designed screen configuration were also made to be reconfigurable simply by rearranging the position and scale of each panel using the minimum amount of touches. Moreover, the process can be completed within seconds. As shown in Figure 41 and Figure 42 below, the PFD panel was dragged above the map panel. Immediately, the map takes the previous position and scale of the PFD and vice versa. Further details regarding the quick swap gesture can be seen in section 4.2 below. It must be noted that the pilot is not allowed to swap the primary instrument panels, such as the map and the PFD, to the secondary screen. This design decision was taken to prevent the pilot from losing focus of the panels that are directly related to the safety of the UAV. Finally, as interesting as reconfigurable panels are, the need to restrict this feature when the pilot needs to focus on critical tasks is necessary. Therefore, a lock icon is placed at the top corner of the AFCS panel (see Figure 43 below). When this button is touched, the GCS locks all the panels to their current position, disallowing panel swipe or drag.



Figure 38 - Start Screen [52]



Figure 39 - Dual Screen Configuration [52]



Figure 40 - Primary and Secondary Screen [52]



Figure 41 - Quick Swapping the PFD and Map [52]



Figure 42 - Final Position After Panels Have Been Swapped [52]



Figure 43 - Lock Icon [52]

4.2. DESIGN OF QUICK SWAP GESTURE

By closely investigating the compiled Human Factors Standards and Guidelines (HFSG) presented in APPENDIX A, the Subsection - Format, 8.1.3.6~8.1.3.17 emphasized the need for the user to be able to prioritize the display and move quickly and easily among items. For this reason, the quick swap gesture concept was developed to give the user the ability to not only

reconfigure the GCS instrument panels, but to also do it quickly and smoothly. Using this concept, the user could resize and reposition both primary and secondary instrument panels within the dual display screens. As stated in the compiled HFSG 8.1.2.13, the information or panels that require an immediate response from the user must be constrained to the main screen. Therefore, the instrument panel using the quick swipe gesture was designed to be constrainable.

During the initial design stage of the quick swap gesture design, a container object was created. A container object can be defined as a holder object that stores a collection of other objects. The objects held by a container are regarded as elements of the container, and each element held by a container retains both its attributes and the attributes of the container. The quick swap gesture was added to the container object created and each instrument panel was held in a container. With this strategy, the design time was minimized because the need to repeatedly create the quick swap gesture for each instrument panel was eliminated.

Firstly, the variables needed to create the quick swap gesture were defined. These variables can be seen in Table 4 below.

After the variables of the container were defined, the logic behind the swap concept was created, and the state chart diagram in VAPS-XT was used to set up and link different logical states together. Figure 44 below shows the exact state chart diagram used for carrying out the quick swap gesture.

62

Variable Name	Variable type	Variable Function	
Is Visible	Bool	Stores the visibility of the element/container	
Position	Coord	Stores the X-Y position coordinates of the	
		element/container	
Velocity	Coord	Stores the drag velocity in both X and Y axis	
DragLimits	Rect	Stores the left, bottom, right, and top coordinates in	
		float data type to specify the swipe gesture limits	
		within the display screen.	
EdgeSprintLimits	Float	Fixed value (2500) that specifies the sprint limit of	
		swipe	
AnimationDuration	Unsigned	Fixed value (250) that specifies the duration of	
	Integer	animation in milliseconds	
EnableDrag	Bool	Specifies the ability to enable or disable the drag	
		feature of a container/element	
EnableDoubletap	Bool	Specifies capacity to enable or disable the double tap	
		feature of a container/element	
IdlePosition	Coord	Stores the zero-velocity position of a container/element	
		in X-Y coordinates	
PreviousPosition	Coord	Stores the previous idle position of a container/element	
		in X-Y coordinates	
MaximizedState	Bool	Identifies if the container/element is in maximized	
		state or not.	
Screen	Integer	Specifies the screen a container/element is present. The	
		main screen is identified by 1, while the secondary	
		screen is identified by 2.	
EnableDragOver	Bool	Specifies the ability to enable or disable the swap	
		feature of a specific container/element	

Table 4 - Variables Definitions for Quick Swap Gesture



Figure 44 - State Chart Diagram for Quick Swap Gesture [52]

4.2.1. Idle State

The idle state is the zero-velocity position of the container. This state happens when the GCS application starts or when a swap gesture is completed. When the GCS application starts, the container goes from the initial state (Init) to the idle state. Similarly, when a swap gesture is completed or interrupted, the container goes from its current state to idle state. To make the container fall back to the idle state, the current velocity value was multiplied by zero each time.

```
Velocity = Velocity * 0.
```

4.2.2. Idle State to Dragging State Transition

When the EnableDrag is set to TRUE, the container's drag feature becomes enabled. In this state, it becomes possible for the user to hold and drag the touch area of the container. When this gesture is initialized, a void function evStoreIdlePos is called to store the idle position of the container and set the Velocity to zero on X-Y coordinates.

Velocity.X = 0; Velocity.Y = 0; evStoreIdlePos();

4.2.3. Dragging State

In the dragging state, a coordinate data object was created to store the start position of the container.

StartPos.Value=Container.Position

4.2.4. Dragging State-Throwing State Transition

When the drag is completed, and the user releases the container, another coordinate data

object (ControlPos) was created to control the position of the container. The ControlPos

records the position of the container after it has been thrown. The value of ControlPos is defined as follow:

ControlPos.Value = Container.Position + Velocity *
(AnimationDuration mSecs) / 2000;

The velocity indicated here refers to the drag distance per time ratio measured in both X and Y coordinates. To get the drag distance, the current drag position is subtracted from the StartPos value defined above. On the other hand, the time is measured by subtracting the current timestamp from the timestamp at the idle state.

An Extent group to house the container was also created. The logic to determine the distance between the extent group and the drag limits before the container is thrown can be seen in APPENDIX I

The relationship between the dragLimits, ExtentGroup and the container is shown in Figure 45 below. Before the dragging or throwing begins, the distance between the container and the four drag limits was determined. This was used to set the minimum and maximum values for the ControlPos. From the code shown in APPENDIX I, two temporary rect variables (Limit and Limit2) were created. The first variable, Limit, stores the four distances between container position and the drag limits, while the second variable, Limit2, stores the maximum ControlPos that can be achieved in every direction at every container position. An if-else statement was created to set the minimum and maximum for ControlPos if Drag limits of the screen have been defined. However, when these limits are not defined, the value of the ControlPos will be assigned to the final container position.

66



Figure 45 - Relationship between the dragLimits, ExtentGroup and the container

4.2.5. Throwing State

In the throwing state, an animation controller object with linear easing function was created. At this state, the container position is reassigned to the StartPos and the animation controller created was made to go to start before animating to the end. What happens in this state is shown below:

StartPos.Value = Children.Position; AnimationController.doGoToStart(); AnimationController.Duration_mSecs = .AnimationDuration_mSecs; AnimationController.doAnimateToEnd();

From this state, the container goes back to the idle position when the container is successfully released or thrown. However, when the drag gesture fails in the dragging state, the container either goes back to the Idle state or to the Animating State. In the Animating State, the container animates to the ControlPos calculated similarly to the Throwing State.

4.2.6. Double Tap Feature

As shown in Figure 46, the double tap feature gives the user the ability to maximize any instrument panel to full screen when the EnableDoubleTap is set to Bool::TRUE. When this gesture is initialized, a function to store the idle position is called.

.evStoreIdlePos()



Figure 46 - Maximized Primary Panels During Automated flight

This function was created because the instrument panels must be able to return to the previously saved position when minimized. Moreover, in the maximized state, the container must be stationary like the Idle state.

```
Container.Position = .Position;
.Velocity = .Velocity * 0;
.evMaximize();
.MaximizedState = Bool::TRUE;
```

Similarly, when the container is double tapped while in the maximized state, the

maximized state is set to Bool::FALSE and the minimize function is called.

```
.evMinimize();
.MaximizedState = Bool::FALSE;
```

4.3. QUICK SWAP GESTURE INTEGRATION

As shown in Table 5, after each instrument panel was inserted into a container, two UShort arrays, one coord array, and a float array was created. The first Ushort array View2Page was used to store the page number of each instrument panel when swapping from a high-scale container to a low-scale container. Similarly, the second UShort array Page2View was utilized to store the page numbers of each panel when swapping from low-scale container to high-scale container. The coord array ViewPos was used to store the position coordinates of each instrument panels, while the float array ViewScale stores the scale factor of each instrument panel at every position in ViewPos.

View2Page	Page2View	ViewPos	ViewScale
3	3	640 px, 180 px	1
1	1	2560 px, 540 px	2
2	2	960 px, 720 px	2
0	0	1280 px, 180 px	1
4	4	3520 px, 270 px	1
5	5	3520 px, 810 px	1

Table 5 - Data Objects for Quick Swap Gesture Integration

When an instrument panel is being dragged from a low-scale container position, the page number of that panel is determined by checking the finger position on the screen. If the finger position falls between any ViewPos range, the Page2View value and scale factor on the same roll is determined. Similarly, when dragging from the high-scale container, the View2Page value is determined using the finger position on the screen. After the drag starting position and page number are recorded, when the instrument panel is dragged on another idle panel, the position, page number and scale factor of that idle panel were also recorded and swapped with the dragged panel. To express this action logically, the following function (see APPENDIX J) was created and called for each instrument panels.

CHAPTER 5: GCS INSTRUMENT PANELS DESIGN

5.1. PRIMARY INSTRUMENT PANELS

5.1.1. Primary Flight Display (PFD)

In every air vehicle, it is necessary for the pilot to know the instantaneous speed, altitude and behavior of the aircraft in the vertical, longitudinal and lateral axes. The PFD panel, as shown in Figure 47 below, was designed to assist the pilot in getting this information. It shows the heading, speed, vertical speed, pitch, roll, and altitude of the aircraft. The design of the PFD followed the traditional layout used in modern commercial aircraft. The heading of the aircraft is displayed at the upper end of the PFD. Contrary to the common circular heading design, the linear design was chosen. Below the heading tape, the roll indicator can be seen. The roll indicator gives the pilot the degree at which the UAV is rolling.





Right in the middle of the PFD, the Attitude Direction Indicator (ADI) can be seen. this acts as the virtual representation of the horizon. The blue region was used to represent the sky and the orange region as the ground. The two horizontal lines represented the position of the UAV; this can be utilized by the pilot to determine the orientation of the aircraft with respect to the horizon, while the horizontal lines with numerical values represent the pitch rate of the aircraft.

On the right-hand side of the ADI, two finite tapes showing the altitude indicator and the vertical speed indicator of the UAV was displayed. The altitude indicator displayed the height at which the UAV is flying in feet (FT), while the vertical speed indicator presented the rate at which the UAV is ascending or descending in feet per minute. Below these tapes is an indicator showing the RADAR altitude of the aircraft.

Finally, on the left side of the ADI, the indicated air speed tape is shown. This was designed to display the speed of the UAV to the pilot. Below this tape, a recovery button can be seen. Pressing this button can force the UAV to fly to the closest recovery waypoint.

5.1.2. Mission

The mission panel (as shown in Figure 48 below) displays the flight path or waypoint data to the pilot. It can also be used to create, load or save waypoint data which can be transferred to the UAV via datalink communication. Saved mission files can be loaded using the pop-up bar at the top end of the panel. As shown in Figure 49 below, the bar is scrollable, and users can easily browse through the list of saved mission files in the specified directory. Below the pop-up bar, the live and edit button can be seen. When the user selects the edit button, the waypoint data can be created or edited without modifying the current data loaded on the UAV.

72

However, when the send button is pressed, the mission created in the edit mode will be sent to the UAV avionics box via the communication link. The flight data displayed on the mission panel is also separated into waypoint and loiter categories. The reason for this is to give the pilot a clear understanding of when the aircraft is going to perform a maneuver or fly to the next one.

MISSION				
Demo				
EDIT		LIVE		
WAYPOINT	LAT COU LON SE ALT	JRSE PROC PEED TIME		
	12°52'55.81 45°04'52.83 0500 PSTD 12°50'08.52 45°04'38.27 0600 PSTD 12°47'20.19 45°03'06.14 1000 PSTD 12°47'41.00 45°00'13.54 1000 PSTD	0 125 IAS 0 ROC 0 90 IAS 0 ROC 0 CIRCLE 90 IAS 0 ROC 0 TRACK 90 IAS 0 ROC		
LOITER RADIUS	LAT LON SF ALT	BRG PEED TIME TURNS		
5 õ E	12°47'20.19 45°03'06.14 1000 PSTD	0 90 IAS 0 1x		
6 💍 N	12°47'41.00 45°00'13.54 1000 PSTD	-45 90 IAS 0 1x		
X				

Figure 48 - Mission Panel [52]



Figure 49 - Pop-Up Bar for Loading Mission Files [52]

As shown in Figure 50 below, after creating multiple waypoints or loiters, the pilot can also input the mission file name which can be loaded and added to the pop-up bar. An on-screen touch keyboard was also designed and integrated for the user to have an easier way of inputting the mission file name.



Figure 50 - File Name Input Screen [52]

The design of this panel followed the STANAG 4586 guidelines [47], and the majority of the design implementation was made using visual studio C++. In C++, the TinyXML library was used to design a code for sending, receiving and modifying waypoint information utilized by the UAV. TinyXML is a simple C++ XML parser that builds a document object model (DOM) which can be modified and implemented into other programs [55].

Before the logic behind the mission panel can be explained, the need to describe the kind of waypoint or loiter information that is readable by the UAV is required. The first is the waypoint/loiter ID, which is an integer number used for identifying a specific waypoint or loiter point. Following the waypoint/loiter ID, the waypoint/loiter type is also specified to determine if the waypoint or loiter is a take-off, landing, recovery, flyby or fly-over point. Take-off waypoint starts the UAV from the ground up, while the landing waypoint lands it. The recovery waypoint is a location the UAV flies to when either the recovery button is pressed, or there is a failure in the system. When there are multiple recovery locations, the UAV will fly to the nearest one when the recovery conditions are met. The fly-by and fly-over relates to how the UAV flies to the waypoint or loiter. In fly-over mode, the UAV will try to hit the waypoint before turning or proceeding to the next waypoint, while in flyby mode it will make the turn without necessarily hitting the point. Next is the altitude type, which identifies if the altitude received was computed using barometric, standard, above ground level or RADAR altitude. Local type and vertical type determine how the UAV will fly to the waypoint, while speed type shows the unit used to compute the UAV speed. The four units readable are indicated airspeed (IAS), true airspeed (TAS), ground speed (GS) and wind speed (WS). Finally, during a loiter maneuver, the procedure type identifies the type of turn the UAV will make at the loiter location. The radius, entry type and turn direction of the maneuver must also be specified.

75

Variable	Identifier
Waypoint Type	0 = FLYBY, 1 = FLYOVER, 2 = TAKEOFF, 3 =
	LANDING, 4 = RECOVER
Altitude type	0 = GEO, 1 = BARO, 2 = STD, 3 = RAD
Local type	0 = NONE, 1 = IN, 2 = OUT, 3 = INOUT
Vertical Type	$0 = MAX_ROC, 1 = PROP_CLIMB,$
	2 = RESTRICTEDALT
Speed type	0 = IAS, 1 = TAS, 2 = GS, 3 = AT
Procedure Type:	0 = NONE, 1 = CIRCLE, 2 = RACETRACK, 3 = FIGURE_8
Entry Type	0 = CENTER, 1 = TANGENT
Turn Direction	0 = RIGHT, 1 = LEFT

Table 6 - Waypoint and Loiter Data Identifier

The TinyXML was used to generate an XML file each time the user loads or creates a waypoint manually on the map panel. It records all the values using the format shown in Table 6 above. When a file containing waypoint, or loiter information is loaded from the mission directory, the TinyXML program will loop through all the attributes of the XML file to extract all the variables. To ensure that the program does not load a wrong value for each variable, variable type is also checked as it loops through the XML file. For example, the waypoint ID is an integer variable; before loading it checks the variable type using the QueryIntAttribute(WaypointID). The longitude and latitude are both stored using type double. Before loading the position coordinates, it checks if the value stored is double using QueryDoubleAttribute(Longitude). The XML file generated from the mission "Demo" shown in Figure 48 above can be seen in APPENDIX K.

5.1.3. Map

The map panel was designed using the Luciadlightspeed engine [50] to visually identify the current position and area the UAV is flying. The map can be displayed in 2D and 3D modes as shown in Figure 51 and Figure 52 below. This panel can also be used to create and store the waypoint utilized by the mission panel explained in the previous section. To create a waypoint, the operator must hold on a location on the map using a single finger. By completing this gesture, a pop-up menu will be displayed for setting all the waypoint or loiter parameters needed to create all the variables needed by the mission instrument panel. This pop-up menu can be seen in Figure 53 below. The icon on the top left brings out the control menu of the map as shown in Figure 52 below, while the button on the top right can be used to creater the map by changing its orientation to north up. On the lower left, the coordinate of the UAV is displayed. The OWN button toggles the track mode which follows the UAV position. The NAV button shows all the waypoints on the map view while the SEL button centers on the selected waypoint on the map panel. The CAM button goes to the current target location the camera is viewing. Moreover, finally, as shown in Figure 54 below, the scale of the map can be changed using the two finger drag gesture.



Figure 51 - 2D Map View [52]



Figure 52 - 3D Map View [52]



Figure 53 - Waypoint/Loiter menu [52]



Figure 54 - Touch Gesture for Changing Map Scale [52]

5.1.4. 3D Heading Display

The 3D heading display panel (see Figure 55 below) shows not only the heading to the pilot but also the entire attitude of the UAV. As shown in the right-side image of Figure 55, the heading of the UAV reads 200 degrees. However, it also indicates that the aircraft is pitching up and rolling to the right. The visual representation of the UAV's attitude gives the pilot a better understanding of the current situation of the UAV.

This panel was designed using the same wave front OBJ file imported from creator 15 as explained in the obstacle visualization section. However, the orientation of the 3D object was directly mapped with the heading pitch and roll of the UAV.



Figure 55 - 3D heading Display Panel [52]

5.1.5. Connection Status

The connection status panel shown in Figure 56 below displays the availability and the connection strength of the data link, GPS and all the payloads connected to the UAV. It also shows the mode in which the UAV is flying. The pilot in control mode (PIC) represents when the UAV is flying in complete manual mode, the arcade mode (ARC) represents when the UAV is flying in semi-automatic mode, and the computer in control (CIC) represents a fully autonomous flight.



Figure 56 - Connection Status Panel [52]

5.1.6. Automatic Flight Control System (AFCS)

The AFCS panel as shown in Figure 57 below is used to control the autonomous flight system of the UAV. On the first row, The COLL button represents the auto collective system. The button is equivalent to the Auto Throttle System (ATC) in fixed wing UAV which takes over the control from the pilot and controls the throttle of the UAV automatically. The AFCS button enables the autonomous control of the UAV attitude (pitch, roll, and speed), while the FMS button enables or disables the flight management system. On the second row, when the AFCS button is toggled, the ALT button holds the current altitude of the UAV. The AAQ represents the altitude acquire or altitude select which takes the UAV to a manually selected altitude. The VS button holds the current vertical speed of the UAV; it can also be used to fly the UAV at a manually selected vertical speed. The "P ATT" represents the pitch attitude, which holds the current altitude and vertical speed at the same time, while the navigation pitch (NAV P) button controls the pitch of the UAV using the navigation data of the flight management

system (FMS). On the third row, the heading hold button (HGD HLD) holds the current heading UAV while the heading select flies the UAV to a manually selected heading. Navigation Roll (NAV R) controls the roll of the aircraft using the navigation data of the FMS, while the roll attitude (R ATT) holds the current roll of the aircraft. Finally, on the fourth row, the IAS button holds the current indicated airspeed of the UAV; when tapped again while it is toggled, the UAV flies to a manually selected speed. The navigation speed (NAV S) controls the speed of the UAV using navigation data of the FMS. The hover button (HVR) forces the UAV to maintain a zero speed in the vertical and lateral axis. Finally, the ALT, VS, and KIAS bring out a pop-up bar for manually selecting the altitude, vertical speed and indicated airspeed, while a dial on the button can be used to choose the heading manually. The lock icon seen on the top right of the AFCS panel is not part of the autonomous control system; it can be used to lock the current position of all the panels in the GCS.



Figure 57 - Automatic Flight Control System Panel [52]

5.1.7. Simple and Detailed Engine Information

The simple and detailed engine information shown in Figure 59 and Figure 60 below displays the state of the UAV engine to the pilot. Since the engine is a crucial part of any mechanical system, close monitoring is always required. Therefore, keeping the pilot informed so the necessary steps can be taken when there is a possibility of engine failure is crucial.

The simple engine shows only the fuel flow, fuel level, engine speed (RPM), oil and battery level, and it can be seen in the lower left corner on the main screen of the GCS. All the parameters in the simple engine are displayed in percentage format because it helps to limit the possibility of overloading the pilot with too much information to monitor on the main screen. Although the simple engine information has been constrained to the main screen, the detailed engine information can be reconfigured just like every other instrument panel. The detailed engine shows more details compared to the simple engine; it also displays the exact value of all parameters. The reconfigurable feature of the detailed engine gives the pilot the chance to move the panel to any position on the screen if the engine needs to be closely monitored.

The numerical figures in the detailed engine were designed to have three color states green, yellow and red. These colors gave the pilot the ability to figure out the state of any parameter without having to memorize the exact value. On both the simple and detailed engine information, each bar was designed to be reusable depending on the type of engine. The maximum, minimum, warning, safe and low region can be recalibrated. The equation used to make this possible is given below from $(1) \sim (6)$.

$$X_{1Low} = \frac{250}{Ratio \ of \ MaxValue} * LowLimit_1 \tag{1}$$

$$X_{2Low} = \frac{250}{Ratio \ of \ MaxValue} * LowLimit_2 \tag{2}$$

$$X_{1Medium} = \frac{250}{Ratio \ of \ MaxValue} * MediumLimit_1 \tag{3}$$

$$X_{2Medium} = \frac{250}{Ratio \ of \ MaxValue} * MediumLimit_2 \tag{4}$$

$$X_{1High} = \frac{250}{Ratio \ of \ MaxValue} * HighLimit_1 \tag{5}$$

$$X_{2High} = \frac{250}{Ratio \ of \ MaxValue} * HighLimit_2 \tag{6}$$

In the equation (1) ~(6), X_{1Low} and X_{1High} represented the lowest points of the gauge bar's red region on the x axis, while the X_{2Low} and X_{2High} bar represented the highest points. Similarly, the $X_{1Medium}$ and $X_{2Medium}$ represented the lowest and the highest points of the warning region. The remaining background color was set to green (safe region). The Ratio of MaxValue and the limits are variables that needs to be inputted by the designer to calibrate the gauges depending on the type of engine being used. As shown in Figure 58, the value received from the engine was assigned to the variable Value and a condition to change the numerical color using an If-else statement was set.



Figure 58 - Engine Value Behavior Procedure



Figure 59 - Detailed Engine Information [52]



Figure 60 - Simple Engine Information [52]

5.1.8. Message Alerts

As part of the requirement of the GCS design, the system must be able to notify the pilot when there is an error or malfunction in the system. The message alert panel (as shown in Figure 61 below) can be seen on the left side of the GCS main screen. Normal messages that pose no threat to the safety of the UAV are shown in green, warning messages in yellow and critical messages in a blinking red. The blinking critical message helps to grab the attention of the pilot. However, to avoid distraction when the malfunction or error identified is being resolved, the message alert panel can be tapped by the pilot to disable the blink feature.



Figure 61 - Message Alert Panel [52]

5.2. SECONDARY INSTRUMENT PANELS

5.2.1. Obstacle Visualization

The ability to see the obstacle around the UAV being flown is an important feature for a UAV pilot. The feature is important because, unlike manned aircraft pilots, the UAV pilots do not have the out-the-window view. For this reason, the obstacle visualization panel was designed to display the surrounding images within the range of each LiDAR and LEDDAR sensors to the pilot. This panel was designed to help the pilot make accurate maneuver when there is an obstacle on the path of the UAV. Each sensor image is represented using different colors and, to help the pilot differentiate easily, on the lower left of the panel, legends showing the sensor each line color represents can be seen. Tandem aircraft are known to be able to make various types of maneuvers. Thus, it is necessary that the pilot can change the orientation of the UAV concerning the sensor images shown. Using drag gesture, the UAV 3D representation and the sensor orientation can be modified simultaneously. Six toggle buttons representing all the avoidance sensors was designed on the lower left of the panel to give the pilot the capability of decluttering the number of images displayed at a time to avoid confusion and reduce workload. However, disabling any of the LiDAR or LEDDAR image does not stop the system from notifying the pilot whenever an obstacle is being detected.

The 3D representation of the Tandem Rotor was designed using Presagis Creator 15 (a 3D CAD design software). The 3D object was designed with the highest level of detail (LOD) and exported as a wavefront obj file. Wavefront obj files can be used to store lines, polygons, and free-form curves and surfaces in ASCII format [56]. Since all geometric data in wavefront obj objects are stored as points in binary form, a plugin was designed using C++ to load these

binary data and display them on VAPS-XT. The propeller of the UAV was removed before the file was exported to avoid obstructing the pilot from seeing some points.

The application for displaying sensor images was created using OpenGL library in C++. OpenGL supports 2D and 3D graphics application programming interface (API), therefore was used to process the coordinate data received from each sensor [57]. The flowchart for carrying out this process is shown in Figure 62 below.



Figure 62 - Process for Displaying Sensor Information

The communication between the sensor and the C++ application was established using Lightweight Communications and Marshalling (LCM) [36]. The HPP file of the point cloud messages sender contain the pointField, data, height, width, point_step, and row_step. The pointField contains the definition field for the structure data representing each point received, while the data contains the point cloud data packed serially one point at a time. Each point in the data packet can be accessed using 0-based indexing. The height is always constant while the width contains all the points received. The point_step is the sum of all bytes defined in PointField. Moreover, finally, the row_step is the multiple of the width and the point_step. When communication is established between LCM sender and receiver, a point cloud handler function to initialize and load the pointcloud data is called. The information contained in each point (elevation, azimuth and range) can be accessed through the data field. After loading point cloud data, all the data was assigned to a pointer msg. By using this pointer, all information can be accessed as follows:

```
msg->num_fields
msg->height
msg->width
msg->point_step
msg->row_step
msg->data
```

After the point information has been parsed, an equation to calculate the X, Y, and Z coordinate for each point was defined using the equation (7), (8) and (9) below [58]

$$X = R\cos w * \sin \alpha \tag{7}$$

$$Y = R\cos w * \cos \alpha \tag{8}$$

$$Z = R\sin w \tag{9}$$

In equation (7), (8), and (9) and Figure 64, w, α and R represent the elevation, azimuth

and range. Since the msg->width contains the total number of points and the msg->data contain the data of each point, as shown in Figure 63, a For-loop to calculate the X, Y and Z for each point was created.



Figure 63 - GCS C++ Plugin Procedure for Displaying Sensor Points

An if-statement used before calculating the X, Y and Z was utilized because the msg->range value will always give a value of -1 each time the laser of the sensor fails to hit an obstacle. This prevents the application from calculating the coordinates for laser points that failed to hit an obstacle. The X, Y and Z coordinates calculated for each point were then parsed and displayed using OpenGL. Figure 66 below shows the LiDAR 2 and LEDDAR 1 points while Figure 65 shows the LiDAR 1 points.



Figure 64 - Data Point X, Y and Z coordinates [58]



Figure 65 - LiDAR 2 Points [52]


Figure 66 - LiDAR 1 and LEDDAR 1 Points [52]

5.2.2. Message Log

Malfunctions and errors are inevitable in any system. However, to continuously improve any mechanical system, or debug the source of problems when they occur, it is necessary for the system to be able to create a log of messages containing malfunctions and errors that occurred during each flight. For this reason, the message log panel (shown in Figure 68) was designed to not only create the log file, but also to show the type and intensity of danger of the error or malfunction in the system. It also gave the pilot the capability of saving these messages for post flight review.

Similar to the mission panel, the message log was also designed using C++ TinyXML library. Since the log messages can range from one to thousands of messages, the need to create

strings dynamically as the messages come in is necessary. Also, for easier access, a dictionary file containing all the possible messages that can occur in the system is created.

The dynamic strings loader application creates three strings and an integer each time a message is received. The first string loads the date the message was received, the second loads the time, while the last one loads the message received. The integer value, on the other hand, represents the color of the message. This color code follows the VAPS-XT color code: standard message - white=0, warning message - yellow=15 while critical message - red=2. In the header file, a public class was defined containing these objects as shown below:

```
class MessageLog{
public:
    std::string Date;
    std::string Time;
    std::string LogMessage;
    int Color;
};
```

In the CPP file, list.h library was included and a MessageLog class list was created. A variable to identify the first visible string was also defined. This string represents the first visible message on the page of the message log panel. For example, in Figure 68 below the first visible string is 1. Each time a message is received, three strings variables (Date, Time and LogMessage) and an integer (Color) are created and assigned by using the iterator of the list created as shown Figure 67 flowchart.



Figure 67 - Log Messages Creation Procedure

In designing the dictionary application for all the messages, an XML file containing the messages was created. The format of the message was written to be the same with the format used in the dynamic strings creator application stated above. A sample of a dictionary XML file containing six messages can be seen below.

```
<?xml version="1.0" ?>
<!--
Messages Definition File
<Color>
White=0 (Normal Message)
Red=2 (Critical Message)
Yellow=15 (Warning Message)
```

When an error or malfunction occurs, and the message name is identified, the application goes into the dictionary XML file and loops through all the attributes of the first child element -MessageGroup to determine the message content and color. For example, when the LiDAR sensor detects an obstacle, this error is identified as LidarMsg. The application loops through all the attributes in the MessageGroup to find LidarMsg. When LidarMsg is identified, the color and the message content is assigned to the dynamic strings. It must be noted that the dictionary XML does not contain the date and time that must be allocated to the dynamic strings. The date and time are assigned using the date and time data sent from the GPS of the UAV at the time the message is received.

The pilot may decide to clear the message in the Message log panel. The "X" icon on the top right of the panel can be used to do this. When this button is pressed, the application clears all the content in the string list iterator. On the other hand, as seen on the top left of the panel, when the save button is pressed, the application saves all the values in the string list iterator using the fstream library.



Figure 68 - Message Log Panel [52]

5.2.3. Radar

The information shown in the RADAR panel (see Figure 69 below) is streamed from the RADAR sensor attached to the UAV. Since the RADAR sensor has not been integrated into the Tandem Rotor UAV, the design of this panel was based on the information received from the virtual RADAR sensor created using Presagis Ondulus RADAR. This panel does not only stream information from the virtual sensor but also sends inputs that can be read by the sensor. There are four different modes available. First is the real beam ground map (RBGM) which shows the polar coordinates of the surroundings. The second mode is the moving target indicator (MTI) which uses the Doppler effect to differentiate moving targets from stationary ones. The rest include the weather (WX), synthetic aperture RADAR (SpotSAR), Strip-Mapping Synthetic

Aperture RADAR (StripSAR) and Inverse Synthetic Aperture RADAR (ISAR). The buttons on the right-hand side of the panel can be used to change the gain, range, and resolution of the sensor while the ones on the right are used to adjust the RADAR range and parameters in each mode.



Figure 69 - RADAR Instrument Panel [52]

5.2.4. Camera

The camera panel shown in Figure 70 below shows the streamed information from the camera attached to the UAV. In simulation mode, all the camera information is streamed from a virtual camera created using Presagis VegaPrime software. On the top right of the panel, the aircraft position and altitude is shown. To avoid confusion, ACFT (representing aircraft) is written at the of these values. On the lower right, the camera position bearing and range are shown. The vertical line represents the field of view(FOV) angle of the camera while the horizontal line represents the bearing of the camera. The calibrate button on the lower left can be

used to center the camera by moving the bearing to 90 degrees and FOV to 45 degrees. Finally, the buttons on the bottom right can be used to switch between normal and infrared view.



Figure 70 - Camera Panel [52]

CHAPTER 6: DESIGN EVALUATION AND RESULTS

6.1. GCS REQUIREMENTS REEVALUATION

In engineering, design is considered to be an iterative process because it is almost impossible for an engineer/designer to get it right the first time. The reason is, in a complex system like the tandem rotor UAV GCS, sub-tasks that needs to be completed are so pervasive that crucial information could be inadvertently left out. Thus, the need to do a sanity check after the final design is necessary. This helps to confirm if all the initial requirements set out are met and nothing was left out.

In section 2.4, in order to determine the minimum user requirements of the tandem rotor UAV GCS, a HTA analysis was performed and each final node was set as a requirement that must be met during the GCS design. Before the final evaluation was carried out to ensure that the system specific user requirements were followed during the design stage, a checklist was created showing the satisfied requirements (0). - Requirements Reevaluation Checklist

REQUIREMENT	SATISFIED
GCS must have a throttle position indicator	YES
GCS must have error and status message panel.	YES
Collective input response must be indicated,	YES
vertical speed must be shown to the pilot,	YES
rate of change of vertical speed must be shown to pilot,	YES
indicated airspeed and altitude of the aircraft must be shown,	YES
Pilot must be able to activate the hover mode	YES
Pilot must be able to control the autopilot from the GCS, which includes	YES
enabling ATS, AFCS.	
Requesting speed and altitude must also be possible from the GCS	YES
GCS must alert pilot when the sensors detects objects,	YES
The pilot must be able to declutter the LiDAR and LEDDAR points,	YES
The pilot must be able to rotate the view angle of the objects from the GCS	YES
The GCS must be able to generate and load flightpath file,	YES
Engine parameters must be shown to the pilot.	YES

Alerts on availability of GPS, connection and datalink and payloads must be	YES
available,	
Position of aircraft must be easily seen	YES
Pilot must be able to create waypoint,	YES
Must be able to set the speed (at least IAS and TAS),	YES
Must be able to set the fly mode (at least fly in/out),	YES
GCS must have the recovery button	YES
GCS must show the camera position,	YES
The GCS must display the pitch, elevation, bearing and range,	YES
The GCS must identify when the camera is in tracking mode	YES
RADAR Heading must be displayed on the GCS,	YES
The RADAR modes must be selectable (at least, RGBM, SAR and MTI),	YES
The GCS must show the RADAR range	YES
Pilots must be able to load flightpath from file,	YES
Pilots must be able to create flightpath file from scratch,	YES
Pilots must be able to name the created flightpath file,	YES
Pilots must be able to save flightpath file	YES

6.2. EVALUATION TEST SET-UP

To properly evaluate the design procedure proposed in this thesis, a human-in-the-loop experiment was conducted, and a NASA task load index was used to assess the operators' mental workload while using the GCS to control the UAV. The NASA task load index is a multidimensional rating tool for measuring participants' mental workload [29, 51]. It is based on six scales rated on a scale of 0(low) to 20 (high). The first is the mental demand scale, which is concerned with how mentally demanding and complex the task performed is. The second is the physical demand, which is mainly about how physically demanding the tasks are. The physical demand includes how often the participant had to move and how uncomfortable these movements are. The third is the temporary demand which is mainly about the time pressure the participant felt. The fourth factor, effort, is concerned with how much physical and mental efforts the participant had to make to complete the task. The fifth factor, performance, was based on the participant's self-evaluation of how well they did. Finally, the frustration level is concerned with how frustrated or irritated the participant felt while performing the tasks [29].

Ten participants were made to perform a series of tests using the simulation environment on the Tandem Rotor UAV GCS. Operators were made to fly the UAV through a series of manually created waypoints using both joystick control and the autopilot system. Before proceeding with the tests, all participants were taken through a 15-minute session on how to use the GCS. The first procedure explained was how to operate each instrument panel needed to complete the tests. The second was the behavior of the Tandem Rotor UAV. None of the participants had prior experience with a tandem rotor aircraft. Therefore, it was necessary to explain the physics and how the UAV will react to certain controls. Finally, a more detailed explanation was given on how to translate the images on the obstacle avoidance instrument panel.

Table 7 below shows the list of participants with prior flying experience.

Participant	Age	Gender	Pilot Experience	Experience
1	23	Male	Helicopter Simulator	3 Months
2	25	Male	Licensed Private Pilot	1.5 Years
3	23	Male	None	None
4	19	Male	None	None
5	29	Male	Flight Simulator	1 year
6	25	Male	None	None
7	22	Male	Helicopter simulator	2 Months
8	23	Male	Ryerson UAV Simulator	2 Years
9	28	Male	Ryerson UAV Simulator	3 Years
10	23	Male	None	None

Table 7 - Experiment Participants Profile

The set-up used for the tests can be seen in Figure 71 below. As explained in section 3.3, the set-up contains two workstations - the pilot and the payload control stations. On the pilot workstation, the UAV can be flown using the two joysticks. However, the payload workstation is dedicated to payload controls only. The pilot workstation is restricted from controlling the payloads, while the payload workstation does not have an option of controlling the UAV manually. This design was chosen for dual pilot configuration requested by the stakeholders.

Since the focus of this research is to test an individual pilot's workload, during the test a combined configuration was used. In this case, the restriction on the pilot workload was removed. In the combined configuration, the pilot can control the payload while flying the UAV in autopilot mode.



Figure 71 - GCS Complete Set-up [52]

During the first test, the participants were asked to create waypoints using the map and the mission instrument panels. After the waypoint creation process wass completed, the participants were asked questions regarding the state of the UAV before take-off. They were asked about the engine, camera, RADAR and the altitude of the UAV before take-off. During the flight, they were required to monitor an entity on the map using visuals only while simultaneously controlling the UAV.

During the second test, participants were asked to fly the UAV from the ground position by creating a takeoff waypoint and flying to a specific altitude. After completing that task, they were asked to load a flight path file named "Continuous Surveillance Mission" from the database and fly the UAV through the points using the automatic flight control system. At every point of the test, the participants were asked questions regarding the visibility of the information they are monitoring, the altitude, speed and position of the aircraft using the information shown on the GCS.

6.3. NASA TLX RESULTS

The Nasa TLX results were collected using the worksheet shown in APPENDIX F. After the completion of the tests, participants were presented with the Pairwise comparison worksheet of the six factors (mental demand, physical demand, temporal demand, performance, effort, and frustration). This comparison helped participants to state the factor that contributed to the workload they felt during the tests. During this process, each time the participant picks a factor, a value of one is assigned to that factor. The results of the Pairwise comparison can be seen in Table 8 below.

103

Participant	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration
1	2	2	4	5	2	0
2	3	1	2	0	4	5
3	3	0	5	2	4	1
4	1	2	4	5	3	0
5	3	1	5	2	4	0
6	3	1	5	4	2	0
7	3	0	5	4	2	1
8	1	0	4	2	3	5
9	1	0	3	5	4	2
10	4	2	3	0	5	1

Table 8 - Sources of Load Results

After the Pairwise comparison, the magnitude of load is compiled for each factor. Here, the participants rate each factor on a scale of 0 (low) to 20 (high). These values were then normalized on a 0 to 100 percent scale. The results of the magnitude of load can be seen in Table 9 below.

Participant	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration
1	65	40	30	30	45	25
2	70	40	50	40	85	90
3	35	5	60	35	45	25
4	15	15	30	85	25	5
5	60	10	70	25	60	10
6	20	0	20	20	10	0
7	25	0	25	20	10	5
8	20	15	40	20	15	50
9	20	10	30	50	40	30
10	25	15	15	0	40	10

Table 9 - Magnitude of Load Results

To calculate the overall workload, the adjusted rating was computed by multiplying the sources of load and the magnitude of load that each participant felt during the tests. After that, the sum of the adjusted rating was then divided by 15 to get the overall workload.

For validation, a comparison of results with the Ryerson University's reconfigurable UAV GCS (Section 1.5) was carried out. Figure 72 below shows the comparison between the average source of load participants felt during the Tandem Rotor UAV GCS test and the one recorded for the Ryerson University's reconfigurable GCS. It must be noted that when the tandem UAV GCS tests were performed, the flight dynamics model of the Tandem Rotor UAV was unstable and the UAV was difficult to control manually. On the other hand, the Ryerson GCS test used a generic F-16 flight model preinstalled in Flight SIM 13. Moreover, the tasks performed in tandem UAV GCS tests are continuous surveillance tasks, which are generally considered to be more challenging when compared to the Ryerson GCS tests that only require participants to fly through predefined waypoints.

In the source of workload results (Figure 72 below), the first point that was observed was the fact that participants felt the quick swipe gesture made the user interface less physically demanding to use when compared to the Ryerson reconfigurable GCS. Most participants who were involved in the Ryerson GCS test felt that the effort it took to bring up the instrument panels and scale them manually significantly contributed to their workload and performance throughout the tests. However, in the case of the Tandem Rotor UAV GCS, the temporal demand was regarded as the significant factor that contributed to their workload. Although the Tandem UAV GCS helped accomplish the task, participants felt the flight model of the Tandem Rotor UAV was not stable, making tasks much more difficult when compared to the Ryerson UAV.

When the source of workload results was further broken down, it was found that inexperienced pilots felt the factor that had the least effect on their workload source was the frustration, regardless of the difficulties encountered flying the tandem rotor UAV GCS. Contrary to the average source of workload results, inexperienced pilots felt that the user

105

interface of the tandem rotor UAV GCS alleviated their frustrations. On the other hand, experienced pilots felt the factor that had the least effect was the physical demand. Figure 73 and Figure 74 show the sources of workload for inexperienced and experienced pilots.



Figure 72 - Average Source of Load



Figure 73 - Average Source of Workload for Inexperienced Pilots



Figure 74 - Average Source of Workload for experienced Pilots

Similarly, comparing the magnitude of load results (Figure 75), participants felt that the Tandem Rotor UAV was helpful, even though the flight model was unstable. Less frustration, effort, and physical demand were recorded compared to the Ryerson GCS. Also, since the tandem UAV GCS had more complex features such as mission, avoidance, autopilot and map system, most participants found it to be more mentally demanding when compared to the Ryerson GCS.

Similarly, by grouping the pilots by their experience, it was found that inexperienced pilots were more concerned by their performance during the tests, while experienced pilots felt the tests required more effort to complete when compared to other simulators they have controlled. These categories of results can be seen in Figure 76 and Figure 77.



Figure 75 - Average Magnitude of Load



Figure 76 - Average Magnitude of Load for Inexperienced Pilots



Figure 77 – Average Magnitude of Load for experienced Pilots

Regarding the average adjusted rating (Figure 78), the tandem rotor UAV GCS was found to be 81% less physical demanding, 7.3% less effort, and 27.7% less frustrating when compared to the Ryerson GCS. However, a 25.7% increase was recorded based on the mental demand, 56.1% increase on temporal demand, and 9.4% increase on operator performance.





The overall workload value recorded in both GCS tests can be seen in Figure 79 below. Even though the flight model of the Tandem Rotor UAV was unstable and the tasks were significantly more difficult, only a 6.5% increase in workload was recorded. However, all participants who took the tandem UAV GCS tests felt that the flight dynamics of the Tandem Rotor UAV had a significant effect on their temporal demand and performance. They mentioned that it took more time to complete tasks due to the poor stability of the UAV, even in autopilot mode. For this reason, the poor stability of the UAV was considered in the workload result by making the temporal demand equal in the average adjusted rating of both tests. As shown in Figure 80 below, by equating this factor, a reduction by 11.1% in overall workload was recorded in favor of the tandem UAV GCS.







Figure 80 - Normalized Overall Workload Rating

6.4. COMPLIANCE RATING RESULTS

In APPENDIX A, publicly available human factors standards and guidelines were compiled. However, to show the design compliance with this document and evaluate its effect on the pilot workload, all the standards and guidelines in the document were grouped into five different categories. The first category is comprised of the standards and guidelines relating to physical demand of the operator. The rest are standards relating to mental demand, performance, effort and frustration of the operator. Regarding the rating system, two values (0 and 1) were used to rate each standard or guideline. "0" signifies that the standard was not followed during the design process, while "1" shows the opposite. APPENDIX B shows the compliance rating results for both the tandem rotor UAV GCS and the Ryerson reconfigurable GCS. The summary of the result can be seen in Figure 81.



Figure 81 - Compliance Rating Results

Since the compiled human factors and guidelines compliance rating does not involve a human-in-the-loop testing and factors such as difficulty of mission are not considered, it is quite difficult to create a relationship between the rating and the pilot workload recorded using the NASA task load index. However, similarity between the results was recorded in terms of the physical demand.

CHAPTER 7: CONCLUSION AND FUTURE WORKS

7.1. CONCLUSION

The technology behind GCS design has slowly grown from simple control station to an advanced workstation that can be used to carry out deadly missions. The reason behind this growth is the rapid improvement in technology in terms of software and hardware development. Software can now be used to develop tasks that were regarded as "too difficult" in the past. Also, hardware and processing power have greatly expanded to accommodate almost any design that is imaginable. However, the advancement in technology also creates room for errors. Errors are inevitable, but in aviation, efforts are always made to completely avoid them, especially the ones that could lead to accidents or possibly life lost. Therefore, factors such as workload, stressful work environment, and other aspects that could lead to operator error are dealt with utmost seriousness. Thus, human factors and a human-centered design plan must be well planned and implemented from the preliminary stage. Human factors consideration in aviation has proven to be useful. This is because when it is properly planned and implemented, the pilot well-being is considered at every stage of the design. However, it has not been strictly applied in GCS design compared to manned aircraft cockpit design. The reason behind this bias can sometimes be related to budget or simply ignorance.

This study focused on designing a GCS for a Tandem Rotor UAV by creating a humancentered design plan using human factors standards and guidelines. Emphasis was made on user requirements gathering, and human factors methods such as hierarchical task analysis (HTA) were used to transform these requirements into design requirements that served as a starting point for the GCS design. After the design requirements stages were completed, the actual design

113

was carried out using C++ programming and COTS software called VAPS-XT (a product of Presagis Inc.). Each stage of the design followed the compiled human factors standards and guidelines, and it was found that the design approach used in this study significantly reduced the pilot physical demand, effort and frustration when compared to similarly designed GCS.

7.2. FUTURE WORKS

The proposed design approach has proven to not only help meet the primary users' and stakeholders' requirements, but also to improve some factors contributing to pilot workload. However, numerous improvements could be made to make the design closer to being perfect.

The first area that needs to be upgraded is the processing power of the computer used for running the simulation and data streaming. The sensor simulation plug-in sends 300,000 points at every rotation. This could sometimes be really hard on the computer, especially when the UAV is in a terrain that has many obstacles.

The second area that could be improved is redesigning the LCM to GCS code to run as an external application that gets all the commands and transfer them to the GCS. Now, the code is implemented in the GCS which causes a bit of lag when the real UAV commands are sent to the GCS. This is because the GCS waits for all messages before refreshing. Increasing the LCM sender frequency is also a good option.

In terms of the avoidance system, it would be nice if the panel could automatically declutter itself based on received points. This would eliminate the need for the pilot to toggle individual sensors within the panel. Also, the ability to show the nature of the surface using color opacity would be very useful for the pilot.

Finally, in the evaluation stage, the accuracy of the NASA TLX results could be improved by increasing the number of participants. Also, during the comparison with Ryerson GCS, the results would be more accurate if both GCS have similar features rather than one being superior to the other. Using similar flight models would also help.

APPENDIX

APPENDIX A - COMPILED HUMAN FACTORS DESIGN

STANDARDS AND GUIDELINES

GENERAL REQUIREMENTS

- 2.2.1 Design for simplicity. The system or equipment design shall be as simple as possible, consistent with the desired human machine system functions, and compatible with the expected maintenance and operational concepts. [Source: MIL-STD-1472F, 1999]
- 2.2.3 Make functions obvious. Systems and equipment should be designed so that basic system functions are obvious to the user. [Source: Martin & Dong, 1999]
- 2.3.1 Make design consistent. Systems and equipment should be designed to be consistent, appearing, behaving, and responding the same throughout. [Galitz, 1993]
- 2.4.2 Maintain identical interfaces for identical functions. Equipment with identical functions shall employ identical or similar interfaces. [Source: MIL-STD-1472F, 1999]
- 2.4.3 Make controls, displays, marking, coding, labeling, and arrangement uniform. Controls, displays, marking, coding, labeling, and arrangement schemes shall be uniform for common functions of all equipment. [Source: MIL-STD-1472F, 1999]
- 2.4.5 Standardize terminology, look, and feel. Systems and equipment should have standardized terminology, look, and feel. [Source: Avery & Bowser, 1992]
- 2.6.3 Use familiar terms and images. Systems and equipment should use terms and images familiar to the user. [Source: Martin & Dong, 1999]
- 2.6.4 Design within user abilities. The design of systems, equipment, and facilities shall conform to the capabilities and limitations of the users to operate and maintain it in its operational environment and not exceed user capabilities. [Source: MIL-HDBK-759C, 1995]
- 2.6.10 Design for 5th to 95th percentile. Systems and equipment shall be, at minimum, designed for personnel from the 5th through the 95th percentile levels of the human physical characteristics that represent the user population. [Source: MIL-STD-1472F, 1999]
- 2.6.13 Provide enough flexibility for different user skill levels. Systems and equipment should be flexible enough to accommodate the interaction styles of users with differing skill and experience levels. [Source: Ameritech, 1998]

AUTOMATION

- 3.1.1 Minimum automation human factors requirements. An automated system should
- provide sufficient information to keep the user informed of its operating mode, intent, function, and output; inform the user of automation failure or degradation; inform the user if potentially unsafe modes are manually selected; not interfere with manual task performance; and allow for manual override. [Source: Veridian (AHCI), 1998; Billings, 1997]
- 3.1.10 Avoid increasing demands for cognitive resources. Automation should not increase the demands for cognitive resources (thinking or conscious mental processes). [Source: Bainbridge, 1983; Parasuraman & Riley, 1997; Wiener & Curry, 1980; Woods, 1996]
- 3.1.12 Prevent distraction from operations. User interaction with automation shall not require the user to take significant amounts of attention away from the primary task. [Source: Danaher, 1980]
- 3.1.16 Provide easy data access. Data that are needed by the user shall be easily accessible. [Source: NUREG/CR-6105, 1994; NUREG-0700, 1996]
- 3.1.17 Prompt for data entry format. The automated system should prompt users as to the correct data entry format. [Source: Billings, 1996]
- 3.1.23 Make systems easy to understand and use. Automated systems and associated integrated information displays should be intuitive, easy to understand, and easy to use. [Source: Billings, 1991; Sarter & Woods, 1994; Woods, 1996]

- 3.4.1 Keep it simple. The automation interfaces should represent the simplest design consistent with functions and tasks of the users. [Source: NUREG-0700, 1996]
- 3.4.2 Provide interface consistency. Human interfaces in automation programs and systems shall have a high degree of consistency. [Source: NUREG-0700, 1996]
- 3.4.5 Make location status obvious. Interfaces and navigation aids shall make it easy for users to know where they are in the data space. [Source: NUREG/CR-6105, 1994; NUREG-0700, 1996]
- 3.5.1 Increasing user trust in automation. To increase user trust in automation, automation performance should be
- reliable and predictable with minimal errors,
- robust (able to perform under a variety of circumstances),
- familiar (use terms and procedures familiar to the user), and useful. [Source: Lee & Moray, 1992; Lerch & Prietula, 1989; Masalonis & Parasuraman, 1999; Muir, 1987 (as found in Riley, 1996); NRC, 1998]
- 3.5.4 Prevent interference with user tasks. The automated system shall not interfere with task performance. [Source: Andes, 1987]
- 3.6.1 Clearly identify modes and functions. When control, display, or automation functions change in different modes of operation, mode and function identification and status should be clear. [Source: Billings, 1991; Sarter & Woods, 1995]
- 3.7.6 Integrate displays. When users must monitor multiple displays, important events should occur in the same display in order to promote effective monitoring performance. [Source: Warm et al., 1996]
- 3.7.8 Provide indication of monitoring. Automated systems that are without incident for long periods of time should provide some type of indication that the automation is still monitoring the system. [Source: AHCI, 1998]
- 3.8.2 Make failures apparent. Automation failures shall be made unambiguously obvious to the user. [Source: AHCI, 1998; Billings, 1991]
- 3.8.8 Make sensor status verifiable. The status of sensors on replacement units shall be verifiable with respect to accuracy and proper operation. [Source: NASA-STD-3000A, 1989]

DISPLAY

- 5.1.2.4 Place the top of the screen below eye level. The top of the screen should not be above the viewer's eye level. [Source: DOE-HFAC 1, 1992]
- 5.1.2.8 Group task-related displays together. All displays necessary to support a user's activities or sequence of activities should be grouped together. [Source: MIL-STD-1472F, 1999]
- 5.1.2.9 Arrange according to function and sequence. Displays shall be arranged in relation to one another according to their sequence of use or the functional relations of the components they represent. [Source: MIL-STD-1472F, 1999]



Exhibit 5.1.2.10 Optimum vertical and horizontal visual fields

- 5.1.2.10 Locate critical displays in central visual field. Critical or frequently used displays shall be located in the central visual field, as illustrated in Exhibit
- 5.1.2.10, and occupy a privileged position in that field (e.g., the top or left-most position). [Source: MIL-STD-1472F, 1999]

VISUAL INDICATORS

- 6.2.1.1.3 Consistency. Visual coding shall be consistent within a system or unit of equipment and between similar units of equipment. [Source: MIL-STD-1472F, 1999]
- 6.2.1.1.4 Visual coding of priority levels. Visual signals should be coded to indicate the priority level of the signal. [Source: MILSTD- 1472F, 1999]
- 6.2.1.1.5 Emergency conditions. Flashing red shall be used to denote emergency conditions that require immediate user action to avert impending injury, equipment damage, or both with an approximately equal on and off time flashing rate from three to five flashes per second. [Source: MIL-STD-1472F, 1999]
- 6.2.2.1.2 Limited use of lights and illuminated displays. Lights and illuminated indicators shall be used sparingly, reserved for displaying only that information necessary for effective system operation. [Source: MIL-STD-1472F, 1999]
- 6.2.2.1.3 Meaning of illumination. Lights, including those used in illuminated push buttons, shall indicate equipment response and not simply control position. [Source: MIL-STD-1472F, 1999]

Color	Use	Examples
flashing red	to indicate an emergency condition that requires immediate action to avert impending injury, equipment damage, or both	
red	to indicate that (1) the system or a portion of the system is inoperative, or (2) successful task completion is not possible until appropriate corrective or override action is taken	"no-go" "error" "failure" "malfunction"
yellow	to indicate (1) a marginal condition, (2) an unexpected delay, (3) that caution is necessary, or (4) that rechecking is necessary	
green	to indicate that (1) equipment is "in tolerance," (2) conditions are satisfactory, or (3) it is all right to proceed	"go ahead" "in tolerance" "ready" "function activated"
white	to indicate system conditions that do not have "right" or "wrong" implications	(1) indicating which of several functions has been selected, (2) indicating a transitory condition such as an action or test in progress, provided such indications have no implications of success or failure
blue	to advise only	

Exhibit 6.2.2.1.27 Color-coding of transilluminated displays

VISUAL INDICATOR-CONTROL INTEGRATION

• 6.3.1.2 No obstruction. The control itself and the user's hand should not obscure the visual indicator. [Source: MIL-STD-1472F, 1999]

ALARMS

- 7.1.1.1 When to use. If equipment is not regularly monitored, an audio alarm shall be provided to indicate malfunctions or conditions that would cause personnel injury or equipment damage. [Source: Department of Defense (MIL-STD-1472F), 1999]
- 7.1.1.1 When to use. If equipment is not regularly monitored, an audio alarm shall be provided to indicate malfunctions or conditions that would cause personnel injury or equipment damage. [Source: Department of Defense (MIL-STD-1472F), 1999]
- 7.1.1.3 Alarm system characteristics. Alarms systems should
 - alert the user to the fact that a problem exists,
 - inform the user of the priority and nature of the problem,
 - guide the user's initial responses, and
 - confirm in a timely manner whether the user's response corrected the problem. [Source: Nuclear Regulatory Commission (NUREG-0700), 1981]
- 7.1.2.8 Distinguish caution signals. Caution signals shall be readily distinguishable from warning signals. [Source: MIL-STD- 1472F, 1999]

COMPUTER-HUMAN INTERFACE

- 8.13.5.1 Consistent appearance. All push buttons in a window should have the same size and shape. [Source: DON UISNCCS, 1992]
- 8.13.5.2 Minimum push button size. The size should accommodate the largest label. [Source: DON UISNCCS, 1992]

- 8.13.5.3 Labels. A push button shall have either a text or graphic label. [Source: DON UISNCCS, 1992]
- 8.13.5.4 Consistent labels. Push button labels shall be consistent throughout an application and related applications. [Source: DON UISNCCS, 1992]
- 8.13.5.5 Text label length. Push button labels should be short and unambiguous. [Source: DON UISNCCS, 1992]
- 8.13.5.6 Push button label. The push button label should describe the results of pressing the button and reflect the action that will be taken by the application rather than the user. [Source: DON UISNCCS, 1992]
- 8.13.5.7 Activating a push button. A user shall be able to activate a push button by moving the pointer onto the button and pressing the appropriate pointer button. [Source: DON UISNCCS,
- 1992]
- 8.14.3.6.1 Map window elements. A map window should include
 - a title;
 - o identifying information such as coordinates, area, and scale;
 - the map itself;
 - a continuous coordinate indicator that states the pointer location; and
 - appropriate controls.
 - [Source: DON UISNCCS, 1992]
- 8.13.5.8 Activated push buttons. The push button shall be highlighted while the pointer button is depressed. [Source: DON UISNCCS, 1992]

SCREEN DESIGN

- 8.1.1.1 Simplicity. Information should be presented simply and in a well-organized manner. Ways to achieve simplicity include the following:
 - The screen should appear to be orderly and clutter-free.
 - Information should be presented in consistent, predictable locations.
 - The language used should be plain and simple.
 - The means for moving around the screen and to related screens should be simple.
 - Interrelationships should be indicated clearly. [Source: Avery & Bowser (DOE HFDG ATCCS V2.0), 1992; Avery & Bowser (DOD HCISG V2.0), 1992]
- 8.1.1.2 Minimal information density. The information density (the amount of information per unit area) of a screen should be minimized by presenting only information that is essential to a user at any given time. [Source: DOE HFDG ATCCS V2.0, 1992; DOD HCISG V2.0, 1992]
- 8.1.1.5 Whole data sets. Whenever possible, users should be able to see the whole data set of interest, such as an entire page, map, or graphic. [Source: Department of Defense (MIL-HDBK-761A), 1989]
- 8.1.1.6 Minimizing the user's short-term memory load. A window should contain all relevant information and should allow a user to complete the task without having to refer to additional information. [Source: Department of the Navy (DON UISNCCS, 1992), 1992]
- 8.1.1.7 Vocabulary. The words used in all non-editable text shall be task-oriented and familiar to users. [Source: DON UISNCCS, 1992]
- 8.1.1.8 Date and time information. When task performance requires or implies the need to assess the timeliness of information, the display should include time and date information associated with the data. [Source: MIL-HDBK-761A, 1989]

CONTEXT

- 8.1.2.3 Highlighting. When a user is performing an operation on a selected object in a display, that object shall be highlighted. [Source: DOE HFDG ATCCS V2.0, 1992; MIL-HDBK-761A, 1989]
- 8.1.2.5 Distinctive position and format. Displayed options, context information, command entry areas, prompts, advisory messages, and other displayed items (for example, titles and time signals) relevant to transaction control shall be distinctive in location and format. [Source: MIL-HDBK-761A, 1989]
- 8.1.2.6 operational modes, the current mode shall be continuously indicated to a user. [Source: DOE HFDG ATCCS V2.0, 1992; MILHDBK-761A, 1989]
- 8.1.2.8 No repetitive entry of data. A user shall not have to reenter data already entered in the current application session or control session. [Source: DOE HFDG ATCCS V2.0, 1992]

• 8.1.2.9 Action history. An application should maintain a summary of the transactions that produced the current context and display it at a user's request with an UNDO feature linked to each step in the action history. [Source: DOE HFDG ATCCS V2.0, 1992; MIL-HDBK-761A, 1989]

FORMAT

- 8.1.3.1 Title. Every screen shall have a title or header at the top that is separate and distinguishable from the body of the screen and describes briefly the contents or purpose of the screen. [Source: DOE HFDG ATCCS V2.0, 1992; DOD HCISG V2.0, 1992]
- 8.1.3.3 Minimal visual competition. Information on a display screen should be organized so that visual competition among distinct items of information is minimized. [Source: DOE HFDG ATCCS V2.0, 1992; DOD HCISG V2.0, 1992]
- 8.1.3.4 Arrangement of screen elements. Screens should be arranged so that there is a clear differentiation between instructions and data. [Source: DOE HFDG ATCCS V2.0, 1992;DOD HCISG V2.0, 1992]
- 8.1.3.6 Matching layout to task. Application designers should design the screen layout so that users can move quickly and easily among items and can manipulate objects in ways that support task performance. [Source: DON UISNCCS, 1992]
- 8.1.3.7 Minimal user effort. Screens should be designed to minimize both eye and pointer movement and the number of keystrokes required to complete a task. [Source: DOE HFDG ATCCS V2.0, 1992; DOD HCISG V2.0, 1992]
- 8.1.3.9 Priority of displayed information. Information should be prioritized so that the most important or critical information is displayed all the time and less important or critical information can be displayed upon a user's request. [Source: DOE HFDG ATCCS V2.0, 1992; DOD HCISG V2.0, 1992]
- 8.1.3.10 User control. Users should be able to control the amount, format, and complexity of displayed data as necessary to meet task requirements. [Source: MIL-HDBK-761A, 1989]
- 8.1.3.13 Primary viewing area. Information that is particularly important or that requires immediate user response shall be displayed in the user's primary viewing area. [Source: DOD HCISG V2.0, 1992]
- 8.1.3.17 Integrated information. When a user needs a variety of data to complete a task, those data should be provided in an integrated window or display, not partitioned in separate windows or displays. [Source: DOE HFDG ATCCS V2.0, 1992; DOD HCISG V2.0, 1992; Smith & Mosier, 1986]

TOUCH SCREEN

- 5.4.6.3 Positive indication. A positive indication of touch-screen actuation shall be provided to acknowledge the system response to the control action.
- 5.4.6.6 Critical tasks. Where a touch screen control is used for a critical task, system response shall require confirming an additional, confirmatory action to ensure that the control actuation is, in fact, intended. If this is impractical, multiple touch actuation shall be incorporated.
- 9.4.2.3 Positive indication. A positive indication of touch-panel activation shall be provided to acknowledge the system response to the control action. [Source: MIL-STD-1472D, 1989; DOE-HFAC1, 1992]
- 9.4.2.4 Dimensions and separation. The dimensions and separation of responsive areas of the touch panel shall not exceed the maximum and minimum values given in Exhibit 9.4.2.4. [Source: MIL-STD-1472D, 1989; DOE-HFAC1, 1992]
 - Note. The maximum values listed in the Exhibit apply to logically grouped touch panel responsive areas. An adverse environment may warrant larger sizes and separations.



Exhibit 9.4.2.4 Touch panel responsive area dimensions

- 9.4.2.5 Display feedback. Display of user command or action feedback for touch panels shall not exceed 0.25 seconds. [Source: MIL-STD-1472D, 1989]
- 9.4.2.6 Minimal parallax. Touch-interactive devices should be selected and mounted to minimize parallax problems. [Source: Avery & Bowser (DOE HFDG ATCCS V2.0), 1992]
- 9.4.2.7 Minimal specular glare. Touch-interactive devices should be selected and mounted to minimize specular glare. [Source: DOE HFDG ATCCS V2.0, 1992]

LOITER PATTERN

• L1.1



Figure B1 - 11: Loiter Pattern

MONITORING SYSTEM

- T_1.1.1 The GCS should enable the pilot to monitor which entity has control of the aircraft and to what extent the entity has control. (Access 5 (2006)
- T_1.1.2 If an on-board camera is used for flight control tasks, the GCS should enable the pilot to center the field of view of the camera.
- P_1.1.1 The GCS should not enable the pilot to disengage automation in flight if the aircraft will depart from controlled flight as a result.
- P_1.1.2 The GCS should prevent multiple operators from operating the same application/procedures at any one time. (NATO, 2004)
- P_1.1.3 The GCS should provide the ability to allow other operators to view the status of aircraft systems. (NATO, 2004)

CONSUMABLE RESOURCES OF AIRCRAFT

- T_1.2.1 The GCS should enable the pilot to monitor the status of consumable resources.
- I_1.2.1 The GCS should provide the pilot with information on the status of consumable resources.

SYSTEM CONFIGURATION

- T_1.3.1 The GCS should enable the pilot to perform ground station performance checks.
- T_1.3.2 The GCS should enable the pilot to perform a pre-flight check on an alternate control station, or confirm that this check has been performed. (RTCA, 2007)

- T_1.3.3 The GCS should enable the pilot to check the control station for damage and function.
- T_1.3.4 The GCS should enable the pilot to monitor the performance of GCS support services, e.g. air conditioning and electrical power.
- I_1.3.1 The GCS should provide the pilot with health and status information on the GCS.

COLLISION AVOIDANCE

- T_1.4.1 The GCS should enable the pilot to identify the threat of an impending collision of the UA with other aircraft, terrain, or objects.
- T_1.4.2 The GCS should enable the pilot to recognize the need for a UA evasive maneuver to avoid an impending collision of the UA with other aircraft, terrain, or objects.
- T_1.4.3 The GCS should enable the pilot to determine an appropriate maneuver to avoid a collision of the UA with other aircraft, terrain, or objects.
- T_1.4.4 The GCS should enable the pilot to execute a UA collision avoidance maneuver.
- T_1.4.5 The GCS should enable the pilot to communicate with ATC about a collision avoidance maneuver and other departures from the assigned flight path.
- T_1.4.6 The GCS should enable the pilot to return the UA to the assigned flight path after maneuvering to avoid a collision.
- T_1.4.7 If the UA is capable of making an autonomous collision avoidance maneuver, the GCS should enable the pilot to monitor the maneuver.
- T_1.4.8 If the UA executes an autonomous collision avoidance maneuver, the GCS should enable the pilot to smoothly regain control of the UA at the conclusion of the maneuver.
- I_1.4.1 The GCS should provide an alert to the pilot when there is a threat of the UA colliding with another aircraft, terrain, or objects. The alert must be provided in time for the pilot to effectively respond to make the UA avoid the collision.
- I_1.4.2 The GCS should provide information about terrain or ground-based objects within proximity of the projected UA flight path and may become a threat for UA collision.
- I_1.4.3 The GCS should provide the pilot with the information necessary to detect aircraft, obstructions or people while the UA is moving on the ground. This information may be provided through a camera located on the aircraft, or Closed Circuit Television (CCTV) cameras located on the ground.
- I_1.4.4 The GCS should provide the pilot with the information necessary to detect obstructions that may affect launch or takeoff. This information may be provided through a camera located on the aircraft, or CCTV cameras located on the ground.
- I_1.4.5 The GCS should provide the pilot with the information necessary to detect obstructions that may affect approach and landing. This information may be provided through a camera located on the aircraft, or CCTV cameras located on the ground.
- I_1.4.6 The GCS should provide the pilot information about the likelihood of the UA colliding with the upcoming threat so that the pilot will be able to make a decision about the need to take evasive action to avoid a collision.
- I_1.4.7 The GCS should provide the pilot with a prediction of the time available until the UA would collide with the threat aircraft, object, or terrain.
- I_1.4.8 The GCS should provide information about the aircraft surrounding the UA and the collision threat to help in making a decision about maneuvers that would not cause additional risks for collision.
- I_1.4.9 The GCS should provide information about the capabilities of the UA for making evasive maneuvers in the current UA situation. This information should include at least the following:
- I_1.4.9a Possible maneuvers that can be made by the UA in the current situation e.g. climb, descend, turn within a certain radius.
- I_1.4.9b Time for the UA to accomplish the maneuvers e.g. how long until the UA reaches a certain turn radius or climb attitude.
- I_1.4.10 The GCS should provide the pilot with information necessary to quickly identify the current state, mode, or setting of all controls that are used to send flight commands to the UA.
- I_1.4.11 The GCS should provide the pilot with information on the flight path that had been assigned to the UA prior to the evasive maneuver.
- I_1.4.12 The GCS should provide information about the necessary UA trajectory needed to return to the assigned flight path. This should include the necessary UA heading and altitude changes.

- I_1.4.13 If an autonomous collision avoidance maneuver is carried out, the GCS should alert the pilot that the maneuver is underway, and must notify the pilot when the maneuver is concluded.
- C_1.4.1 The GCS should provide a control to cancel the collision alert if it will be ongoing and distract the pilot in accomplishing other tasks.
- C_1.4.2 Flight controls should be provided to enable the pilot to rapidly command the UA to execute an effective maneuver to avoid an impending collision. The controls should be readily available at all times and must be designed to enable the pilot to make the command to the UA in the time needed to perform the collision avoidance maneuver. The flight controls must include means to control:
- C_1.4.2a UA attitude.
- C_1.4.2b UA heading.
- C_1.4.2c UA speed and/or thrust.
- P_1.4.1 Information and controls should be readily accessible for the pilot to recognize and accomplish collision avoidance maneuvers.
- P_1.4.2 Collision avoidance alerts must attract the pilot's attention in all expected lighting and operating conditions.
- P_1.4.3 Time-consuming or complicated sequences of actions (e.g. involving multiple levels of menu structures) must not be necessary to accomplish collision avoidance maneuvers.
- P_1.4.4 Primary flight controls should be designed in a manner for the pilot to quickly execute critical collision avoidance maneuvers in all expected operating conditions.

MANAGE CONTROL LINK STATUS

- T_1.5.1 The GCS should enable the pilot to confirm spectrum availability before selecting link.
- T_1.5.2 The GCS should enable the pilot to select the appropriate communication mode (e.g. terrestrial/satellite, frequency).
- T 1.5.3 The GCS should enable the pilot to maintain awareness of selected communication mode.
- T_1.5.4 The GCS should enable the pilot to confirm that communication link is effective, and established with the correct UA.
- T_1.5.5 The GCS should enable the pilot to identify if more than one control station is linked with the UA.
- T_1.5.6 The GCS should enable the pilot to maintain awareness of link strength, or link abnormalities.
- T_1.5.7 The GCS should enable the pilot to maintain awareness of link latency, where relevant.
- T_1.5.8 The GCS should enable the pilot to anticipate link degradations or diminished link strength.
- T_1.5.9 The GCS should enable the pilot to maintain an awareness of the geographic limits of the link and potential obstructions to signal.
- T_1.5.10 The GCS should enable the pilot to maintain awareness of crew actions or control inputs that could interrupt or degrade the link.
- T_1.5.11 The GCS should enable the pilot to respond to interference with the signal, (e.g. other users of frequency, jamming attempts).
- T_1.5.12 The GCS should enable the pilot to change the link during flight operations as necessary.
- T_1.5.13 The GCS should enable the pilot to assess link strength and quality before switching link.
- T_1.5.14 The GCS should enable the pilot to define the duration of a loss of link that must occur before the lost link alert is activated, or the UA enters its lost link procedure.
- T_1.5.15 The GCS should enable the pilot to manage resumption of the signal after a lost link.
- I_1.5.1 The GCS should be capable of providing the pilot with predictive information on the quality and strength of a C2 link before the link is actively used to control the UA.
- I_1.5.2 The GCS should provide information to enable the pilot to identify which C2 link settings are active (e.g. selected frequency, satellite vs terrestrial).
- I_1.5.3 The GCS should provide the pilot with information to confirm that effective control is established with the correct UA.
- I_1.5.4 The GCS should provide the pilot with information on the geographic limits of the link.
- I_1.5.5 The GCS should provide the pilot with information on spectrum activity from a spectrum analyzer.
- I_1.5.6 The GCS should alert the pilot when the UA is approaching an area where link is likely to be lost.
- I_1.5.7 The GCS should alert the pilot when the link is lost.
- I_1.5.8 The UA will transmit a pre-determined transponder code when the link is lost.
- I_1.5.9 The GCS should provide information to enable the pilot to monitor the strength of the link.

- I_1.5.10 The GCS should alert the pilot whenever the C2 link experiences interference, whether resulting from natural phenomena, payload or other equipment associated with the UAS, or human activities (such as jamming or other users on frequency).
- I_1.5.11 The GCS should display to the pilot the source of downlink transmissions. (Access 5,2006)
- I_1.5.12 Where relevant, the GCS should provide the pilot with information on link latency, in milliseconds.
- I_1.5.13 The GCS should provide information to enable the pilot to anticipate link degradations or diminished link strength. This information may include link footprint, including areas that may be affected by terrain masking.
- I_1.5.14 The GCS should provide information to enable the pilot to manage link security.
- I_1.5.15 The GCS should inform the pilot when a lost link is resumed.
- C_1.5.1 The GCS should enable the pilot to select the communication mode (e.g. terrestrial/satellite, frequency, transmission power).
- C_1.5.2 The GCS should provide a control to enable the pilot to request a link status report.
- C_1.5.3 If antenna selection is performed by the pilot, then the GCS should support an external command to set the antenna used for communication.
- C_1.5.4 The GCS should enable the pilot to set the duration of a link outage that must occur before a lost link response is triggered.
- P_1.5.1 "There must be an alert for the UAS crew, via a clear and distinct aural and visual signal, for any total loss of the command and control data link". (NATO, 2009)
- P_1.5.2 The aural warning for lost control link should be a unique sound, not also used to signify other conditions.
- P_1.5.3 The maximum range of the C2 datalink (datalink footprint) for all altitudes and directions relative to the signal source should be presented visually to the pilot, overlaid on a map display.
- P_1.5.4 Areas where the C2 link (datalink footprint) are predicted to be masked by terrain should be displayed on the C2 datalink display.
- P_1.5.5 If the datalink footprint can be suppressed, it should be automatically displayed when the UA is approaching a location where a loss of link is likely.
- P_1.5.6 The C2 datalink footprint should be easily distinguishable from other footprints that may be present on the operator map display. (NATO, 2004).
- P_1.5.7 If the payload utilizes a link separate to the aircraft control link, any display of payload link quality should be separate and clearly distinguishable from displays for the aircraft control link.
- P_1.5.8 If an aural warning is used to indicate loss of payload link, the sound should be dissimilar to that used to indicate loss of control link.
- P_1.5.9 Security features designed to prevent unapproved access (logon and logoff functions) should not result in inadvertent lockouts of authorized personnel.
- P_1.5.10 The GCS, in combination with the other elements of the UAS should comply with control link latency (time from initiation of a maneuver to a measurable response by the UA) requirements that are established at a level similar to manned aircraft. (FAA, 2013b)
- T_2.1.1 The GCS should enable the pilot to monitor and control the position of the UA when on the ground and in the air.
- T_2.1.2 The GCS should enable the pilot to ensure that both the runway and approach path are clear of traffic before taxiing onto the active runway. (FAA, 2013b)
- T_2.1.3 "The UAS shall be capable of transitioning from an instrument approach procedure to a safe landing, either by visual reference of a flight crewmember at the airport or by other means acceptable to the FAA".

(FAA, 2013b)

- I_2.1.1 UA position in airspace. The GCS should provide a representation of the UA within the airspace. This information should provide:
 - I_2.1.1a Representation of UA within the airspace.
 - \circ I_2.1.1b Heading of UA.
 - \circ I 2.1.1c Altitude of UA.
 - \circ I_2.1.1d Speed of UA.
 - \circ I_2.1.1e Attitude of UA.

- I_2.1.1f Position of UA relative to other aircraft, terrain, and obstacles.
- I_2.1.2 Programmed flight plan and predicted flight path of UA. The GCS should provide a representation of the predicted flight path of the UA based on the flight plan programmed into the flight management system based on the assigned flight clearance. This information should include:
 - I_2.1.2a Indication of UA current position along programmed flight path.
 - \circ I_2.1.2b.Predicted flight path relative to UA and other traffic, terrain, and obstacles.
 - \circ I_2.1.2c Distance to waypoints along flight path.
 - \circ I_2.1.2d Indication of position in flight path when new commanded altitude will be attained.
 - I_2.1.2e Indication of turning radius and path when making turns along flight path.
- C_2.1.3 The GCS should enable the pilot to maneuver the UA.
- P_2.1.1 The map display should be able to support a variety of map types including aeronautical charts and presentations of Digital Terrain Elevation Data (DTED).
- P_2.1.2 The presentation scale of the map should be selectable. Continuous scaling is preferred to discrete. (NATO, 2004)
- P_2.1.3 The pilot should be able to derive the scale of the map from the display. (NATO, 2004)
- P_2.1.4 The map display should enable the pilot to customize the Aircraft's Information Trail. (NATO, 2004)
- P_2.1.5 The map display should be configurable to "North up" or "Track up".
- P_2.1.6 If control is via a terrestrial radio, the location of (or direction to) the ground transmitter/receiver should be shown on the map.
- P_2.1.7 Primary flight controls for controlling the UA (heading, attitude, speed) should be available at all times through dedicated physical controls. If the use of software-based controls cannot be avoided, then the controls should be immediately accessible at the top level of the control interface. (NATO, 2009).
- P_2.1.8 Map displays should have means to select the scale of the map to be presented. The scales presented on the maps should be evident to the pilot. (NATO, 2004)
- P_2.1.9 The pilot should have the means to customize the information trail for an aircraft shown on the traffic display. (NATO, 2004)

CLEAR OF TERRAIN, AIRSPACE BOUNDARIES AND WEATHER

- T_2.2.1 The GCS should enable the pilot to ensure that both the runway and approach path are clear of traffic before taxiing onto the active runway. (FAA, 2013b)
- T_2.2.2 The GCS should enable the pilot to "observe" and comply with signage and warning lights during surface operations. (FAA, 2013b)
- T_2.2.3 The GCS should enable the pilot to monitor weather that has the potential to affect the flight. (RTCA, 2007)
- T_2.2.4 The GCS should enable the pilot to avoid weather that has the potential to affect the flight.
- $T_{2.2.5}$ The GCS should enable the pilot to avoid icing conditions.
- I_2.2.1 "The operator should be able to display flight corridors, controlled airspace and any other relevant airspace co-ordination information". (NATO, 2004)
- I_2.2.2 The GCS should display weather information to the pilot.
- I_2.2.3 The GCS should provide the pilot with information on the location of icing conditions, especially if the UA is not certificated for flight in icing conditions.
- I_2.2.4 The GCS should alert the pilot when the UA enters icing conditions.
- I_2.2.5 The GCS should alert the pilot when the UA encounters significant air turbulence.

SELF-SEPARATE FROM OTHER AIRCRAFT

- T_2.3.1 The GCS should enable the pilot to monitor all traffic in the airspace around the UA to identify potential for upcoming well-clear violations.
- T_2.3.2 The GCS should enable the pilot to quickly identify any threat of an aircraft violating the wellclear airspace of the UA
- T_2.3.3 The GCS should enable the pilot to track the surrounding traffic flight paths, assess the risk of well-clear violations, and recognize the need for the UA to maneuver to maintain self-separation criteria
- T_2.3.4 The GCS should enable the pilot to determine the appropriate maneuver for the UA to make to maintain self-separation.
- T_2.3.5 The GCS should enable the pilot to command the UA to execute the evasive maneuver
- T_2.3.6 The GCS should enable the pilot to communicate with ATC about making the separation maneuver and departing from the assigned flight path
- T_2.3.7 The GCS should enable the pilot to return the UA to the assigned flight path after maneuvering for self-separation.

APPROPRIATE LOST LINK PROCEDURE

- T_2.4.1 The GCS should enable the pilot to remain aware of the aircraft's lost link procedure as the flight progresses.
- T_2.4.2 The GCS should enable the pilot to update the aircraft's lost link procedure as the flight progresses.
- I_2.4.1 The GCS should provide the pilot with a display indicating the future flightpath of the aircraft should a lost link occur.
- I_2.4.2 The GCS should alert the pilot whenever the execution of a lost link procedure would create a hazard (such as directing the aircraft towards terrain, or into non-authorized airspace).
- P_2.4.1 The flightpath that would be taken by the aircraft in the event of a lost link should be clearly distinguishable from the programmed normal flightpath of the aircraft.
- P_2.4.2 Information on the programmed lost link behavior of the aircraft should be readily available to the pilot, without the need for complex interactions with the human-machine interface.

GENERAL GUIDELINES

- G_1 UAS developers should follow recognized human-centered design processes including the following:
- G_1a. Develop a full set of pilot tasks and intended operations for which the GCS will be used. These will help drive ensuring a thorough design that provides all systems, information, and controls that the pilots will need.
- G_1b. Develop an understanding of the potential safety critical errors that the pilots may make when accomplishing their tasks. These will provide the foundation for making tradeoffs in design decisions by focusing on design attributes that will mitigate critical errors as needed.
- G_1c. Develop a full set of information requirements for the tasks the pilots will need to accomplish. These requirements should be developed with other design requirements at the beginning of the systems engineering process. They will help ensure that the appropriate information is provided to the pilots and provide the foundation for making design decisions.
- G_1d. Develop a full set of requirements for controls that the pilot will need to accomplish their tasks. These requirements should be developed with other design requirements at the beginning of the systems engineering process. They will help ensure that all the pilot controls are planned for as design decisions are made.
- G_1e. Document all of the results of these processes so that they can be continually updated when design decisions and trade-offs are made during the design process. Good documentation will also help the human factors design processes to be integrated with the other systems engineering development and design processes.
- G_3 If changing a mode selection of an automated system has a safety consequence, the action to select that mode should be alerted, and additional precautions should be taken to prevent inadvertent selection
- G_4 Payload controls should be separate from controls with safety-of-flight functions.
- G_5 It should not be possible to reconfigure a safety-of-flight control to perform a payload function.
- G_6 Activation of a key or button should provide tactile or auditory feedback to the pilot. (ANSI/HFES, 2007)
- G_7 There should be a clear indication to the pilot when a command has been received by the UAS.
- G_8 Any unrecognized entry made by the pilot at the GCS should cause an informative error message to be displayed and not affect the status or operation of any system. (Access 5, 2006)
- G_10 Systems that alert the pilot to a critical anomaly should not be subject to a silent failure.
- G_11 The GCS should provide a work environment that maintains pilot engagement, and minimizes the negative impact of extended periods of low workload.

- G_12 The GCS should provide consistency of operation for common functions.
- G_13 The functions needed to safely control the aircraft under usual flight situations should be located in the pilot's primary field-of-view.
- G_14 Warnings and cautions should not be obscured by other GCS displays.
- G_15 "Part-time display. If it is desired to inhibit some parameters from full-time display, an equivalent level of safety to full-time display should be demonstrated. Criteria to be considered include the following:
 - Continuous display of the parameter is not required for safety of flight in all normal flight phases.
 - The parameter is automatically displayed in flight phases where it is required.
 - The inhibited parameter is automatically displayed when its value indicates an abnormal condition, or when the parameter reaches an abnormal value.
 - Display of the inhibited parameter can be manually selected by the UAV crew without interfering with the display of other required information.
 - If the parameter fails to be displayed when required, the failure effect and compounding effects must meet the requirements of USAR.1309. The analysis is to clearly demonstrate that the display(s) of data is consistent with safe operation under all probable operating conditions.
 - The automatic, or requested, display of the inhibited parameter should not create unacceptable clutter on the display; simultaneous multiple "pop-ups" must be considered.
 - If the presence of the new parameter is not sufficiently self-evident, suitable alerting must accompany the automatic presentation". STANAG 4671 AMC.1722
- G_16 Wherever possible, text messages, whether in dialog boxes, warning messages or other screen displays, should be presented in plain language, or using standard aviation terminology.
- G 17 Controls intended to be operated by the pilot should be reachable from a seated position
- G_18 The GCS should provide a bookrest to enable the pilot to refer to documents without risk that the document will come into contact with a keyboard or other flight controls.
- G_19 Appropriate priority controls should be available for UAS functions that require either quick accessibility or constant availability. Priority control devices can include, but are not limited to: (a) Touch panels, (b) Buttons, (c) Switches, (d) Joysticks, (e) Keyboard shortcuts. (NATO, 2004)
- G_20 If a display screen enables the pilot to move or rearrange display or control windows, it should not be possible to place a window so as to obscure primary flight controls or displays.

• Table A.1 — Checklist for assessing applicability and conformity with ISO 9241-210:2010	
---	--

Clause/	Requirement or recommendation	Applicability		Conformance		
subclause		Yes/No	Reason not applicable	Yes	No	Comments
4	Principles of human-centred design	<u>.</u>				
4.1	Whatever the design process and allocation of responsibilities and roles adopted, a human-centred approach should follow the principles listed [in 4.1].					
4.2	Products, systems and services should be designed to take account of the people who will use them as well as other stakeholder groups including those who might be affected (directly or indirectly) by their use.					
4.2	All relevant user and stakeholder groups should be identified. [see also 6.2.2 a)]					
4.3	User involvement should be active.					
4.3	The users who are involved should have capabilities, characteristics and experience that reflect the range of users for whom the system is being designed. [see also 6.2.2 b)]					
4.4	User-centred evaluation should take place as part of the final acceptance of the product to confirm that requirements have been met.					
4.5	Iteration should be used to progressively eliminate uncertainty during the development of interactive systems.					
4.6	The user's experience of previous or other systems and issues such as branding and advertising should also be considered.					
4.6	Users' strengths, limitations, preferences and expectations should be taken into account when specifying which activities are carried out by the users and which functions are carried out by the technology.					
4.6	Representative users should generally be involved in decisions related to the allocation of function.					
4.6	The human activities resulting from the allocation of function should form a set of tasks that is meaningful as a whole to the user.					
4.7	Human-centred design teams do not have to be large but the team should be sufficiently diverse to collaborate over design and implementation trade-off decisions at appropriate times.					

	•	Table A.1 ((continued)
--	---	-------------	-------------

Clause/	Requirement or recommendation		Conformance			
subclause		Yes/No	Reason not applicable	Yes	No	Comments
5	Planning human-centred design					
5.1	Human-centred design shall be planned and integrated into all phases of the product life cycle.	Y				
5.2	Those responsible for planning the project shall consider the relative imp	ortance of	human factors/ergonomics	s in the p	oroject b	by evaluating:
5.2 a)	how usability relates to the purpose and use of the product, system or service	Y				
5.2 b)	the levels of the various types of risk that might result from poor usability	Y				
5.2 c)	the nature of the development environment	Y				
5.3	The planning of human-centred design shall include:					
5.3 a)	identifying appropriate methods and resources for the activities described in Clause 6	Y				
5.3 b)	defining procedures for integrating these activities and their outputs with other system development activities	Y				
5.3 c)	identifying the individuals and the organization(s) responsible for the human-centred design activities and the range of skills and viewpoints they provide	Y				
5.3 d)	developing effective procedures for establishing feedback and communication on human-centred design activities as they affect other design activities and "trade-offs", and methods for documenting outputs from these activities	Y				
5.3 e)	agreeing on appropriate milestones for human-centred activities that are integrated into the overall design and development process	Y				
5.3 f)	agreeing on suitable timescales to allow iteration, use of feedback and possible design changes to be incorporated into the project schedule	Y				

•

•

5.4	The plan for human-centred design shall form part of the overall system development project plan.	Y		
5.4	To ensure that it is followed through and implemented effectively, the plan for human-centred design should be subject to the same project disciplines (e.g. responsibilities, change control) as other key activities.			

٠	Table A.1	(continued)
---	-----------	-------------

Clause/	Requirement or recommendation	Applicability		Conformance		
subclause		Yes/No	Reason not applicable	Yes	No	Comments
5.4	The human-centred design aspects of the project plan should be reviewed and revised appropriately as requirements change throughout the life of the project.					
5.5	Project planning shall allocate time and resources to the human- centred activities.	Y				
5.5	[The plan] shall include time for iteration and the incorporation of user feedback, and for evaluating whether the design solution satisfies the user requirements.	Y				
5.5	Additional time should be allocated to communication among design team participants and to reconciling potential conflicts and trade-offs that involve human–system issues.					
5.5	Human-centred design activities should start at the earliest stage of the project.					
5.5	The human-centred design aspects of the project plan should be reviewed throughout the life of the project.					
6	Human-centred design activities					
6.1	There are four linked human-centred design activities that shall take place	e during th	ne design of any interactive	system:	:	
6.1 a)	Understanding and specifying the context of use	Y				
6.1 b)	Specifying the user requirements	Y				
6.1 c)	Producing design solutions	Y				
6.1 d)	Evaluating the design	Y				
6.2.2	The context of use description shall include the following:	Y				
6.2.2 a)	Relevant groups shall be identified and their relationship with the proposed development described in terms of key goals and constraints.	Y				
6.2.2 b)	Relevant characteristics of the users shall be identified.	Y				
6.2.2 b)	If necessary, the characteristics of different types of users should be defined.					

6.2.2 b)	In order to achieve accessibility, products, systems and services should be designed to be used by people with the widest range of capabilities			
	in intended user populations.			

٠	Table A.1	(continued)
---	-----------	-------------

Clause/	Permirement or recommendation	Applicability		Conformance		
subclause	Requirement of recommendation	Yes/No	Reason not applicable	Yes	No	Comments
6.2.2.c)	The goals of the users and the overall goals of the system shall be identified.	Y				
6.2.2 c)	The characteristics of tasks that can influence usability and accessibility shall be described.	Y				
6.2.2 c)	Any potential adverse consequences for health and safety should be identified.					
6.2.2 c)	If there is a risk that the task might be completed incorrectly, this should be identified.					
6.2.2 c)	Tasks should not be described solely in terms of the functions or features provided by a product or system.					
6.2.2 d)	The technical environment, including the hardware, software and materials, shall be identified.	Y				
6.2.2 d)	The relevant characteristics of the physical, social , organizational and cultural environment shall be described.	Y				
6.2.3	The context of use of the system should be described in sufficient detail to support the requirements, design and evaluation activities.	Y				
6.2.4	The intended context of use should be specified as part of the user requirements specification to clearly identify the conditions under which the requirements apply.					
6.3.1	Identifying user needs and specifying the functional and other requirements for the product or system shall be extended to create an explicit statement of user requirements in relation to the intended context of use and the business objectives of the system.					
6.3.1	If it is known that the proposed interactive system will affect organizational practice, the development process should involve organizational stakeholders in the design process with the aim of optimizing both the organizational and technical systems.					
6.3.2	User and other stakeholder needs should be identified, taking account of the context of use.					

6.3.2	User and other stakeholder needs should include that which users need to achieve (rather than how to achieve it) and any constraints imposed by the context of use.			
•				

Clause/	De suissment es se emmendation		Conformance			
subclause	Requirement or recommendation	Yes/No	Reason not applicable	Yes	No	Comments
6.3.3	The specification of user requirements shall include:					
6.3.3 a)	the intended context of use	Y				
6.3.3 b)	requirements derived from user needs and the context of use	Y				
6.3.3 c)	requirements arising from relevant ergonomics and user interface knowledge, standards and guidelines	Y				
6.3.3 d)	usability requirements and objectives including measurable usability performance and satisfaction criteria in specific contexts of use	Y				
6.3.3 e)	requirements derived from organizational requirements that directly affect the user	Y				
6.3.4	Potential conflicts between user requirements should be resolved.					
6.3.4	The rationales, the factors and the weighting of human–system issues for use in any trade-offs should be documented so that they can be understood in the future.					
6.3.5	The user requirements specification should be:					
6.3.5 a)	stated in terms that permit subsequent testing					
6.3.5 b)	verified by the relevant stakeholders					
6.3.5 c)	internally consistent					
6.3.5 d)	updated as necessary during the life of the project					
6.4.1	Producing design solutions should include the following sub-activities:			-		
6.4.1 a)	designing user tasks, user-system interaction and user interface to meet the user requirements, taking into consideration the overall user experience					

6.4.1 b)	making the design solutions more concrete			
6.4.1 c)	altering the design solutions in response to user-centred evaluation and feedback			
6.4.1 d)	communicating the design solutions to those responsible for their implementation			

Clause/			Applicability	Conformance					
subclause	Requirement or recommendation	Yes/No	Reason not applicable	Yes	No	Comments			
6.4.2.1	The following principles (taken from ISO 9241-110) should be taken into a	account wh	nen designing interactive sy	/stems:					
6.4.2.1 a)	suitability for the task								
6.4.2.1 b)	self-descriptiveness								
6.4.2.1 c)	conformity with user expectations								
6.4.2.1 d)	suitability for learning								
6.4.2.1 e)	controllability								
6.4.2.1 f)	error tolerance								
6.4.2.1 g)	suitability for individualization								
6.4.2.2	Designing the interaction should include:								
6.4.2.2 a)	making high-level decisions								
6.4.2.2 b)	identifying tasks and sub-tasks								
6.4.2.2 c)	allocating tasks and sub-tasks to user and other parts of system								
6.4.2.2 d)	identifying the interaction objects required for the completion of the tasks								
6.4.2.2 e)	identifying appropriate dialogue techniques								
6.4.2.2 f)	designing the sequence and timing (dynamics) of the interaction								
6.4.2.2 g)	designing the information architecture of the user interface of an interactive system to allow efficient access to interaction objects								
6.4.2.3	Ergonomics and user interface knowledge, standards and guidelines should be used to inform the design of both hardware and software of the user interface.								
6.4.3	The level of detail and realism [of prototypes] should be appropriate to the issues that need to be investigated.								
6.4.4	Feedback from evaluation should be used to improve and refine the system.								

6.4.4 The costs and benefits of proposed changes should be evaluated and used to inform decisions about what will be modified.			
--	--	--	--

Clause/	Pequirement or recommendation		Applicability	Conformance						
subclause	Requirement of recommendation	Yes/No	Reason not applicable	Yes	No	Comments				
6.4.4	Project plans should allow sufficient time for making the changes as a result of such feedback.									
6.4.5	There should be some sustained channel of communication between those responsible for human-centred design and other members of the project team.									
6.4.5	When design solutions are communicated, they should be accompanied by an explanation and justification of the design decisions, especially where trade-offs are necessary									
6.4.5	The communication [of details of the design] should take account of the constraints imposed by the project and the project team's knowledge and understanding about ergonomics and user interface design.									
6.5.1	User-centred evaluation (evaluation based on the user's perspective) is a required activity in human-centred design.	Y								
6.5.1	Even at the earliest stages in the project, design concepts should be evaluated to obtain a better understanding of user needs.									
6.5.1	If user-based testing is not practical or cost-effective at a particular stage of a project, design solutions should be evaluated in other ways.									
6.5.2	User-centred evaluation should involve:									
6.5.2 a)	allocating resources both for obtaining early feedback to improve the product, and later for determining if requirements have been satisfied									
6.5.2 b)	planning the user-centred evaluation so that it fits the project schedule									
6.5.2 c)	carrying out sufficiently comprehensive testing to give meaningful results for the system as a whole									
6.5.2 d)	analysing the results, prioritizing issues and proposing solutions									

6.5.2 e)	communicating the solutions appropriately so that they can be used effectively by the design team			
6.5.3	To obtain valid results, the evaluation should be carried out by experienced evaluators.			
6.5.3	To obtain valid results, the evaluation should use appropriate methods.			

Clause/	De muinement en merennen detter		Applicability	Conformance						
subclause	Requirement or recommendation	Yes/No	Reason not applicable	Yes	No	Comments				
6.5.3	Resources for evaluation should be allocated both to obtain early feedback with which to improve the product, and, at a later stage, to validate whether the user requirements have been satisfied.									
6.5.3	The extent of the latter (summative) evaluation should depend on the extent of the risks associated with not meeting requirements.									
6.5.4	When prototypes are being tested, users should carry out tasks using the prototype rather than just be shown demonstrations or a preview of the design.									
6.5.6	A human-centred design process should include long-term monitoring of the use of the product, system or service.									
6.5.6	Criteria and measurements [for long-term monitoring] should be sensitive enough to identify system failure, or system problems, as early as possible.									

APPENDIX B - COMPLIANCE RATING RESULTS

				•		Table B.1	<u> </u>	nec	klist for asses	sing	app	licability and	cor	ntorm	nity with ISO	924	1-21	0:2010					
								PI	IYSICAL DE	MAN	D C	OMPLIANC	CE R	ATI	NG								
					(T	= Rating Score	re for	Tan	dem Rotor UA	V GC	CS, R	=Rating Scor	e foi	r Rye	erson Reconfi	gurał	ole G	CS)					
Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R
2.4.3	1	1	5.4.6.3	1	1	T_1.4.3	1	0	I_1.4.4	1	0	I_1.4.11	1	0	P_1.4.2	1	0	G_3	1	0	G_13	1	1
2.6.4	1	1	5.4.6.6	1	1	T_1.4.4	1	0	I_1.4.5	1	0	I_1.4.12	1	0	P_1.4.3	1	0	G_4	1	0	G_14	1	0
2.6.10	1	1	9.4.2.3	1	1	T_1.4.5	1	0	I_1.4.6	1	0	I_1.4.13	1	0	P_1.4.4	1	0	G_5	1	0	G_15	1	1
2.6.13	1	1	9.4.2.4	1	1	T 1.4.6	1	0	I 1.4.7	1	0	C 1.4.1	1	0	G 1	1	0	G 6	1	1	G 16	1	1
5.1.2.4	1	1	9.4.2.5	1	1	T 1.4.7	1	0	I 1.4.8	1	0	C 1.4.2	1	1	G la	1	0	G 7	1	0	G 17	1	1
5.1.2.8	1	0	9.4.2.6	1	1	T_1.4.8	1	0	I_1.4.9	1	0	C_1.4.2a	1	1	G_1b	1	0	G_8	1	0	G_18	0	0
5.1.2.9	1	0	9.4.2.7	1	1	I_1.4.1	1	0	I_1.4.9a	1	0	C_1.4.2b	1	1	G_1c	1	0	G_10	1	1	G_19	1	1
5.1.2.10	1	0	T_1.4.1	1	0	I_1.4.2	1	0	I_1.4.9b	1	0	C_1.4.2c	1	1	G_1d.	1	0	G_11	1	0	G_20	1	0
5.1.2.10b	1	0	T_1.4.2	1	0	I_1.4.3	1	0	I_1.4.10	1	0	P_1.4.1	1	0	G_1e	1	0	G_12	1	1			
								Ν	IENTAL DEN	[AN]	D CO	OMPLIANC	E R	ATIN	NG								
					(T	= Rating Scor	re for	Tan	dem Rotor UA	V GC	CS, R	=Rating Scor	e foi	r Rye	erson Reconfi	gurał	ole G	CS)					
Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R
2.2.1	1	1	3.1.12	1	0	3.7.8	1	0	8.13.5.3	1	1	8.1.1.7	1	1	8.1.3.7	1	0	G_3	1	0	G_15	1	1
2.3.1	1	1	3.1.16	1	0	3.8.2	1	0	8.13.5.4	1	1	8.1.1.8	1	0	8.1.3.9	1	0	G_4	1	0	G_16	1	1
2.4.2	1	1	3.1.17	1	1	3.8.8	1	0	8.13.5.5	1	1	8.1.2.3	1	1	8.1.3.10	1	1	G_5	1	0	G_17	1	1
2.4.3	1	1	3.1.23	1	0	6.2.1.1.3	1	1	8.13.5.6	1	1	8.1.2.5	1	0	8.1.3.13	1	0	G_6	1	1	G_18	0	0
2.4.5	1	1	3.4.1	1	1	6.2.1.1.4	1	0	8.13.5.7	1	1	8.1.2.6	1	0	8.1.3.17	1	0	G_7	1	0	G_19	1	1
2.6.3	1	1	3.4.2	1	1	6.2.1.1.5	1	0	8.14.3.6.1	1	0	8.1.2.8	1	1	G_1	1	0	G_8	1	0	G_20	1	0
2.6.4	1	1	3.4.5	1	1	6.2.2.1.2	1	1	8.13.5.8	1	0	8.1.2.9	1	0	G_1a	1	0	G_10	1	1			
2.6.10	1	1	3.5.1	1	0	6.2.2.1.3	1	0	8.1.1.1	1	0	8.1.3.1	0	0	G_1b	1	0	G_11	1	0			
2.6.13	1	1	3.5.4	1	0	6.3.1.2	1	0	8.1.1.2	1	0	8.1.3.3	1	0	G_1c	1	0	G_12	1	1			
3.1.1	1	0	3.6.1	1	0	8.13.5.1	1	1	8.1.1.5	1	1	8.1.3.4	1	0	G_1d.	1	0	G_13	1	1			
3.1.10	1	0	3.7.6	1	0	8.13.5.2	1	1	8.1.1.6	1	1	8.1.3.6	1	0	G_1e	1	0	G_14	1	0			
							Ol	PER	ATOR PERFO	ORM	ANC	CE COMPLI	AN	CE R	RATING								
					(T	= Rating Score	re for	Tan	dem Rotor UA	V GC	CS, R	=Rating Scor	e foi	r Rye	erson Reconfi	gurał	ole G	CS)				-	
Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R
2.2.1	1	1	6.2.2.1.2	1	1	8.1.3.3	1	0	T_1.4.7	1	0	T_1.5.2	1	0	I_1.5.14	1	0	P_2.1.6	1	0	G_1	1	0
2.3.1	1	1	6.2.2.1.3	1	0	8.1.3.4	1	0	T_1.4.8	1	0	T_1.5.3	1	0	I_1.5.15	1	0	P_2.1.7	1	0	G_1a	1	0
2.4.2	1	1	6.3.1.2	1	0	8.1.3.6	1	0	I_1.4.1	1	0	T_1.5.4	1	0	C_1.5.1	1	0	P_2.1.8	1	0	G_1b	1	0
2.4.5	1	1	7.1.1.1	1	0	8.1.3.7	1	0	I_1.4.2	1	0	T_1.5.5	1	0	C_1.5.2	1	0	P_2.1.9	1	0	G_1c	1	0
2.6.3	1	1	7.1.1.3	1	1	8.1.3.9	1	0	I_1.4.3	1	0	T_1.5.6	1	0	C_1.5.3	1	0	T_2.2.1	1	0	G_1d.	1	0
2.6.4	1	1	7.1.2.8	1	0	8.1.3.10	1	1	I_1.4.4	1	0	T_1.5.7	1	0	C_1.5.4	1	0	T_2.2.2	1	0	G_1e	1	0

11 1

2.6.10	1	1	8.13.5.1	1	1	8.1.3.13	1	0	I 1.4.5	1	0	T 1.5.8	1	0	P 1.5.1	1	0	Т 2.2.3	1	0	G 3	1	0
2.6.13	1	1	8.13.5.2	1	1	8.1.3.17	1	0	I 1.4.6	1	0	T 1.5.9	1	0	P 1.5.2	1	0	T 2.2.4	1	0	G 4	1	0
3.1.1	1	0	8.13.5.3	1	1	L1.1	1	0	I_1.4.7	1	0	T_1.5.10	1	0	P_1.5.3	1	0	T_2.2.5	1	0	G_5	1	0
3.1.10	1	0	8.13.5.4	1	1	T_1.1.1	1	0	I_1.4.8	1	0	T_1.5.11	1	0	P_1.5.4	1	0	I_2.2.1	1	0	G_6	1	1
3.1.12	1	0	8.13.5.5	1	1	T_1.1.2	1	0	I_1.4.9	1	0	T_1.5.12	1	0	P_1.5.5	1	0	I_2.2.2	1	0	G_7	1	0
3.1.16	1	0	8.13.5.6	1	1	P_1.1.1	1	0	I_1.4.9a	1	0	T_1.5.13	1	0	P_1.5.6	1	0	I_2.2.3	1	0	G_8	1	0
3.1.17	1	1	8.13.5.7	1	1	P_1.1.2	1	0	I_1.4.9b	1	0	T_1.5.14	1	0	P_1.5.7	1	0	I_2.2.4	1	0	G_10	1	1
3.1.23	1	0	8.14.3.6.1	1	0	P_1.1.3	1	0	I_1.4.10	1	0	T_1.5.15	1	0	P_1.5.8	1	0	I_2.2.5	1	0	G_11	1	0
3.4.1	1	1	8.13.5.8	1	0	T_1.2.1	1	1	I_1.4.11	1	0	I_1.5.1	1	0	P_1.5.9	1	0	T_2.3.1	1	0	G_12	1	1
3.4.2	1	1	8.1.1.1	1	0	I_1.2.1	1	1	I_1.4.12	1	0	I_1.5.2	1	0	P_1.5.10	1	0	T_2.3.2	1	0	G_13	1	1
3.4.5	1	1	8.1.1.2	1	0	T_1.3.1	1	1	I_1.4.13	1	0	I_1.5.3	1	0	T_2.1.1	1	0	T_2.3.3	1	0	G_14	1	0
3.5.1	1	0	8.1.1.5	1	1	T_1.3.2	1	1	C_1.4.1	1	0	I_1.5.4	1	0	T_2.1.2	1	0	T_2.3.4	1	0	G_15	1	1
3.5.4	1	0	8.1.1.6	1	1	T_1.3.3	1	1	C_1.4.2	1	1	I_1.5.5	1	0	T_2.1.3	1	0	T_2.3.5	1	1	G_16	1	1
3.6.1	1	0	8.1.1.7	1	1	T_1.3.4	0	0	C_1.4.2a	1	1	I_1.5.6	1	0	I_2.1.1	1	1	T_2.3.6	0	0	G_17	1	1
3.7.6	1	0	8.1.1.8	1	0	I_1.3.1	1	1	C_1.4.2b	1	1	I_1.5.7	1	0	I_2.1.2	1	0	T_2.3.7	1	0	G_18	0	0
3.7.8	1	0	8.1.2.3	1	1	T_1.4.1	1	0	C_1.4.2c	1	1	I_1.5.8	1	0	C_2.1.3	1	1	T_2.4.1	1	0	G_19	1	1
3.8.2	1	0	8.1.2.5	1	0	T_1.4.2	1	0	P_1.4.1	1	0	I_1.5.9	1	0	P_2.1.1	1	0	T_2.4.2	1	0	G_20	1	0
3.8.8	1	0	8.1.2.6	1	0	T_1.4.3	1	0	P_1.4.2	1	0	I_1.5.10	1	0	P_2.1.2	1	0	I_2.4.1	1	0			
6.2.1.1.3	1	1	8.1.2.8	1	1	T_1.4.4	1	0	P_1.4.3	1	0	I_1.5.11	1	0	P_2.1.3	1	0	I_2.4.2	1	0			
6.2.1.1.4	1	0	8.1.2.9	1	0	T_1.4.5	1	0	P_1.4.4	1	0	I_1.5.12	1	0	P_2.1.4	1	0	P_2.4.1	1	0			
6.2.1.1.5	1	0	8.1.3.1	0	0	T_1.4.6	1	0	T_1.5.1	1	0	I_1.5.13	1	0	P_2.1.5	1	0	P_2.4.2	1	0			
									EFFORT	CO	MPL	IANCE RA	ΓΙΝ	G									
					(T	= Rating Scor	e for	Tan	dem Rotor UA	V GO	CS, R	=Rating Scor	re fo	r Rye	erson Reconfi	gural	ble G	CS)					
Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R
2.4.3	1	1	5.1.2.8	1	0	5.4.6.6	1	1	9.4.2.7	1	1	G_1d.	1	0	G_6	1	1	G_12	1	1	G_17	1	1
2.6.4	1	1	5.1.2.9	1	0	9.4.2.3	1	1	G_1	1	0	G_1e	1	0	G_7	1	0	G_13	1	1	G_18	0	0
2.6.10	1	1	5.1.2.10	1	0	9.4.2.4	1	1	G_1a	1	0	G_3	1	0	G_8	1	0	G_14	1	0	G_19	1	1
2.6.13	1	1	5.1.2.10b	1	0	9.4.2.5	1	1	G_1b	1	0	G_4	1	0	G_10	1	1	G_15	1	1	G_20	1	0
5.1.2.4	1	1	5.4.6.3	1	1	9.4.2.6	1	1	G_1c	1	0	G_5	1	0	G_11	1	0	G_16	1	1			
									FRUSTRAT	ION	CON	IPLIANCE	RA	ΓING	Ĵ								
			1		(T	= Rating Scor	e for	[.] Tan	dem Rotor UA	V GO	CS, R	=Rating Scor	re fo	r Rye	erson Reconfi	gural	ble G	CS)					
Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R
2.2.1	1	1	3.1.10	1	0	3.7.6	1	0	8.13.5.2	1	1	8.1.1.6	1	1	8.1.3.6	1	0	G_1e	1	0	G_14	1	0
2.2.3	1	1	3.1.12	1	0	3.7.8	1	0	8.13.5.3	1	1	8.1.1.7	1	1	8.1.3.7	1	0	G_3	1	0	G_15	1	1
2.3.1	1	1	3.1.16	1	0	3.8.2	1	0	8.13.5.4	1	1	8.1.1.8	1	0	8.1.3.9	1	0	G_4	1	0	G_16	1	1
2.4.2	1	1	3.1.17	1	1	3.8.8	1	0	8.13.5.5	1	1	8.1.2.3	1	1	8.1.3.10	1	1	G_5	1	0	G_17	1	1
2.4.3	1	1	3.1.23	1	0	6.2.1.1.3	1	1	8.13.5.6	1	1	8.1.2.5	1	0	8.1.3.13	1	0	G_6	1	1	G_18	0	0
2.4.5	1	1	3.4.1	1	1	6.2.1.1.4	1	0	8.13.5.7	1	1	8.1.2.6	1	0	8.1.3.17	1	0	G_7	1	0	G_19	1	1
263	1	1	342	1	1	62115	1	0	8 14 3 6 1	1	0	8128	1	1	G 1	1	0	G 8	1	0	G 20	11	0

2.6.4	1	1	3.4.5	1	1	6.2.2.1.2	1	1	8.13.5.8	1	0	8.1.2.9	1	0	G_la	1	0	G_10	1	1			
2.6.10	1	1	3.5.1	1	0	6.2.2.1.3	1	0	8.1.1.1	1	0	8.1.3.1	0	0	G_1b	1	0	G_11	1	0			
2.6.13	1	1	3.5.4	1	0	6.3.1.2	1	0	8.1.1.2	1	0	8.1.3.3	1	0	G_1c	1	0	G_12	1	1			
3.1.1	1	0	3.6.1	1	0	8.13.5.1	1	1	8.1.1.5	1	1	8.1.3.4	1	0	G_1d.	1	0	G_13	1	1			
									ISO 9241-210:	:2010) CO	MPLIANCE	RA	TIN	G								
					(T	= Rating Sco	re for	Tan	dem Rotor UA	V GC	CS, R	Rating Score	re foi	r Rye	erson Reconfi	gural	ole G	CS)					
Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R	Clause	Т	R
4.1	1	0	5.2 a)	1	0	5.5	1	1	6.2.2.c)	1	0	6.3.3 a)	1	0	6.4.1 c)	1	0	6.4.2.2 e)	1	0	6.5.2 a)	1	1
4.2	1	0	5.2 b)	1	0	5.5	1	0	6.2.2 c)	1	0	6.3.3 b)	1	0	6.4.1 d)	1	0	6.4.2.2 f)	1	0	6.5.2 b)	1	1
4.2	1	0	5.2 c)	1	0	5.5	1	0	6.2.2 c)	1	0	6.3.3 c)	1	0	6.4.2.1 a)	1	0	6.4.2.2 g)	1	0	6.5.2 c)	1	1
4.3	1	0	5.3 a)	1	0	5.5	1	0	6.2.2 c)	1	0	6.3.3 d)	1	0	6.4.2.1 b)	1	0	6.4.2.3	1	0	6.5.2 d)	1	1
4.3	1	0	5.3 b)	1	0	6.1 a)	1	1	6.2.2 c)	1	0	6.3.3 e)	1	0	6.4.2.1 c)	1	0	6.4.3	1	0	6.5.2 e)	1	1
4.4	1	0	5.3 c)	1	0	6.1 b)	1	1	6.2.2 d)	1	0	6.3.4	1	0	6.4.2.1 d)	1	0	6.4.4	1	0	6.5.3	1	1
4.5	1	0	5.3 d)	1	0	6.1 c)	1	1	6.2.2 d)	1	0	6.3.4	1	0	6.4.2.1 e)	1	0	6.4.4	1	0	6.5.3	1	1
4.6	1	0	5.3 e)	1	0	6.1 d)	1	1	6.2.3	1	0	6.3.5 a)	1	0	6.4.2.1 f)	1	0	6.4.5	1	0	6.5.3	1	1
4.6	1	0	5.3 f)	1	0	6.2.2	1	1	6.2.4	1	0	6.3.5 b)	1	0	6.4.2.1 g)	1	0	6.4.5	1	0	6.5.3	1	1
4.6	1	0	5.4	1	0	6.2.2 a)	1	0	6.3.1	1	0	6.3.5 c)	1	0	6.4.2.2 a)	1	0	6.4.5	1	0	6.5.4	1	1
4.6	1	0	5.4	1	0	6.2.2 b)	1	0	6.3.1	1	0	6.3.5 d)	1	0	6.4.2.2 b)	1	0	6.5.1	1	1	6.5.6	1	1
4.7	1	0	5.4	1	0	6.2.2 b)	1	0	6.3.2	1	0	6.4.1 a)	1	0	6.4.2.2 c)	1	0	6.5.1	1	0	6.5.6	1	1
5.1	1	0	5.5	1	1	6.2.2 b)	1	0	6.3.2	1	0	6.4.1 b)	1	0	6.4.2.2 d)	1	0	6.5.1	1	0			

APPENDIX C - PARTICIPANTS CONSENT FORM



You are being invited to participate in a research study. Please read this consent form so that you understand what your participation will involve. Before you consent to participate, please ask any questions to be sure you understand what your participation will involve.

Evaluating UAV Ground Control Station Design Using Available Human Factors Guidelines and Standards

INVESTIGATORS

This research study is being conducted by Taiwo Amida, a master of applied science student under the supervision of Dr. Chung, from the Department of Aerospace Engineering at Ryerson University.

If you would like more details about something mentioned here, or information not included here, please feel free to contact either Taiwo Amida (tamida@ryerson.ca) or Dr. Joon Chung (j3chung@ryerson.ca, (416) 979-5000 Ext. 7213)

PURPOSE OF THE STUDY

This research study will investigate, and evaluate the potential benefits of implementing all the available human factors standards to the design of an Unmanned Aerial Vehicle Ground Control Station. This study specifically seeks to examine how the implementation of these standards will affect the operator's workload. The Human Factors Compliance Rating will be used to evaluate effectiveness of each standards implemented on the ground control station design while the user's workload will be evaluated by using NASA's Task Load Index.

To be eligible for participation, you must be able to have functional control over your hands, so that you can operate a wide variety of control inputs (e.g., Mouse, Keyboard, Throttle, Joystick, Touch Screens) in a rapid and accurate manner as you will have to be able to respond quickly to potential changes in flight dynamics (due to system failures, changes in velocity, etc.). In addition, you must be able to view a monitor for periods of time (up to 20 minutes at a time) and thus cannot be visually impaired.

The research results taken from this study will be submitted as part of a scholarly journal publication and might be presented within a conference proceeding.

What you Will Be Asked to Do

If you volunteer to participate in this study, you will be asked to complete the following tasks:

- 1. Complete four short simulated flight missions. These missions will require you to take off with the aircraft, fly through five or seven waypoints (depending on the test) as well as using auto pilot system and payload controller within a pre-defined area within the simulation environment.
- 2. Fill out Task Load Index worksheets created from NASA. These worksheets are used to evaluate a user's perceived workload. Where workload has been defined as the mental and physical costs that a human operator feels subjected to while trying to obtain a particular level of performance in a task. The workload is determined based off of the contributions of six human factors, Mental Demands, Physical Demands, Temporal (time based) Demands, Overall Performance, Effort, and Frustration.
- 3. The first worksheet will require you to complete 15 pair-wise comparisons, choosing in each pair which of two mentioned human factors was the higher contributor to their feeling of workload.

- 4. The second worksheet will require you to rate each of the six human factors on a 21-point rated scale that ranges from 0 to 100 (i.e., each scale tick relates to an increase in 5 points), where a 0 means the factor had little effect on yourself and a score of 100 means the factor affected you greatly.
- 5. Fill out a questionnaire after all of the flight tests have been completed which is comprised of four questions. The first question is in regards to your potential flight and simulator experience. The second questions asks which of the two systems you preferred (i.e., the mouse and keyboard or multi-touch interface). The third question asks what you happened to like about the systems they were interacting with. The fourth question asks what you would like to see improved/ disliked about the systems they were interacting with.

This session will be comprised of the a tutorials (roughly 10 to 15 minutes in length), four flight tests (that take on average between 10 to 20 minutes to complete) as well as Task Load Index and worksheet and the additional questionnaire (the worksheets take roughly 5 to 10 minutes to complete). You will be required to take breaks after every test. Thus, it will take about 2 hours to complete the entire testing session depending on your speed and experience with flight controls.

Potential Benefits

We cannot guarantee that you will receive any direct benefits from participating in this study. However, the data that is obtained from this study will be used to determine if close implementation of human factors standards affects the user's workload and performance

What are the Potential Risks to you as a Participant?

There is minimal risk associated with participation in this study. However, some potential risks and discomforts do exit. Due to the physical nature of using a mouse and keyboard/ multi-touch monitor, there is a possibility that you will experience a Repetitive Strain Injury through the prolonged use of the system. For this reason, you will be expected to take multiple breaks during the test session. In addition, the simulation could potentially cause you to feel anxious or frustrated due to the virtual simulation environment. It will be clearly explained to you that the aircraft you are controlling is completely virtual and crashing it does not void the research or put anyone in danger. Also, although the researchers are taking several steps to protect your confidentiality, it is possible that your identity could be discerned because some other researchers/people may be present in the same room while you complete the testing process. However, only members of the research team will be given access to the data that you provided for the testing procedure.

Confidentiality

Information collected for this research study will be kept secure and confidential, as required by law. All the worksheets that are required to be filled out will not have your name identified on them in order to keep your results confidential. The results provided from the completed worksheets will be a part of data presented within scholarly journal publications and potentially at conference proceedings. However, your name will not be provided in such cases. In addition, the data collected will only be available to the research team and will be destroyed after 1 year. In addition, it is important to note that your confidentiality will be strictly maintained and any reference to your identity will be destroyed.

Cost Of Participation

You will not be paid for participating in this research. Likewise, you will not be expected to pay for any research-related costs.

Voluntary Participation and Withdrawal

Participation in this study is completely voluntary. You can choose whether to be in this study or not. If any question makes you uncomfortable, you can skip that question. You may stop participating at any time. If you choose to stop participating, you may also request to not have your data included in the study. Your choice of whether to participate will not influence your future relations with Ryerson University or the investigators, Dr. Joon Chung and/or Taiwo Amida, involved in the research.

Questions about the Study

If you have any questions about the research now, please ask. If you have questions later about the research, you may contact:

Taiwo Amida tamida@ryerson.ca

Dr. Joon Chung j3chung@ryerson.ca (416) 979-5000 Ext. 7213

This study has been reviewed by the Ryerson University Research Ethics Board. If you have questions regarding your rights as a participant in this study, please contact:

Research Ethics Board c/o Office of the Vice President, Research, and Innovation Ryerson University 350 Victoria Street Toronto, ON M5B 2K3 416-979-5042 rebchair@ryerson.ca

CONFIRMATION OF AGREEMENT

Your signature below indicates that you have read the information in this agreement and have had a chance to ask any questions you have about the study. Your signature also indicates that you agree to participate in the study and have been told that you can change your mind and withdraw your consent to participate at any time. You have been given a copy of this agreement.

You have been told that by signing this consent agreement you are not giving up any of your legal rights.

Name of Participant (please print)

Signature of Participant

Date

APPENDIX D - IN-PERSON RECRUITMENT TRANSCRIPT



In-person Recruitment Transcript

Hello,

My name is Taiwo Amida. I am a Master of Applied Science student at Ryerson University in the Department of Aerospace Engineering, and I would like to invite you to consider taking part in a research study I am currently working on.

This research is being done as part of a research project, and my supervisor's name is Dr. Joon Chung who is an Associate Professor within the Department of Aerospace Engineering at Ryerson University. The focus of the research is to explore how the close implementation of human factors standards and guidelines in the ground control station design could affect an Unmanned Aerial Vehicle operator's feeling of workload within a simulated virtual environment.

In order to participate in this research study, you must be able to have functional control over your hands, so that you can operate a wide variety of control inputs (e.g., Mouse, Keyboard, Throttle, Joystick, Touch Screens) in a rapid and accurate manner in order to respond quickly to potential changes in flight dynamics (e.g., system failures, changes in velocity, etc.). In addition, you must be able to view a monitor for up to 20 minutes at a time, and thus it is required that you are not visually impaired.

If you agree to volunteer, you will be asked to participate in 4 short flight simulation tests using 2 different ground control stations (2 tests for each control station). Following these tests, you will be asked to fill out two worksheets, provided by NASA's Task Load Index, which will be used to determine your workload score. All of the collected data will be kept completely anonymous; however, the overall results might be presented within journal publications or at conferences.

There will be an initial tutorial session on how to use the 2 ground controls. After that, you will be required to carry out 2 tests on each ground control station (4 tests in total). These tests can last between 10 to 20 minutes each (depending on your speed and familiarity with flight controls) along with up to 10 minutes after each test for the completion of the Task Load Index worksheets. You will be also required to take mini breaks of about 5-10 minutes after every test. Thus, the entire testing session will last 2 hours.

Please note that Ryerson University is not directly involved in this research and your participation is not mandatory. In addition, refusing to participate in the research will not in any way affect your relationship with me, Dr. Joon Chung or Ryerson University.

If you are interested in knowing more about the study or would like to volunteer, please email me at (tamida@ryerson.ca).

Thank you for your time and help!

APPENDIX E - MAIL RECRUITMENT TRANSCRIPT



Hello,

My name is Taiwo Amida. I am a student at Ryerson University in the Department of Aerospace Engineering. I am contacting you to see if you would consider participating in a research study about Evaluating UAV ground control station design using available human factors standards.

This research is being done as part of a research project, and my supervisor's name is Dr. Joon Chung who is an Associate Professor within the Department of Aerospace Engineering at Ryerson University. The focus of the research is to explore how the close implementation of human factors standards and guidelines in the ground control station design could affect an Unmanned Aerial Vehicle operator's feeling of workload within a simulated virtual environment.

In order to participate in this research study, you must be able to have functional control over your hands, so that you can operate a wide variety of control inputs (e.g., Mouse, Keyboard, Throttle, Joystick, Touch Screens) in a rapid and accurate manner in order to respond quickly to potential changes in flight dynamics (e.g., system failures, changes in velocity, etc.). In addition, you must be able to view a monitor for up to 20 minutes at a time, and thus it is required that you are not visually impaired.

If you agree to volunteer, you will be asked to participate in 4 short flight simulation tests using 2 different ground control stations (2 tests for each control station). Following these tests, you will be asked to fill out two worksheets, provided by NASA's Task Load Index, which will be used to determine your workload score. All of the collected data will be kept completely anonymous; however, the overall results might be presented within journal publications or at conferences.

There will be an initial tutorial session on how to use the 2 ground controls. After that, you will be required to carry out 2 tests on each ground control station (4 tests in total). These tests can last between 10 to 20 minutes each (depending on your speed and familiarity with flight controls) along with up to 10 minutes after each test for the completion of the Task Load Index worksheets. You will be also required to take mini breaks of about 5-10 minutes after every test. Thus, the entire testing session will last 2 hours.

Please note that Ryerson University is not directly involved in this research and your participation is not mandatory. In addition, refusing to participate in the research will not in any way affect your relationship with me, my supervisor (Dr. Joon Chung) or Ryerson University.

If you are interested in knowing more about the study or would like to volunteer, please email me at (tamida@ryerson.ca).

Thank you for your time and help. And please, kindly forward this email to anyone you feel might also be interested in the study.

Regards,

Taiwo Amida



Temporal Demand Physical Demand Mental Demand Mental Demand Performance Frustration Frustration Effort ٩ ٥ſ OF or 16 Physical Demand Mental Demand Mental Demand Performance Effort Effort or 9 10 Subject ID: MENTAL DEMAND EFFORT PHYSICAL DEMAND Good FRUSTRATION PERFORMANCE TEMPORAL DEMAND Low Low Low Low Low _ RATING SHEET Appendix C. Task ID: 17 High High High Poor High High

APPENDIX G - ADDITIONAL QUESTIONNAIRE

1. Do you have any prior flight experience, simulator or flight game experience? If so, please indicate below.

- 2. Which of the two Ground Control Stations did you prefer to use? (Please mark one the options below to indicate your preference)
 - $\circ \quad \text{New reconfigurable Ground Control Station} \\$
 - Old reconfigurable Ground Control Station
- 3. Please indicate the features like in the Ground Control Stations?

4. Please indicate the features you dislike or would you like to see improved in the Ground Control Stations?

APPENDIX H - USER REQUIREMENTS AND GROUPING

HTA Task	Final Nodes	Requirements	Group
Execute Manual Take- off Execute Manual Landing	1.1 Check Current Throttle Position1.2 Move Throttle Position Back to zero	GCS must have a throttle position indicator	Primary
	3.1.1 Check If No Engine Error Message 3.2.1 Check If No Link Error Message 3.3.1 Check If No Obstacle Error Message	GCS must have error and status message panel.	Primary
	 4.1.1 Move Throttle Position to 100%, 4.1.2 Gradually Move Collective Position to 100%, 4.1.3.1 Check the rate of Change of the Vertical Speed, 4.1.3.2 Adjust Vertical Speed, 4.1.4.1 Check if Indicated Air Speed is zero, 4.1.5 Adjust Collective Back to Zero, 4.2.1 Check If 250FT is Attained, 4.2.2 Activate Hover. 	 Collective input response must be indicated, vertical speed must be shown to the pilot, rate of change of vertical speed must be shown to pilot, indicated airspeed and altitude of the aircraft must be shown, Pilot must be able to activate the hover mode 	Primary
Execute Automatic Take-off Execute automatic landing	 4.1.1 Activate Automatic Throttle System (ATS) 4.1.2 Activate Automatic Flight Control System (AFCS) 4.1.3 Request an Altitude of 250FT 	Pilot must be able to control the autopilot from the GCS, which includes enabling ATS, AFCS. Requesting speed and altitude must also be possible from the GCS	Primary
Obstacle Visualization and Avoidance	 1.0 Obstacle detection alert 2.1.1 LiDAR Points ON 2.1.2 LiDAR Points OFF 2.2.1 LEDDAR Points ON 2.2.1 LEDDAR Points OFF 	GCS must alert pilot when the sensors detects objects, The pilot must be able to declutter the LiDAR and LEDDAR points, The pilot must be able to rotate the view angle of the objects from the GCS, The GCS must be able to generate and load flightpath file,	Secondary Primary(Alerts)

	 3.0 Rotate View Angle of Obstacle 4.1 Receive Generated Avoidance Flight Path File 4.2 Load Generated Avoidance Flight Path File on UAV 4.3.1 Check LiDAR and LEDDAR Points 4.3.2 Maneuver UAV to Avoid LiDAR and LEDDAR Points 		
UAV Status	1.1 Battery, Fuel, Fuel Flow, RPM and Oil Level 2.1.1 GPS Availability 2.1.1.1 GPS Connection Alert 2.2.2.1 Data Link Connection Alert 2.3.1 Payloads Availability 2.3.2.1 Payloads Connection Alert 3.2 Position	Engine parameters must be shown to the pilot. Alerts on availability of GPS, connection and datalink and payloads must be available, Position of aircraft must be easily seen	Primary
Waypoints Creation	Initialize Waypoints Creation 2.1.1 IAS 2.1.2 TAS 2.2 Change Speed at Waypoint 4.1 Fly in/out 4.2 recovery Point	Pilot must be able to create waypoint, Must be able to set the speed (at least IAS and TAS), Must be able to set the fly mode (at least fly in/out), GCS must have the recovery button	Primary
Entity Tracking Using the Camera	Verify Camera Position 2.1 Control Pitch/Elevation 2.2.1 Zoom In 2.2.1 Zoom Out 2.3 Control Camera Bearing 4.0 Track Target	GCS must show the camera position, The GCS must display the pitch, elevation, bearing and range, The GCS must identify when the camera is in tracking mode	Secondary
RADAR Control	 1.0 Confirm RADAR Heading 2.1 Ream Beam Ground Map (RGBM) 2.2 Stationary Aperture RADAR (SAR) 2.3 Moving Target Indicator (MTI) 3.0 Confirm RADAR range 	RADAR Heading must be displayed on the GCS, The RADAR modes must be selectable (at least, RGBM, SAR and MTI), The GCS must show the RADAR range	Secondary

Flight Plan Execution1.1 Browse File to Load 1.2 Confirm File load 1.3 Cancel File load 2.1 Initialize file plan creation 2.2 Input file Name 2.3 Confirm File Name 2.4 Cancel File Creation 3.0 Save Flight Plan Data 4.0 Upload Flight Data on UAV	Pilots must be able to load flightpath from file, Pilots must be able to create flightpath file from scratch, Pilots must be able to name the created flightpath file, Pilots must be able to save flightpath file	Primary
--	---	---------

APPENDIX I - DRAG LIMITS DEFINITION VAP-XT CODE

```
var Limit type Rect;
Limit.Left = .DragLimits.Left - ExtentGroup.Extent.Left +
Container.Position.X;
Limit.Right = .DraqLimits.Right - ExtentGroup.Extent.Right +
Container.Position.X;
Limit.Top = .DragLimits.Top - ExtentGroup.Extent.Top +
Container.Position.Y;
Limit.Bottom = .DragLimits.Bottom - ExtentGroup.Extent.Bottom +
Container.Position.Y;
var Limit2 type Rect;
Limit2.Left = Limit.Left - 2*.EdgeSpringLimit;
Limit2.Right = Limit.Right + 2*.EdgeSpringLimit;
Limit2.Top = Limit.Top + 2*.EdgeSpringLimit;
Limit2.Bottom = Limit.Bottom - 2*.EdgeSpringLimit;
if(Limit.Left < Limit.Right && Limit.Bottom < Limit.Top)
{
ControlPos.Value.X = ClipToRange(ControlPos.Value.X,
Limit2.Left, Limit2.Right);
ControlPos.Value.Y = ClipToRange(ControlPos.Value.Y,
Limit2.Bottom, Limit2.Top);
.Position.X = ClipToRange(ControlPos.Value.X, Limit.Left,
Limit.Right);
.Position.Y = ClipToRange(ControlPos.Value.Y, Limit.Bottom,
Limit.Top);
} else {
.Position = ControlPos.Value;
}
if (.EnableDragOver)
{
.evEndDrag(.PreviousPosition);
}
else
{
.Position = .PreviousPosition;
}
```

APPENDIX J - PANELS SWAP VAPS-XT CODE

```
var thisPage type UShort = 0;
var swapView type UShort = 0;
// define specific swapping border limits depending if swapping
from left or right screen
if (Page0.IdlePosition.X>1920)
if (540 px < e.Y&&e.X<2200 px)
\{swapView = 2\}
else if (540 \text{ px} > e.Y\&\&e.X < 960 \text{ px})
{ swapView = 0}
else if (1080 px > e.Y&&e.X>2200 px&&e.X<3200 px)
{ swapView = 1}
else if (540 px > e.Y&&e.X>960 px&&e.X<2200 px)
{ swapView = 3}
else if (540 px < e.Y&&e.X>3200 px)
{ swapView = 5}
else if (540 \text{ px} > e.Y\&\&e.X>3200 \text{ px})
\{ swapView = 4 \};
}else{
if (540 px < e.Y&&e.X<1600 px)
\{swapView = 2\}
else if (540 \text{ px} > e.Y\&\&e.X < 960 \text{ px})
{ swapView = 0}
else if (1080 px > e.Y&&e.X>1600 px&&e.X<3200 px)
{ swapView = 1}
else if (540 px > e.Y&&e.X>960 px&&e.X<1600 px)
{ swapView = 3}
else if (540 px < e.Y&&e.X>3200 px)
{ swapView = 5}
else if (540 \text{ px} > e.Y\&\&e.X>3200 \text{ px})
\{ swapView = 4 \};
}
var swapPage type UShort = View2Page.Element[swapView];
var thisView type UShort = Page2View.Element[thisPage];
// disallow swapping with map and PFD if swapping from the right
screen
if (Page0.IdlePosition.X >= 2200 && (swapPage == 2 || swapPage
== 3))
{
```

```
Page0.EnableDragOver = Bool::FALSE;
}
else
{
Page0.EnableDragOver = Bool::TRUE;
.doMovePageToView(thisPage, swapView);
.doMovePageToView(swapPage, thisView);
}
//doMovePageToView Function
var page type UShort = e.Value1;
var view type UShort = e.Value2;
if(0 == page) \{
Page0.PreviousPosition = Page0.Position;
Page0.Position = ViewPos.Element[view];
Page0.ScaleFactor = ViewScale.Element[view];
} else if(1 == page) {
Page1.PreviousPosition = Page1.Position;
Page1.Position = ViewPos.Element[view];
Page1.ScaleFactor = ViewScale.Element[view];
} else if(2 == page) {
Page2.PreviousPosition = Page2.Position;
Page2.Position = ViewPos.Element[view];
Page2.ScaleFactor = ViewScale.Element[view];
} else if(3 == page) {
Page3.PreviousPosition = Page3.Position;
Page3.Position = ViewPos.Element[view];
Page3.ScaleFactor = ViewScale.Element[view];
} else if(4 == page) {
Page4.PreviousPosition = Page4.Position;
Page4.Position = ViewPos.Element[view];
Page4.ScaleFactor = ViewScale.Element[view];
} else if(5 == page) {
Page5.PreviousPosition = Page5.Position;
Page5.Position = ViewPos.Element[view];
Page5.ScaleFactor = ViewScale.Element[view];
}
```

View2Page.Element[view] = page; Page2View.Element[page] = view;

<Loiter id="5" lat="12.788940909651565" lon="45.051704332643752" alt="1000" AltitudeType="0" Time="0" SpeedType="0" LoiterSpeed="90" Procedure="1" EntryType="0" TurnDirection="0" NumberOfTurns="1" Radius="1.0" Bearing="0" Length="0.0" /> <Loiter id="6" lat="12.794723660132783" lon="45.003761902816507"</pre> alt="1000" AltitudeType="0" Time="0" SpeedType="0" LoiterSpeed="90" Procedure="2" EntryType="0" TurnDirection="1" NumberOfTurns="1" Radius="0.9" Bearing="-45" Length="1.1" /> </Mission>

<Waypoint id="6" lat="12.794723660132783" lon="45.003761902816507" alt="1000" AltitudeType="0" WaypointType="0" LocalType="3" VerticalType="0" SpeedType="0" Time="0" SpeedToWaypoint="90" Procedure="2" Course="0" NextWaypoint="0" RotationSpeed="0" />

<Waypoint id="5" lat="12.788940909651565" lon="45.051704332643752" alt="1000" AltitudeType="0" WaypointType="0" LocalType="3" VerticalType="0" SpeedType="0" Time="0" SpeedToWaypoint="90" Procedure="1" Course="0" NextWaypoint="6" RotationSpeed="0" />

<Waypoint id="2" lat="12.835700658327863" lon="45.077296025943234" alt="600" AltitudeType="0" WaypointType="1" LocalType="0" VerticalType="0" SpeedType="0" Time="0" SpeedToWaypoint="90" Procedure="0" Course="0" NextWaypoint="5" RotationSpeed="0" />

<Waypoint id="1" lat="12.882169948384567" lon="45.081340702106374" alt="500" AltitudeType="0" WaypointType="0" LocalType="0" VerticalType="0" SpeedType="0" Time="0" SpeedToWaypoint="125" Procedure="0" Course="0" NextWaypoint="2" RotationSpeed="0" />

<Mission version="1.0">

APPENDIX K - DEMO MISSION XML FILE

REFERENCES

- "Development and operation of UAVs for military and civil applications = Développement et utilisation des avions sans pilote (UAV) pour des applications civiles et militaires.," 2000.
- [2] R. Austin, *Unmanned aircraft systems: UAVS design, development, and deployment.* Chichester, West Sussex, U.K: Wiley, 2010.
- [3] M. D. F. Bento, "unmanned aerial Vehicles: An Overview." InsideGNSS.
- [4] D. J. Pines and F. Bohorquez, "Challenges Facing Future Micro-Air-Vehicle Development," *Journal of Aircraft*, vol. 43, no. 2, pp. 290–305, Mar. 2006.
- [5] "MQ-1 Predator Unmanned Combat Aerial Vehicle | Military-Today.com." [Online]. Available: http://www.military-today.com/aircraft/mq1_predator.htm. [Accessed: 18-Nov-2016].
- [6] T. Rogoway, "Why The USAF's Massive \$10 Billion Global Hawk UAV Is Worth The Money," *Foxtrot Alpha*. [Online]. Available: http://foxtrotalpha.jalopnik.com/why-the-usafs-massive-10-billion-global-hawk-uav-was-w-1629932000. [Accessed: 21-Nov-2016].
- [7] "Datasheet_GH_Block_40.pdf.".
- [8] J. F. Keane and S. S. Carr, "A brief history of early unmanned aircraft," *Johns Hopkins APL Technical Digest*, vol. 32, no. 3, pp. 558–571, 2013.
- [9] J. Stoff, *Historic aircraft and spacecraft in the Cradle of Aviation Museum*. Mineola, N.Y.: Dover Publications, 2001.
- [10] "Nesta Drones." [Online]. Available: http://www.nesta.org.uk/drones-history-flying-robots. [Accessed: 18-Nov-2016].
- [11] J. D. Anderson, Introduction to flight, 7th ed. New York: McGraw Hill, 2012.
- [12] "Oemichen." [Online]. Available: http://www.aviastar.org/helicopters_eng/oemichen.php. [Accessed: 18-Nov-2016].
- [13] "History of Quadcopter." [Online]. Available: http://ffden-2.phys.uaf.edu/webproj/212_spring_2014/Clay_Allen/clay_allen/history.html.
- [14] "Convertawings Model A." [Online]. Available: http://www.aviastar.org/helicopters_eng/convertawings.php. [Accessed: 18-Nov-2016].
- [15] "Flight Global Archive." [Online]. Available: https://www.flightglobal.com/pdfarchive/view/1960/1960%20-%201367.html. [Accessed: 17-Nov-2016].
- [16] "Strange Vehicles: Curtiss-Wright VZ-7," Diseno-art, 09-Sep-2012. .
- [17] "Amazon Drones: Amazon Unveils Futuristic Delivery Plan CBS News." [Online]. Available: http://www.cbsnews.com/news/amazon-unveils-futuristic-plan-delivery-bydrone/. [Accessed: 06-Oct-2015].
- [18] O. Zeitoun, "A Distributed System Interface for a Flight Simulator," 2014.
- [19] R. L. Page, "Brief history of flight simulation," SimTecT 2000 Proceedings, pp. 11–17, 2000.
- [20] J. Mendes, "UAV Flight Simulator based on ESA Infrastructure."
- [21] FAA, "Aviation Maintenance Technician Handbook General," 2008. [Online]. Available: https://www.faa.gov/regulations_policies/handbooks_manuals/aircraft/amt_handbook/. [Accessed: 18-Mar-2016].
- [22] J. A. Wise, V. D. Hopkin, and D. J. Garland, Eds., *Handbook of aviation human factors*, 2nd ed. Boca Raton: CRC Press, 2010.

- [23] C. Sandom and R. S. Harvey, *Human factors for engineers*, vol. 2. let, 2004.
- [24] K. W. Williams, "A summary of unmanned aircraft accident/incident data: Human factors implications," DTIC Document, 2004.
- [25] Q. R. Waraich, T. A. Mazzuchi, S. Sarkani, and D. F. Rico, "Minimizing human factors mishaps in unmanned aircraft systems," *ergonomics in design*, vol. 21, no. 1, pp. 25–32, 2013.
- [26] ISO, "Ergonomics of human-system interaction Part 210: Humancentred design for interactive systems (ISO 9241-210:2010)," *Part*, vol. 1, p. 44, 2010.
- [27] M. Maguire, "Methods to support human-centred design," *International Journal of Human-Computer Studies*, vol. 55, no. 4, pp. 587–634, Oct. 2001.
- [28] N. Stanton, *Handbook of human factors and ergonomics methods*. Boca Raton: CRC Press, 2005.
- [29] N. A. Stanton, Ed., *Human factors methods: a practical guide for engineering and design*, Reprinted. Aldershot: Ashgate, 2011.
- [30] D. L. Cotner, "Computer program for performance prediction of tandem-rotor helicopters," 1985.
- [31] "Why the CH-47F Chinook gives Indian pilots a high," *Rediff*. [Online]. Available: http://www.rediff.com/news/report/why-the-ch-47f-chinook-gives-indian-pilots-a-high/20160903.htm. [Accessed: 03-Oct-2017].
- [32] AUVSI, "UAS-Uses-Saving-Time-Saving-Money-Saving-Lives.pdf," *The Association for Unmanned Vehicle Systems International (AUVSI)*, 2012. [Online]. Available: https://epic.org/events/UAS-Uses-Saving-Time-Saving-Money-Saving-Lives.pdf. [Accessed: 09-Jan-2015].
- [33] D. K. Barton, Radar system analysis and modeling. Boston, MA: Artech House, 2005.
- [34] LeddarTech Inc., "Leddar Solid-State LiDAR: M16 Multi-element sensor module," *Leddartech*, 2015. [Online]. Available: http://leddartech.com/modules/m16-multi-elementsensor-module. [Accessed: 29-Jul-2017].
- [35] "Presagis News | Embedded Graphics Newsletter Spring 2010." [Online]. Available: http://www.presagis.com/resources/newsletter/embedded_graphics_newsletter_-_spring_2010/. [Accessed: 05-Aug-2017].
- [36] GitHub, "LCM: Lightweight Communications and Marshalling (LCM)," 2016. [Online]. Available: https://lcm-proj.github.io/. [Accessed: 15-Jul-2017].
- [37] J. Haber and J. Chung, "Assessment of UAV operator workload in a reconfigurable multitouch ground control station environment," *J. Unmanned Veh. Sys.*, vol. 4, no. 3, pp. 203– 216, Apr. 2016.
- [38] D. L. Stone and Open University, Eds., *User interface design and evaluation*. Amsterdam; Boston, Mass: Elsevier : Morgan Kaufmann, 2005.
- [39] J. N. Briggs, *Target detection by marine radar*. Stevenage, Herts., UK: Institution of Electrical Engineers, 2004.
- [40] Rotax, "Rotax 912 iS/iSc Sport Rotax Aircaft Engines," 2017. [Online]. Available: http://www.flyrotax.com/produkte/detail/rotax-912-is-isc-sport.html. [Accessed: 25-Apr-2017].
- [41] J. Huddlestone, D. Harris, D. Richards, S. Scott, and R. Sears, "Dual Pilot and Single Pilot Operations – Hierarchical Task Decomposition Analysis of Doing More with Less," in *Engineering Psychology and Cognitive Ergonomics*, vol. 9174, D. Harris, Ed. Cham: Springer International Publishing, 2015, pp. 365–376.

- [42] A. Hobbs and B. Lyall, "Human Factors Guidelines for Unmanned Aircraft Systems," *Ergonomics in Design*, vol. 24, no. 3, pp. 23–28, Apr. 2016.
- [43] "MicroPilot World Leader in Professional UAS Autopilots | Product Page Horizon mp." [Online]. Available: https://www.micropilot.com/products-horizonmp.htm. [Accessed: 07-May-2017].
- [44] FAA, "Human Factors Design Standard," 2003. [Online]. Available: http://www.tc.faa.gov/its/worldpac/techrpt/ct03-5.pdf. [Accessed: 05-May-2017].
- [45] Department of Defense, "Handbook for Human Engineering Design Guidelines," 1995. [Online]. Available: http://www.deepsloweasy.com/HFE%20resources/MIL-HDBK-759C.pdf. [Accessed: 05-May-2017].
- [46] Department of Defense, "Department of Defense Design Criteria Standard," 1999. [Online]. Available: http://www.product-lifecycle-management.com/download/mil-std-1472f.pdf. [Accessed: 05-May-2017].
- [47] S. A. NATO, "STANAG 4586 NAVY (Edition 2) Standard Interface of UAV Control System (UCS) for NATO Interoperability." 2007.
- [48] A. Hobbs and B. Lyall, "Human Factors Guidelines for Unmanned Aircraft System Ground Control Stations.," 2015. [Online]. Available: https://pdfs.semanticscholar.org/ae40/13db11a72e48901bfeb9a0c3ac8eacd5fc81.pdf. [Accessed: 31-May-2017].
- [49] "Roy Aéronef & Avionique Simulation." [Online]. Available: http://www.raasi.ca/. [Accessed: 07-Sep-2017].
- [50] "LuciadLightspeed." [Online]. Available: http://www.luciad.com/solutions/luciadlightspeed. [Accessed: 31-Aug-2017].
- [51] webmaster, "Presagis COTS Modeling & Simulation Software | Presagis." [Online]. Available: http://www.presagis.com/. [Accessed: 08-Sep-2017].
- [52] webmaster, "UAV CRAFT | Presagis." [Online]. Available: http://www.presagis.com/products_services/products/modelingsimulation/craft_series/uav_craft/#overview. [Accessed: 31-May-2017].
- [53] "Scorpion Technologies LTD. X-Series." [Online]. Available: http://www.controlgrips.com/controlgripsdetails.php?productid=32#. [Accessed: 07-Sep-2017].
- [54] "News CH Products, News." [Online]. Available: http://www.chproducts.com/13-28504-News.php?id=106. [Accessed: 07-Sep-2017].
- [55] "TinyXml: Main Page." [Online]. Available: http://www.grinninglizard.com/tinyxmldocs/index.html. [Accessed: 08-Jul-2017].
- [56] "Wavefront OBJ: Summary from the Encyclopedia of Graphics File Formats." [Online]. Available: http://www.fileformat.info/format/wavefrontobj/egff.htm. [Accessed: 15-Jul-2017].
- [57] "OpenGL The Industry Standard for High Performance Graphics." [Online]. Available: https://www.opengl.org/. [Accessed: 15-Jul-2017].
- [58] "Velodyne LiDAR." [Online]. Available: http://velodynelidar.com/index.html. [Accessed: 15-Jul-2017].
- [59] NASA, "Task Load Index (NASA-TLX)." [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20000021488.pdf. [Accessed: 02-Jun-2017].