A Prototype GUI for Unmanned Air Vehicle Mission Planning and Execution ⋆

Sean R. Perry [*] **James H. Taylor** [**]

[*] *Environment Canada, Calgary, Alberta, Canada X4X 4X4*
(e-mail: sean.perry@ec.gc.ca).
[**] *University of New Brunswick, Fredericton, N. B., Canada E3B 5A3*
(e-mail: jtaylor@unb.ca)

Abstract: This paper describes the development of a graphical user interface which provides unique mission control services to unmanned aerial vehicle (UAV) operators. The software has two modes of operation, Planning Mode and Execution Mode. In Planning Mode an operator can readily simulate a desired flight plan to determine mission feasibility. Key features of this mode include integration of forecast weather data and interactive trajectory planning. Harsh North Atlantic weather can jeopardize a mission, so running a simulation with predicted wind conditions may greatly reduce the risk of a UAV loss. In Execution Mode the operator can deploy an autonomous vehicle, change the route "on the fly", hold the aircraft over a point of interest, and return the UAV to home base whenever necessary. In both modes the predicted and actual weather conditions are the key to mission success.
All weather data required for these functions is automatically downloaded via the internet. Operation of the software, including interactive trajectory planning, weather observation, mission planning and execution, is achieved through a user-friendly graphical user interface. Simulation results, flight trajectories, and weather data can be saved for further analysis. Comparing missions with and without weather information, it was determined that meteorological information can have a very significant impact on the mission success or failure. In a typical example mission discussed here, including forecast weather conditions increased the UAV's fuel consumption by 18 % and mission time by 26 %.

Keywords: Unmanned aerial vehicle control, UAV mission planning, graphical user interface, UAV mission control, autonomous vehicles, airborne surveillance, maritime security

1. INTRODUCTION

Operation of an unmanned aerial vehicle (UAV) requires planning and assessing the feasibility of a proposed flight plan and then executing it conveniently and safely. UAV surveillance can be highly beneficial both for security and commercial purposes. In the Canadian context, UAV surveillance can be extremely beneficial for locating lost persons or vehicles, finding and tracking icebergs, detecting illegal bilge dumping or fishing, and monitoring oil spills. Figure 1 depicts a typical scenario: The operator is located at a remote terminal, which has the ability to communicate with the UAV. Before the UAV is deployed, the operator defines and simulates a proposed mission trajectory to determine feasibility, based on forecast weather conditions. Once an acceptable route has been determined, the operator proceeds to execute the mission.

An idealized vision of this activity is depicted in Fig 1. We call the corresponding graphical user interface (GUI) the **U**AV **M**ission **P**lanning and **E**xecution **G**UI (UMPEG). The mission route is shown by the red line in this figure. The UAV proceeds to track this route while constantly scanning for events. Two points of detection have been shown along the aircraft's route, shown in Fig. 1 by the yellow stars, the first being an iceberg sighting and the second an illegal fishing vessel being interdicted. Once an event has been detected, the operator has the option to investigate it by changing the flight trajectory (e.g.,

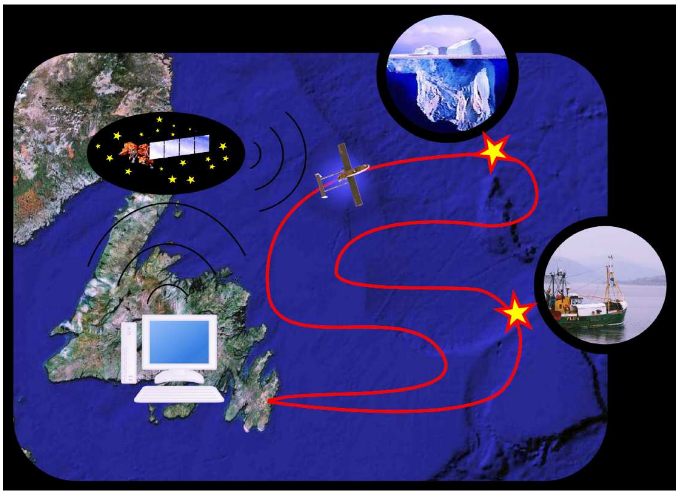


Fig. 1. Architypical UMPEG Scenario

⋆ Sponsored by the RAVEN project, an AIF-funded program led by Dr. Siu O'Young of the Memorial University, Newfoundland, Canada

loitering over the site), or to log it and proceed as originally planned.

For routine UAV deployment to be feasible and sustainable, it is necessary to minimize unnecessary losses of unmanned aircraft, as may happen in adverse weather conditions or on overly ambitious missions leading to fuel becoming exhausted. A planning system to reduce such losses needs to include an aircraft model, an aircraft simulator, a remotely-operated on-board control system, a weather forecasting system, a means of simulating the effect of the predicted weather conditions, and a convenient and effective graphical user interface (GUI). Together, these components allow an operator to define and simulate an anticipated mission and predict the likelihood of success or failure.

Meteorological conditions are the key factor when considering the feasibility of a proposed flight plan, especially in Atlantic Canada (Newfoundland) where the climate is harsh. Therefore, accounting for weather effects is critical in the planning and execution of a UAV mission. This paper addresses the design, implementation and testing of UMPEG. It may be used as part of the Remote Aerial Vehicles for ENviromental monitoring project (RAVEN), O'Young and Hubbard (2007), an Atlantic Innovation Fund (AIF) sponsored program led by Dr. Siu O'Young of the Memorial University of Newfoundland, under which UNB had mission planning and control responsibility. In addition, we anticipate that this paper will serve as a roadmap for the development of similar GUIs for various autonomous vehicle applications.

2. RESOURCES AND BACKGROUND

This project incorporates technology and data from many sources. Integrating data and tools into a software package requires researching, choosing and understanding the main ingredients. The following sections briefly describe the groups, software, and technologies that were involved in the creation of UMPEG.

(1) **RAVEN** O'Young and Hubbard (2007) – The main goal of the RAVEN project is to supplement manned missions through the use of UAVs. Manned aerial surveillance is currently performed off the coast of Newfoundland by Provincial Aerospace Limited (PAL). Using a Beechcraft KingAir 200 aircraft, PAL's missions are usually about 2 000 km long, last 6 to 8 hours, and are typically flown at an altitude of 300 m. In order to successfully develop our GUI, the harsh maritime operating environment needs to be accurately approximated for the purpose of testing proposed flight plans using high fidelity UAV models and synthetic wind inputs based on accurate weather forecasts. By supplementing PAL's current maritime reconnaissance regime with operational UAVs, increased cost efficiency and improved surveillance can be achieved.

(2) **Environment Canada** EC-www (2013a)[1] – Environment Canada is responsible for meteorology through the Meteorological Service of Canada, which produces weather forecasts and severe weather warnings. Environment Canada also produces numerical weather models, i.e., mathematical models of the atmosphere and oceans which can be used to produce short-term weather forecasts based on current weather conditions as inputs, e.g., observations relayed from ground stations, radiosondes or weather satellites.

(3) **MATLAB®** Hahn & Valentine (2013); MathWorks (2013a) – MATLAB is a numerical computing environment and programming language that was created and is maintained by The MathWorks. Matrix manipulation, plotting of functions and data, implementing algorithms, creating user interfaces, and interfacing with programs in other languages are all easily accomplished in MATLAB. It uses the MATLAB programming language commonly called M-code, which provides the user with a full development environment. Commands can be run via the command window, variables can be displayed in the workspace and scripts can be created/edited in the editor. MATLAB also includes facilities for debugging M-code.

MATLAB specializes in numerical computing, but is not limited to that alone. Toolboxes, groupings of highly specialized functions, have been developed for various purposes. Of particular importance here are the Mapping Toolbox™and the Image Processing Toolbox™. The Mapping Toolbox provides the ability to easily display different mapping data and images and perform geo-spacial translations. The Image Processing Toolbox allows MATLAB to properly read, process, and render image files.

MATLAB also provides the ability to create complex graphical user interfaces using its GUI Design Environment or GUIDE™. GUIDE allows GUIs to be created and edited interactively from figures. Callback commands are located in an accompanying M-file. GUIDE developments can include a menu system for easy program navigation. Through GUIDE it is also possible to interact with the MATLAB workspace in a normal manner. Variables can be written to MATLAB-files and loaded back into memory at a later time. GUIDE also utilizes MATLAB's method of manipulating graphics. Handles are used to make alterations to objects in a figure.

(4) **SIMULINK®** Dabney (2003); MathWorks (2013b) – SIMULINK was developed by MathWorks and is used for modeling, simulating, and analyzing multi-domain dynamic systems. SIMULINK uses interconnected graphical objects or blocks; it comes with an extensive standard blockset that is capable of most basic tasks. Specialized blocksets can also be developed for particular tasks. An example of the latter

[1] Many website references do not have an author *per se*, in which case we use the notation "Abbr-www" where 'Abbr' is the organization's abbreviation, e.g., EC = Environment Canada.

would be a blockset designed specifically to model an aircraft. The integration between MATLAB and SIMULINK is very tight, and it is possible to program SIMULINK blocks using M-code. The resulting blocks are called S-functions.

(5) **The FlightGear Flight Simulator** FG-www (2013) – The FGFS is a sophisticated open-source flight simulation package that is released under the GNU General Public License. It can be interfaced with SIMULINK via a compiled S-function, which was done in this project; furthermore, FGFS can be used with external aircraft models such as those developed using the powerful Aerosim Blockset which provides FlightGear blocks for that purpose.

(6) **The Aerosim Blockset** UD-www (2013) – The Aerosim Blockset, produced by Unmanned Dynamics, is a complete set of aeronautical simulation tools that provide the user with the ability to rapidly develop a nonlinear six-degree-of-freedom aircraft dynamic model. Aerosim includes both basic aircraft dynamics blocks and complete aircraft models. It is possible to customize these models through the use of parameter files.

Other resources used in this project include:

(1) **Mapping data** – Maps displayed in UMPEG require both raster and vector data formats. Raster data files were obtained from NASA's Blue Marble next generation website NASA-www (2013) and many vector datasets were obtained from Natural Resources Canada NRC-www (2013). A series of weather maps/charts from NAV CANADA (a private-sector civil air navigation services provider) are also readily available NavCan-www (2013).

(2) **Weather data** – The World Meteorological Organization WMO-www (2013b) promotes standardization of meteorological data and helps facilitate worldwide cooperation. The most common format to distribute regional forecast data is called GRIB or GRIdded Binary CMC-www (2013b). In particular, Environment Canada's numerical weather data is stored in a GRIB Edition 1 format GRIB-www (2013a); Environment Canada EC-www (2013a) provides its gridded numerical weather prediction models, as well as documentation (TGS-www (2013d); CMC-www (2013b); OMF-www (2013c)), for free. Software exists for reading numeric GRIB data directly into MATLAB. A function called `read-grib` Blanton (2013) has been well developed and was used in this work. An efficient binary searching routine was found in Bailey (1999); the basic bilinear interpolation used to interpolate weather data is described in Burger and Burge (2008).

3. REQUIREMENTS

This work involved the aggregation and integration of various technologies, data sets and software routines and packages to produce UMPEG, an interactive utility which ties together and processes all relevant information necessary to plan and execute a UAV mission. Several attributes and capabilities are critical for this GUI to be effective and practical:

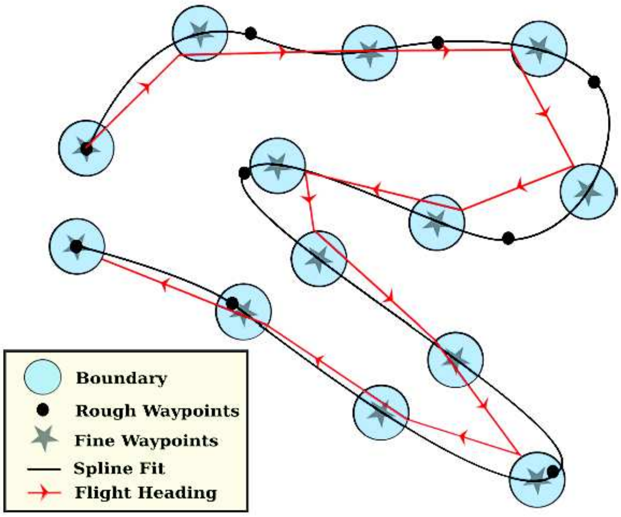


Fig. 2. Flight Path Definition and Navigation via Waypoints

(1) A variety of useful weather charts should be easily accessible for reference, e.g., forecast temperature, wind conditions and barametric pressure (the standard "weather map"), turbulence, icing, freezing levels, cloud cover, etc.

(2) The definition of a mission flight path should be simple and intuitive, in the standard three dimensional frame (latitude, longitude, altitude, hereafter **lat, lon, alt**).

(3) The modification of a flight path should be simple and intuitive, to allow iteration during planning and change of plan as the situation evolves during execution.

(4) Two automatic features of flight path modification are needed during execution: "go home", in case of an emergency situation such as an unpredicted squal or high turbulence, and "loiter", in case an item of interest is spotted that requires surveillance, such as illegal fishing or a ship in distress.

(5) The RAVEN UAV guidance and control system works by navigating from one waypoint to the next, so the GUI should receive the UAV's current position via telemetry and control it by specifying the next waypoint. The flight path is defined by "rough waypoints" by the user and fitted by splines, from which "fine waypoints" are generated for navigation, as shown in Fig. 2. The UAV switches to the next fine waypoint when it is within the boundary of the present target; note that there are many more fine waypoints than rough ones, so the difference between the ideal flightpath ("Spline Fit") and the actual one is greatly exaggerated in Fig. 2.

(6) Mission feasibility during both planning and execution must be based on weather conditions, so UMPEG must include a flight simulator that allows forecast wind speed to be an exogenous input for Planning

Mode and access to weather forecasts in real time during Execution Mode. Such forecast data should be supplied in the form of GRIB numerical weather models and processed to produce the required displays.

4. IMPLEMENTATION

Many steps and processes had to be determined, researched and implemented in order to build UMPEG. Here we restrict discussion to the more important and/or complicated issues. More information is provided in Perry (2009).

4.1 Weather Data Processing

A robust source of numerical weather data is needed to assess the flight conditions over the duration of a given mission. An ideal meteorological numeric model would cover a large area, be presented over a spatial grid, include various meteorological variables, and span a suitable time frame. After considerable research a suitable model from Environment Canada was found; it is called the low-resolution Canadian Meteorological Centre (CMC) GRIdded Binary (GRIB) database CMC-www (2013b). The low-resolution version was chosen because it was not too granular for our application and it is much easier to process and store; see Fig. 3 in which you can see the northeastern USA and Atlantic Canada, including Newfoundland (bold red boundaries), in both the lat, lon frame (faint red curves) and the GRIB data polar stereographic grid (PSG) in black.

The required interpolations to determine weather conditions along a flight path proved to be very complex. First of all, in the end we have to deal with four dimensions, lat, lon, alt and time, and we have to transform from the GRIB data grid to this frame (again, refer to Fig. 3). The GRIB data is available over a PSG, with horizontal axes denoted X and Y. This is further complicated by the fact that the CMC GRIB data is vertically stratified not by altitude but by pressure, P, as shown in Fig. 4. At each X, Y corner of the interpolation frame that encompasses a given point on the UAV trajectory we have another interpolation to determine the weather conditions at the four corners. Then we interpolate one more time to determine the conditions at that point, and transform the results back into our lat, lon, alt frame.

4.2 Simulation with Wind Conditions

The UAV model used for Planning Mode in UMPEG has six control inputs, elevator, aileron and rudder deflection, throttle setting, air/fuel ratio, ignition (on/off); and three wind velocities, northerly, easterly and downward. Environment Canada provides wind forecasts on an hourly basis. Data is extracted for the anticipated mission period, e.g., if a seven-hour mission is to start at 09:30 then forecast data for 09:00, 10:00, …16:00 and 17:00 is gathered and interpolation is used to generate wind data for the points in time and space required for the simulation.

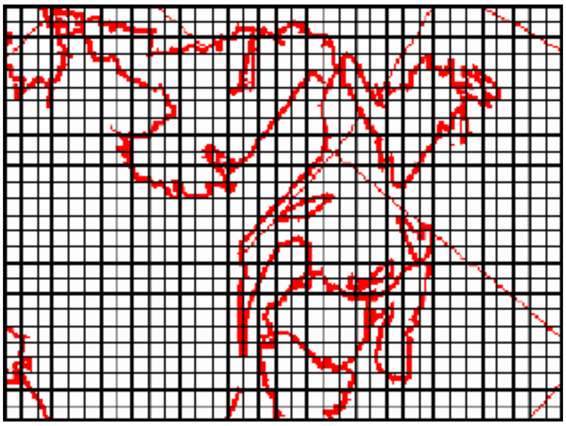


Fig. 3. Polar Stereographic Grid *vs* Lat & Lon

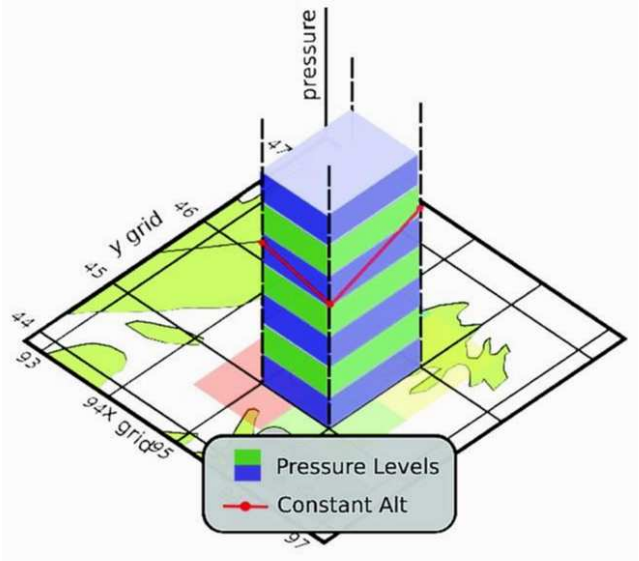


Fig. 4. Interpolation over X, Y, P

A typical Planning Mode simulation result is depicted in Fig. 5; note that the profile with wind included had to travel farther than in the case without wind. The small-scale flight anomalies shown in Fig. 5 are artifacts of an earlier RAVEN control system design; they also account for the longer distance traveled compared with the desired trajectory. It was recommended that the controller should be refined to produce smoother flight trajectories.

5. ILLUSTRATIVE EXAMPLE

It is recommended that the operator consult appropriate weather charts before planning a mission. Available charts include wind speed and direction, turbulence, upper air analysis, precipitation, surface conditions, radar imagery, and satellite imagery. A sample wind chart is portrayed in Fig. 6 as an illustration.

5.1 Planning Mode

Once weather conditions have been assessed, the operator can use UMPEG in Planning Mode, as portrayed in Fig. 7. This GUI has been designed to be clear, uncluttered and intuitive; it has only the labels, buttons and displays

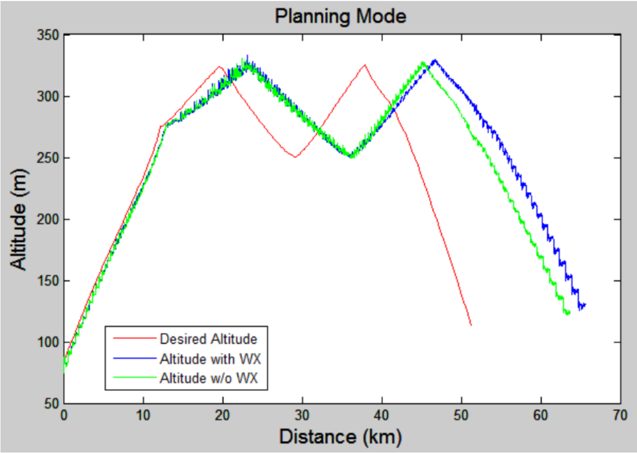


Fig. 5. Planning Simulation with and without Weather

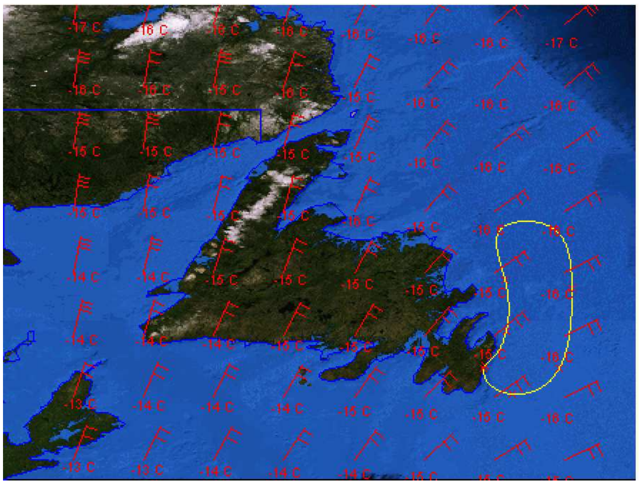
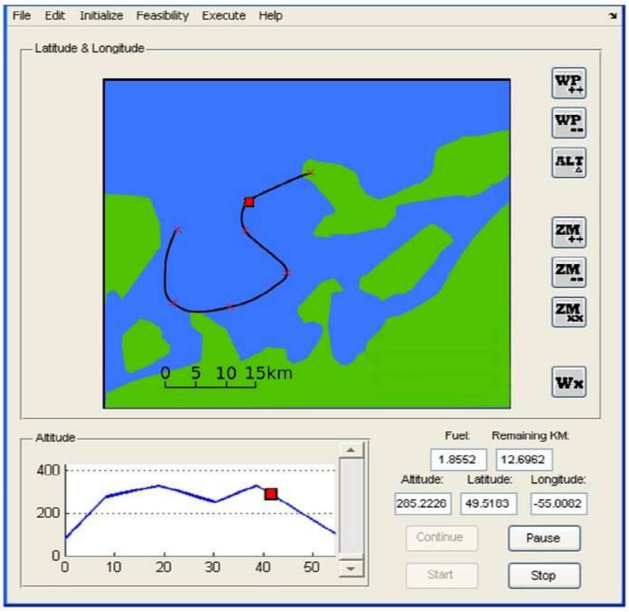


Fig. 6. Weather Map Display in UMPEG

needed for the tasks at hand, under the expectation that the operator is familiar with the system. For example, we assume that it is understood that the scale on the altitude plot is meters *vs* distance along the trajectory in kilometers. Drop-down menus are available via the top horizontal tabs, `Initialize`, `Feasibility`, `Execute` and `Help`; they also include adding/deleting waypoints (**WP** and **ALT** buttons) to modify the trajectory, zooming the map display (**ZM** buttons), and displaying current wind conditions (**Wx** button). Clicking `Initialize` produces the options `Define Trajectory`, `Display Wx Chart(s)` etc.; for further detail please refer to Perry (2009).

Once a proposed mission trajectory is entered (in this case, the flightpath covers Notre Dame Bay in northeastern Newfoundland) its feasibility can be assessed via simulation, as shown in Fig. 7[2]. The progress of the feasibility simulation is indicated by the colored squares. At the end of the run the predicted results are displayed; in this example the output was as shown in Table 1. Note that including the moderate forecast wind (20 knots or 37 km/h to the northeast, see Fig. 6) significantly increased the fuel consumption (by 18 %) and elapsed time (by 26 %) while



Fuel: 1.8552

Remaining KM: 12.6962

Altitude: 285.2226

Latitude: 49.5183

Longitude: -55.0082

Fig. 7. Implemented GUI for UMPEG (Planning)

the distance traveled only increased slightly (by 3.0 %). The additional fuel consumption and elapsed time may be sufficient to impact the probability of mission failure quite adversely; if so the planner may modify the mission trajectory easily by adding/deleting rough waypoints (red x symbols in Fig. 7) and try again.

Table 1. Planning Simulation Results

	Without Weather	**With Weather**
Fuel Consumed	0.2038 kg	0.2414 kg
Elapsed Time	6251 s	7852 s
Distance Travelled	63.68 km	65.60 km

5.2 Execution Mode

The operator controls the UAV in real time using UMPEG, which allows the operator to revise the flight trajectory in mid-flight. In this example the UAV is commanded to loiter near an event of interest (`Hold`), and later to directly return to base (`Go Home`); Fig 8 shows the GUI lat-lon display for this mission. In testing this mode, the “UAV” being controlled was, for reasons beyond our control, only the planning model.

Executing a mission can be started by selecting the `Execute Mission` item in the `Execute` menu. A `Start` button appears which can be used to begin the mission. Real time changes are accomplished by clicking either the `Change Route`, `Hold`, or `Go Home` items (not shown here). This part of the project was completed as a proof of concept and will require significant future development for use with an actual UAV. This development will entail interfacing the UAV’s hardware and software with UMPEG via a wireless link.

[2] Simplified maps are usually displayed during planning and execution, to reduce run time.

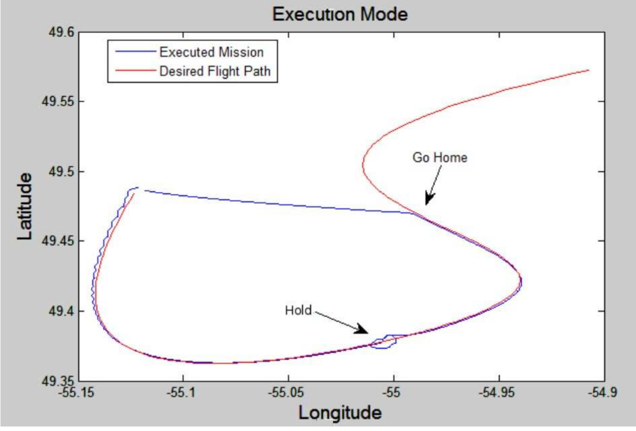


Fig. 8. GUI Map Display for Execution

6. SUMMARY AND CONCLUSION

The main contribution of this paper is the unique synthesis of technologies, information and software used to construct a prototype software package for UAV mission planning and execution. It was designed for operations in areas with harsh climate conditions, especially wind, although other factors such as turbulance and icing conditions are viewable in the charts and could be added to the feasibility simulations later. Mission feasibility planning can be evaluated iteratively and, when finalized, seamlessly rolled into an operational flight.

This GUI is proposed as a complete solution for deploying unmanned aerial vehicles. Relevant data from a mission is archivable for easy confirmation of conditions after the fact or for reproducing the same mission specification for a later feasibility study of the same scenario. A majority of the mission feasibility features are enabled by using integrated nationwide numerical weather model forecast data. A complete weather monitoring solution is provided by making overlays and graphical charts easily accessible to the operator. Customizable mapping allows the operator to display additional relevant graphical information such as hydrology, transportation routes, and populated areas. Control during real-time flights is facilitated though UMPEG, as operators have the ability to modify flight routes on the fly to react to unforeseen circumstances.

These concepts and developments can be applied to autonomous land, marine or undersea vehicles as well. For example, terrain descriptions, surface sea condition forecasts or undersea current forecasts could be used in feasibility studies and to aid mission execution. Of course the vehicle model would have to be changed accordingly.

UMPEG is decidedly still in the developmental/prototype stage. The primary need is for a substantial reduction in computer run time during feasibility studies. This could be accomplished by reimplementing the UAV simulation model and the S-function that performs weather interpolation in C code. It is also incomplete in that it cannot handle take-off and landing modes of operation – these would have to be done manually in the current version of UMPEG.

REFERENCES

Bailey, D.A. (1999). *Java Structures: Data Structures in Java for the Principled Programmer.* McGraw-Hill.

Blanton, B. (2013). *GRIB Edition 1 Reader for MATLAB* `https://github.com/BrianOBlanton/read_grib`.

Burger, W. and Burge, M.J. (2008). *Digital Image Processing: An Algorithmic Introduction Using Java.* Springer.

Env't Canada (2013a). Environment Canada Web Site[3]. `http://www.ec.gc.ca/`.

Env't Canada (2013b). *Numerical Data - GRIB Format* `http://weather.gc.ca/grib/usage_tips_e.html`.

Env't Canada (2013c). *Operational Model Forecasts.* `http://weather.gc.ca/model_forecast/index_e.html`.

Env't Canada (2013d). *Technical Grid Specifications.* `http://weather.gc.ca/grib/CMC_GRIB_Technical_Grid_Specifications_e.html#low_res_reg`.

Environmental Systems Research Institute. (1998). *ESRI Shapefile Technical Description: An ESRI White Paper.* `www.esri.com/library/whitepapers/pdfs/shapefile.pdf`.

O'Young, S. and Hubbard, P. (2007). RAVEN: A maritime surveillance project using small UAV. *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on.*

NASA (2013). *Blue Marble next generation.* `http://earthobservatory.nasa.gov/Features/BlueMarble/`.

Natural Resources Canada (2013). *GeoGratis, Download Directory, Vector Data.* `http://geogratis.gc.ca/geogratis/search`.

Nav Canada (2013). *Aviation Weather Web Site, Forecasts and Observations.* `http://www.flightplanning.navcanada.ca/cgi-bin/CreePage.pl&Page=forecast-observation&TypeDoc=html`.

Perry, S. (2009). *Operator interface for autonomous unmanned aircraft mission feasibility and control.* MScEng Thesis, University of New Brunswick.

The FlightGear Project (2013). *FlightGear Multi-platform Flight Simulator.* `http://www.flightgear.org`.

The MathWorks (2013a). *MATLAB - The Language of Technical Computing.* `http://www.mathworks.com/products/matlab/`.

Hahn, B. and Valentine, D. *Essential Matlab for Engineers and Scientists.* Academic Press.

The MathWorks (2013b). *SIMULINK* - Simulation and Model-Based Design. `http://www.mathworks.com/products/simulink/`.

Dabney, J. *Mastering SIMULINK.* Prentice Hall.

Unmanned Dynamics, LLC (2013). *AeroSim, Aeronautical Simulation Blockset, Version 1.2, User's Guide.* `http://www.u-dynamics.com/aerosim_ug.pdf`.

World Meteorological Organization (2013a). *A Guide to the Code Form FM 92-IX Ext. GRIB Edition 1.* `http://www.wmo.ch/pages/prog/www/WDM/Guides/Guide-binary-2.html`.

World Meteorological Organization (2013b). *WMO Web Site.* `http://www.wmo.int/pages/index_en.html`.

[3] All website URLs were confirmed at the time of writing.