AUTOMATION OF CONCEPTUAL DESIGN OF VERTICAL TAKE-OFF UAVS – DEMONSTRATION MODEL

Ankush Rathore*, Arvind Sinha*, Joseph Khreish*
**** Sir Lawrence Wackett Centre for Aerospace Design Technology, Royal Melbourne Institute of Technology, GPO Box 2476V, Melbourne, Victoria, 3001, Australia***
Ph +61 3 9645 4541, Fax +61 3 9645 4534, Email: arvind.sinha@rmit.edu.au

Keywords: *VTUAV, Concept Design, Automation*

Abstract

The development and induction in-service of Unmanned Air Vehicles (UAV) systems in a variety of civil, paramilitary and military roles have proven valuable on high-risk missions. These UAVs based on fixed wing configuration concept have demonstrated their operational effectiveness in recent operations. New UAVs based on rotary wing configuration concept have received major attention worldwide, with major resources committed for its research and development.

The design process of a rotary-wing aircraft was re-visualised from an unmanned perspective to address the requirements of rotary-wing UAVs – Vertical Take-off UAVs (VTUAV). It investigated the conventional helicopter design methodology for application in UAV design and developed a modified design process for VTUAV addressing the requirements of unmanned missions by providing remote command-and-control capabilities. The modified design methodology is automated to address the complex design evaluations and optimisation process.

In this paper, an illustration of the automated design process developed for VTUAVs is provided through a series of inputs of the requirements and specifications, resulting in an output of a proposed VTUAV design configuration for "design decision support".

The VTUAV automated design process has been developed to pioneer an aerospace design tool for further detailed development and application as a – Design Decision Support System.

1 Introduction

Aerospace industries around the globe have invested major resources for the design and development of Unmanned Aerial Vehicles (UAV) as a viable alternative to the manned aircraft, on various high risk missions. UAVs are emerging as the next generation of airborne reconnaissance assets due to their ability to penetrate deep into enemy airspace, loiter on designated target areas, detect lethal weapons and conduct damage assessments. They are the ideal aerial platforms for emerging warfare concepts of 21st century.

The major developments of UAVs to-date have been Fixed-Wing (FW) focussed. FW-UAVs have undergone major operational trials and are in various stages of induction and utilisation. The present inventory of FW-UAVs addresses the military (strategic & tactical) and civil (Relief, monitor, & research) operational needs. In addition to FW-UAVs the missions envisaged for UAVs require vertical take-off and landing capabilities – hence the requirement of a Rotary-Wing (RW) UAV. The development of RW-UAVs has gathered momentum to fill-in the operational gaps of FW-UAVs, and it is envisaged that it will play an increasingly important role in both military and civilian missions. This is due to their inherent take-off and landing capabilities that allow operation from unprepared sites, with no requirements of runways, net, launch rail or parachutes. Further, its ability to hover and manoeuvre in limited spaces makes them suitable for operation in restricted or urban terrain, beyond the operational capabilities of FW-UAVs. In addition, to their small size, the viability of detection and destruction is low. Industrial programs have been initiated for the research and development of RW-UAVs, also referred as "Vertical Takeoff UAVs" (VTUAVs). The present inventory of VTUAVs only addresses close-to-ground reconnaissance and surveillance requirements. Of late, the operational need for combat support is also being considered.

Various concepts, beyond the conventional helicopter (Fuselage, and main & tail rotor system) are being proposed and include rocket and magnetic propulsion take-off. Jet-lift principle is another concept for addressing vertical take-off issues. These concepts have not gathered much support; making the conventional helicopter concept as the base-line design concept for the development of VTUAVs.

The rotary-wing design methodology has been traditionally applied for the conceptual design of helicopters. The methodology is based on a series of estimations and evaluations of weight, power, speed, and drag for sizing the helicopter. The computations refers to past helicopter design data and statistics to provide the design benchmarks for the development of a variable conceptual configuration to meet the slated design specifications.

The design methodology is analytical and requires major iterations for the design optimisation; hence it is time intensive. To address the optimisation process more efficiently an automated design process is needed. The traditional design methodology needs to incorporate an autonomous component by the development of an expert system to generate the conceptual design configuration of the VTUAV from the slated mission requirements.

2 Automation

The design methodology [1] needs to be automated to design an expert system based on 'flexible automation' principle [2, 3] to generate the conceptual design configuration of the VTUAV from the slated mission requirements. The expert system to be developed to address automation of the conceptual design methodology needs to display the conceptual design configuration on a Graphical User Interface (GUI).

The design of the expert system requires the selection of a suitable programming language and the database in a programming environment to implement the supervisory and developmental logic of the design methodology. This selection is to be based on inter-compatibility as follows [4-7]:

- <u>Programming environment</u>: Microsoft Visual Studio .Net was selected to provide the dynamic environment for implementing the design methodology developed due to its modular and object-oriented base;
- <u>Programming language</u>: Microsoft Visual C++ was selected from the various inbuilt languages of Microsoft Visual Studio .Net to program required modules of the design methodology as 'reusable classes' due to its system and web interaction ability; and
- <u>Database</u>: Microsoft SQL Server was selected to be the relational database management system for its ability to interact with Microsoft Visual C++ to analyse and store exchanges through its administrative tools of database analysis, maintenance, and administration. The connectivity between Microsoft Visual C++ and Microsoft SQL Server was established through a customised portable file-connection string. The application consistency across windows and web-based platforms was maintained with Microsoft .NET Framework 1.1, an inherent feature of Microsoft Visual Studio .Net environment.

The design methodology of VTUAVs needs to be analysed to identify various reusable classes that provide computational details of the design stages. These classes, their variables and functional methods will provide the base design for the expert system. Accordingly the classes need to be categorised.

2.1 Base Category

The various stages of the design methodology for VTUAV design were analysed to identify the classes for the base design category of the expert system. The investigation established the functionality of the various design stages to be distinctive, and derived from common inputs. Thus, each stage of the design methodology can be represented by a "variable-dependent" and "method-independent" class. The investigation also established the class of design-input (requirements and constraints) needs at Stage 1, and defined design-output at Stage 10 of the design methodology. Thus, the various classes, their corresponding variables and methods based on base category and identified by the analysis of the design methodology stages are as follows:

- 'Design-input' class: The class provides validated inputs to the VTUAV design methodology. The class variables are the design requirements and constraints identified from the attributes of the components of the refined design architecture. The class methods comprise of acquiring design requirements & constraints, and validating specific design parameters including payload & maximum take-off weight; speed, endurance & range; and power & maximum take-off weight against the established relative benchmarks;
- 'Mission-time' class: The class estimates and validates the time required to complete the slated mission. The class variables are the generic mission phases - take-off, transit, ingress, engage, disengage, egress, transit back, and landing. The class methods include an aggregate of time periods of the mission phases to evaluate the mission time, and the validation of mission time against the slated endurance time (design input). Since the class involves mission requirements (Table 1), there are independent "mission time" classes required to compute mission time based on the mission requirements. The variables of all these "mission time" classes are similar but for 'engage' and 'disengage' class variables that are mission specific;
- 'Payload' class: The class involves the selection of mission systems that meet slated mission requirements. The class methods involve the addition of the selected mission systems weights to compute the payload weight. Since the class involves the analysis of the mission requirements that are the functional characteristics of the components of mission systems, classes are to be designed based on the elements of the components of the mission system. The variables of these classes are the elements of the components of the mission system;
- 'Database-I' class: The class involves identification of VTUAVs based on design requirements for the estimation of power. The class variables are the design requirements comprising of power, and also include the make/model and number of the engines. The class methods include matching of the design requirements with database for similar VTUAVs, and selection of engine based on the model and number of engines onboard these VTUAVs;
- 'Fuel' class: The class involves estimation of mission fuel. The class variables are power, specific fuel combustion (sfc), mission time, and mission fuel. The class method includes computation of weight of total fuel required for the mission including the reserve and trapped fuel in lines;
- 'Structural weight' class: The class involves evaluation of the structural weight. The class variables are MTOW, payload and engine weight. The class method includes the evaluation of remainder weight of various airframe sections of the VTUAV based on engine weight as 15% of the MTOW;
- 'Vertical drag' class: The class calculates the vertical drag and involves development of inboard and external profile of the VTUAV based on mission systems load stations, governed by its functional characteristics. The class variables are drag coefficient, disc loading, projected area in remote wake, and dimensions of selected mission systems. The class methods comprise of calculation of the vertical drag based on the computation of

projected area in remote wake from the external profile developed of the VTUAV;

- 'Rotor system-I' class: The class involves the computation of the size of main rotor system and the number of rotor blades based on the coaxial configuration; other configurations have not been addressed being an illustration. The class variables are MTOW, vertical drag, disc loading, forward speed, rotational speed, & tip speed, and also include the air density. The class methods comprise of computation of the rotor radius from disc area, and computation of solidity to determine the number of rotor blades;
- 'Rotor system-II' class: The class involves the selection of blade airfoil. The class variables in airfoil design include the type (symmetrical/cambered) & thickness of airfoil section, and the positioning of maximum camber. The class method comprises of selection of the airfoil section;
- 'Drag' class: The class involves evaluation of parasite and profile drag for further computation of power required to sustain forward flight at maximum speed. The class variables include MTOW, payload weight, dialog factor, and coefficient of friction. The class methods comprises of evaluation of the coefficients of parasite and profile drag;
- 'Power' class: The class involves the evaluation of power required in forward flight for maximum speed. The class variables include MTOW, air density, solidity, rotor radius & disc area, forward speed, coefficient of profile drag, angular velocity, and tip speed. The class method comprises of power evaluation at maximum speed in forward flight;
- 'Database-II' class: The class involves the reselection of engine based on significant variance between the evaluated power at previous class and estimated power at 'Database-I' class. The class variables include evaluated and estimated power. The class methods comprise of comparison of evaluated power with the estimated power and reselection of the engine from the database;
- 'Airframe-section weight' class: The class involves computation of the structural weight of the airframe sections including rotor blades, propulsion, electrical, avionics, optional nacelles and dry systems of VTUAV. The class variables include MTOW, fuselage length & projected area, and engine weight. The class methods involve computation of structural weights of airframe sections of VTUAV for their comparison with the evaluated structural weight at 'Structural weight' class;
- 'Balancing' class: The class involves the 'weight and balance' exercise to compute the longitudinal cg. The class variables include the load stations of airframe sections and mission systems. The class methods comprise of evaluation of longitudinal cg; and
- 'Design-output' class: The class involves the refinement of inboard and external profile of the VTUAV. The class variables include the selected mission systems and the airframe sections. The class method comprises of refinement of inboard and external profiles of the VTUAV.

2.2 Database Category

The design of the expert system involves another category of classes; the databases of the design parameters of various VTUAVs in-development/in-service and their power-plants. The databases are used in the design methodology to refer to past VTUAV design data for estimations. It is also used for information collection, and comparisons between various VTUAVs. These databases are represented by editable and non-editable classes as follows and comprises of 'search' functionality for the analysis of collected information:

- 'Non-editable database' class: The class provides 'read-only' facilities of the design parameters of various VTUAVs. The class variables include design parameters and the class methods comprise of 'search and sort' function as required by the designer; and
- 'Editable database' class: The class provides the option to update the collated VTUAV

design parameters. The class variables include design parameters and the class methods comprise of update of edited design parameters.

2.3 Utility Category

Additional category of classes is identified based on the need to execute and record interim computations involved in the design methodology of VTUAVs. These classes based on utility category comprise of operating-system classes including "Calculator", "Notepad", and "WordPad" to execute interim computations. To record these interim computations, classes that mail and print the design-configurations of the VTUAV are identified. The operating system class of "Send-by-mail" provides mailing functions, but an independent class is designed as follows to provide printing functions:

- <u>'Print' class</u>: The class involves print of the database and the design configuration of VTUAV. The class variables include page-set-up, page-copies, and page-orientation. The class method comprises of printing of the design-configuration/database of VTUAVs.

2.4 Navigation Category

There is a need for a category of classes to address the navigation between 'mission-time' classes. Hence, classes based on navigation category are designed on the mission requirements as follows:

- <u>'Mission categories' class</u>: The class provides the option to select the desired category of mission requirements based on the operational needs. The class variables are various mission categories including generic, relief & monitoring, research, strategic, and tactical. The class method comprises of navigation to mission requirement based on the selected mission category; and
- <u>'Mission requirements' class</u>: The class involves various mission requirements based on the selected mission category. The class variables are mission requirements and the class method comprises of navigation to mission phases of selected mission requirement for computation of mission time.

2.5 Framework Category

The development of the application framework in .NET [11, 12] programming environment will comprise of another category of classes. These classes are based on the internal framework of Microsoft Visual C++ involving "application", "view", and "document" classes. The class variables include global/private variables that are used in 'base classes'. The class methods comprises of user-application interactions as follows:

- <u>'Application' class</u>: It responds to menu and taskbar interactions;
- <u>'View' class:</u> It responds to interactions involving input devices including mouse and keyboard; and
- <u>'Document' class</u>: It involves database usage vis-à-vis configured connection string.

A total of 27 classes were identified based on the five categories of base, database, utility, navigation, and framework for the automation of the design methodology.

Examples of the windows user interface of the automated design process is illustrated in figure 1, 2, and 3.

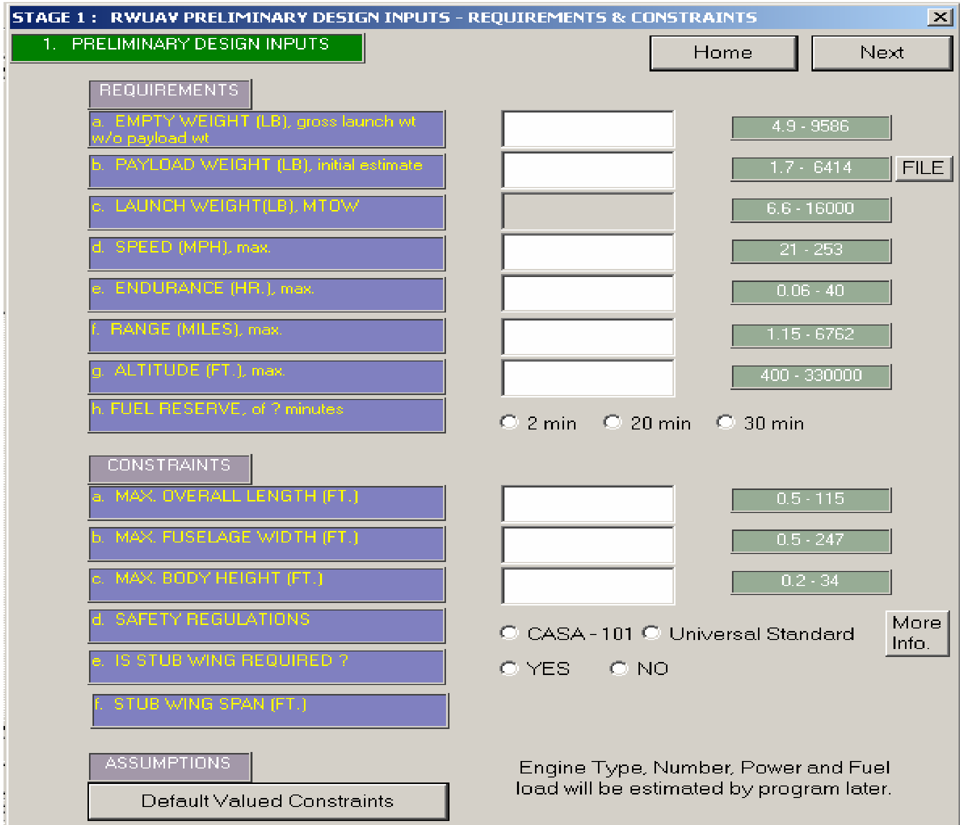

Fig 1. The output dialog of the 'Design input' Class

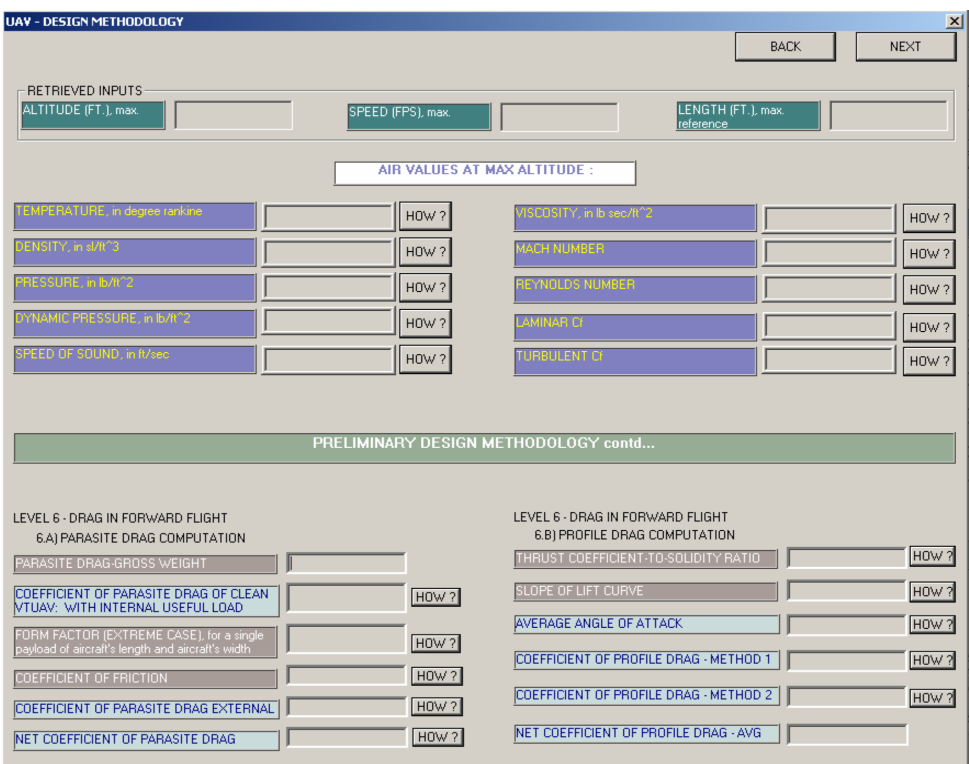

Fig 2. The output dialog of the 'Drag' Class

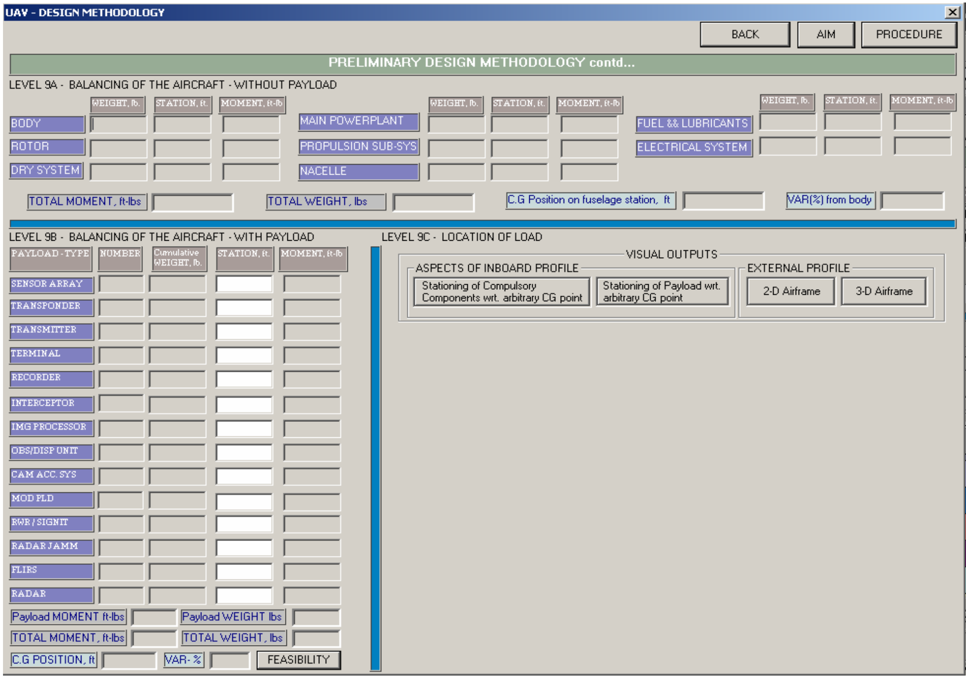

Fig 3. The Output Dialog of 'Balancing' and 'Design-output' Classes

3 Results and Discussion

The development of the conceptual design methodology for VTUAVs and its subsequent automation was based on the traditional helicopter design methodology. A systems approach was adopted to gain an insight in the required design methodology from a systems perspective. The autonomous component was integrated in the helicopter design methodology followed by the development of an expert system to automate the design.

The automation involved the selection of suitable programming language and the database in a programming environment for the design of an expert system to provide output dialogs based on the updated rotary-wing design methodology. Microsoft Visual C++ was selected as the programming language; database was on Microsoft SQL Server and the dynamic environment was provided by Microsoft Visual Studio .Net. The design of the expert system comprised of the identification of various reusable classes, variables and methods from the analysis of updated design methodology. The dialogs of the expert system covered the programming of the classes identified to achieve the desired functionality.

The design of expert system resulted in the identification of five class categories based on design, database, service, transition and support; under which 27 reusable classes were covered to provide the computational details of the design stages of updated rotary-wing methodology as follows: a) Base category – Design-input, mission-time, payload, database-I, fuel, structural weight, vertical drag, rotor system-I, rotor system-II, drag, power, database-II, airframe-section weight, balancing, & design-output; b) Database category – Non-editable, & editable database; c) Utility category – Calculator, notepad, word-pad, send-by-mail, & print; d) Navigation category – Mission categories, & mission requirements; and e) Framework category – Application, view, & document. The corresponding class variables and class methods for each of these were also identified and summarised for ready reference.

The dialogs of the expert system are the programmed outputs of the classes identified.

The dialogs comprised of various controls including buttons, sliders, option-boxes, check-boxes, text-boxes, and data-grids to implement the class methods based on the class variables identified. These buttons provided for design decisions and validations based on established statistical benchmarks.

The automated expert system was developed to assist in the generation of conceptual design configuration of VTUAVs based on the user slated inputs comprising of mission requirements derived from operational needs and operational environment.

4 References

[1] Sinha K. A. *Introduction to Preliminary Helicopter Design*. Department of Aerospace Engineering, RMIT, Melbourne, Australia, 2002.

[2] *Introduction to Expert Systems Tutorial.* 2002, Module 1, [Available Online] URL: http://www.wordreference.com/definition/automation , Date Updated: 20/7/2002, Date Downloaded: 21/12/2005.

[3] Grover, P. *Introduction to Automation.* Automated Production Systems, Lecture 1, University of Tennessee, 2005. [Available Online] URL: http://www.utm.edu/departments/engin/lemaster/Auto%20Prod%20Sys/Lecture%2001.pdf, Date Downloaded:11/12/2005.

[4] *Word Reference Dictionary.* 2005, [Available Online] URL: http://en.wikipedia.org/wiki/Programming_paradigm, Date Downloaded: 23/12/2005.

[5] *Open Database Connectivity*. [Available Online] URL: http://www.pcigeomatics.com/cgi-bin/pcihlp/GDB%7CSupported+File+Formats%7CODBC+(Open+Database+Connectivity), Date Downloaded: 25/12/2005.

[6] *Microsoft .NET Framework Developer Centre*. 2005, [Available Online] URL: http://msdn.microsoft.com/netframework/downloads/framework1%5F1/, Date Downloaded: 25/12/2005.

[7] *Microsoft Visual C++ Developer Centre*. 2005, [Available Online] URL: http://msdn.microsoft.com/visualc/, Date Downloaded: 26/12/2005.