DEVELOPMENT OF A GENERIC FLIGHT TEST MANEUVER INJECTION MODULE

Christoph Krause[*] , Christoph Goettlicher[*] , Florian Holzapfel[*]
[*]Institute of Flight System Dynamics, Technical University of Munich, Garching, 85748, Germany

Keywords: *Flight Test, Maneuver Injection, System Identification, Statemachine, MATLAB Simulink*

Abstract

At the Institute of Flight System Dynamics of the Technical University Munich, various unmanned and optionally piloted aerial vehicles are being used to demonstrate novel flight control and flight management functions.

In this context, flight tests are not only being conducted to prove functionality, but also to verify flight dynamic models, or to evaluate controller performance. Additional ground tests may be used to identify actuator dynamics.

Those tests consist of special maneuvers, e.g. doublets, sweeps or multi-sine signals, which are fed directly to the control loops or to the actuators. Some of those tests could be conducted by a pilot, but the necessary accuracy and repeatability cannot be guaranteed. More complex signals, like phase optimized multi-sines, cannot be performed manually at all.

To overcome those problems a software module, executed directly on the flight control computer, was developed. It is able to inject various flight test maneuvers into different control loops or directly to the actuators. This paper presents the development of this generic Flight Test Maneuver Injection module (FTMI).

1 Introduction

To test innovative flight control and flight management concepts, Optionally Piloted Vehicles (OPVs) and Unammed Aerial Vehicles (UAVs) are increasingly used ([1] and [2]).

The Institute of Flight System Dynamics of the Technical University Munich is developing flight control software for different OPVs and UAVs. In order to prove the real-world operational feasibility of the software, functional flight tests are being performed.

Additional maneuver flight tests, might be necessary for various reasons. For the development of the control loops a model-based approach is being used [3].

Therefore the quality of the flight dynamics model is a key requirement. With maneuvers involving multi-step or multi-sine control surface commands, the characteristic dynamics of the aircraft can be excited, and differences between the model and the real aircraft can be analyzed and possibly reduced.

If actuator position measurements are available, the same approach may be used to identify the actuator's dynamic behavior from command to actual position.

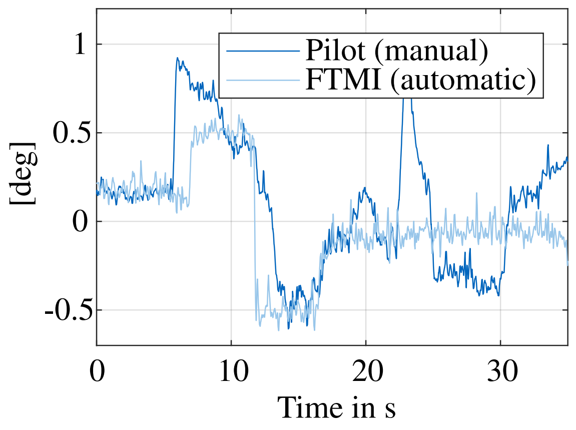


Fig. 1 Elevator Doublet Comparison

Furthermore, for a complete evaluation of the controller, the cascaded control loops are tested one by one, starting from the innermost level. In these cases the generated signals are used as a command for the basic controller or the autopilot, and the respective characteristics such as tracking performance or overshoots may be analyzed based on a clean data-set with perfectly executed excitation signals.

To get valid results, maneuvers often need to be repeated with high accuracy. To illustrate the shortcomings of manual execution, figure 1 shows flight test data from one of the platforms used at the institute. Both curves are elevator surface deflections, measured during a flight test. The high frequency noise originates in the sensor, which is used to measure the deflection and can be neglected during this discussion.

The light blue shows the elevator response to a doublet commanded by the Flight Test Maneuver Injection (FTMI). The dark blue line shows the same maneuver flown manually by the test pilot. It can be seen, that the pilot overshoots the targeted amplitude in the first half of the doublet and the transition to the second half is much slower than with the automatic maneuver, thus illustrating a more accurate execution of the latter.

It can be seen, that the pilot overshoots the targeted amplitude in the first half of the doublet and the transition to the second half is much slower than with the automatic maneuver, thus illustrating the more accurate execution of the latter. Additionally, the elevator is moved during the second half of the maneuver, where it was supposed to be steady.

In the beginning an introduction of the system architecture will clarify the role of the injection module within the FCC software. Following this general overview, a more detailed description of the FTMI is given. Additionally, the features of the module are introduced, before example flight test data proves the real-world applicability.

2 System Architecture

The system architecture of the flight control computer (FCC) includes various modules and numerous connections between them. This section gives an introduction to a simplified system architecture, which only includes the modules and connections relevant for the FTMI. An overview of this architecture is shown in figure 2. The FTMI, which is an optional part of the FCC software, is shown with dashed lines, while the nominal modules are shown with solid lines.

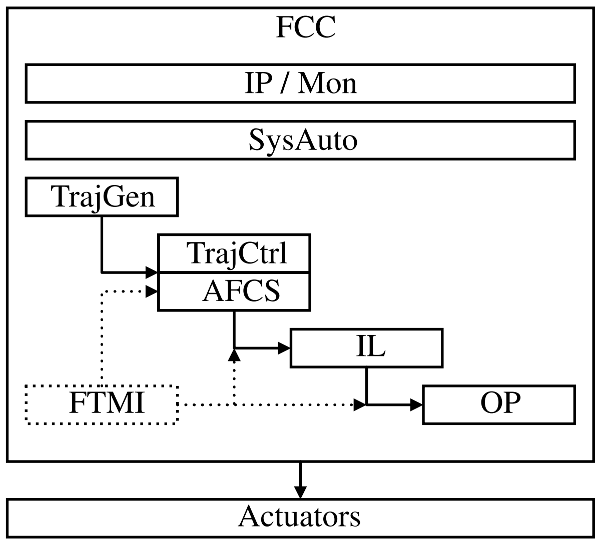


Fig. 2 Simplified Software Architecture of the FCC

The FCC receives, among others, commands from the flight operator, data from the navigation system and status information from other on board systems. This information is combined and checked for integrity within the Input Processing and Monitoring module (IP / Mon). The System Automation (SysAuto), an administrative module, distributes this data and activates or deactivates the lower modules. Depending on the configuration other modules are also part of the FCC software.

To enable GNSS based way point flying capabilities, the Trajectory Generation (TrajGen) module, computes an online 3D flight path [4], based on provided way point lists. Further, it calculates the offset of the aircraft's position to that trajectory. The active waypoint list can be selected from a number of predefined flight plans, which can be adapted to different missions. Those lists generally include en route, takeoff, landing and link loss lists.

If flying according to those GNSS-based flight plans, the offset of the aircrafts position is forwarded to the Trajectory Control (TrajCtrl) module. This module calculates load factor commands in order to reduce the offsets, which are inherited by the next module [5].

Additionally, the aircraft can be controlled by the flight operator using conventional autopilot commands like heading, speed and altitude. In this mode the necessary load factor commands are computed by the Auto Flight Control System (AFCS) [6]. This module also includes an auto thrust system which is used for velocity control during all operational modes.

The Inner Loop (IL) inherits the load factor commands from either TrajCtrl or AFCS and calculates surface deflection commands in order to achieve the desired load factors. Those commands are used by the platform specific Output Processing (OP) to calculate actuator position commands, which are then forwarded to the respective actuators.

The Flight Test Maneuver Injection module is connected to the Auto Flight Control System, the Inner Loop and the Output Processing. Additionally, it receives commands from the flight operator and reports status information to the ground station via the administrative modules. The interfaces of the FTMI to the three modules include replacement signals for the respective module and activation information for the different axis.

3 Module Architecture and Interfaces

This section describes the architecture of the FTMI module and its interfaces.

The Flight Test Maneuver Injection is separated into two parts, the master and slave module. Those two modules contain the five functional blocks of the FTMI as illustrated in figure 3.

The main state machine and data handling is included in the master module. The maneuver generation, state execution and protections are computed within the slave module.

The interfaces include internal/on board connections to the airplane as well as external connections to the flight operator or the ground station. The interface is designed as bidirectional generic interfaces to support further development and integration into other platforms.

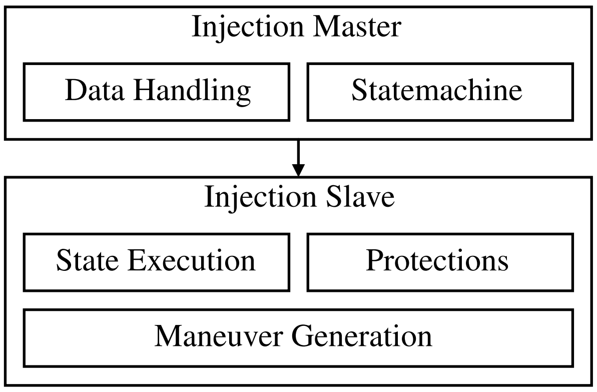


Fig. 3 Module Architecture

The on board inputs are sensor data from the navigation system of the aircraft, which are, among others, used for the evaluation of the trim point. Additionally, the maneuver definitions can be considered inputs, even though they are not send to the module but are rather included in the FTMI itself. The outputs include commands to the different control loops (dashed lines in figure 2).

The basic external interface from the flight operator consists of an "activate_flg" and a "flight_test_number". The same variables are used as report from the FTMI to the ground station. Additionally, waiting for the trim point or even automatic trim point capturing can be activated or deactivated by the flight operator. Furthermore, the protection methods can be enabled or disabled using the interface.

The Data Handling function block processes the data defining the flight test maneuvers. This is done using five compact but expandable data structures. The data stored to define the maneuvers include information about the maneuver index, control surface configuration, trim point values, variables for the maneuver itself and possible protection methods.

Further functionality of the FTMI will are described based on the six states of the main state machine. Those are:

- `standby`,
- `wait auto trim point`,
- `wait trim point`,
- `execute maneuver`,
- `wait` and
- `index error`.

While in `standby` the flight control computer is running, but the FTMI is not active. That can happen during manual flight as well as during automatic flight by the FCC. Each maneuver has a specific trim point consisting of a subset of the user-defined altitude, speed and heading values. If the maneuver is activated and at least one of the trim point parameters is activated as well, the maneuver injection will wait until the given value is reached, which has to be achieved by the pilot. If additionally an automatic trim point flag is set, the maneuver injection will utilize the auto flight control system to automatically reach the desired trim point values. During that time either the `wait auto trim point` (automatic trim point capture) or `wait trim point` mode will be active. If none of the trim point values are activated these states are skipped, and the maneuver may be started directly.

After the trim point phase the actual maneuver is executed (see section 4.2 for available signal shapes) after which the previous flight mode is automatically restored. During that time the `wait` mode is active. The flight operator has to acknowledge the end of the flight test by deactivating the "activate_flg". This will be reported by the mode change back to `standby`. If during a maneuver the "activate_flg" is cleared or the "flight_test_number" is changed, the maneuver will be aborted immediately leading to `wait` or `standby` depending on the "activate_flg". The mode `index error` is active if a non existing test case is selected. This will remain active until a valid test case number has been selected.

The slave module is connected via a slim interface only containing information about the currently selected flight test maneuver and status information of the state machine.

This information is used by the State Execution block which controls the data flow within the slave module depending on the active state of the main state machine within the master module.

Protections are available within the slave module to keep the airplane close to the trim point during maneuver execution. These are necessary, since many of the subsequent analysis steps assume linearity of the plant, which could be violated if the aircraft drifted too far from the trim point.

The computation of excitation signals is done within the Maneuver Generation functional block. The available maneuvers are described in detail in section 4.2.

Even though the software is running on only one computer in the current configuration, the modules are separated to support future separation to two computers. The functional blocks within those two models have different requirements concerning storage space and fast real time execution capability. The blocks in the master module, especially the data handling, need storage space for the maneuver definitions but do not necessarily need to run with the same high frequency as the main flight control algorithms. The blocks within the slave module must run with the same frequency as the main flight control loops but do not need much storage space. The interface between both modules is designed to support asynchronous execution of the master and slave module on two separate computers.

4 Features

4.1 Injection Points

As described in the previous section the maneuvers, generated by the FTMI, can be injected on different layers of the controller, represented by the switch blocks in figure 2. For every injection point, it can be decided, if the generated signal form is to be added incrementally to the trim value at the beginning of the maneuver, or if it is to replace the signal completely.

On the lowest level they can be used directly as actuator commands, for e.g. flight testing for bare-airframe system identification. The maneuvers are defined as surface deflection, where the platform-dependent matching to the respective actuator command is performed in the output processing. Here, it often makes sense to treat the signals as increments on the trim-values, since otherwise the aircraft may drift too far away from the trim point, which is in general undesired.

To investigate closed loop behavior of the controller, the maneuvers can also be fed to the Inner Loop. In this case the signals contain commands for the vertical and lateral load factor (fz, fy) or bank angle (phi). Thus clean data to assess tracking performance, overshoot, settling time, etc. of the controller can be generated.

The commands at this level can also be used as auxiliary control functions when injecting signals on a different level: e.g. a bank angle command of 0° effectively implements a wing leveler, which can for example be used, when investigating the phugoid motion of the aircraft. The phugoid maneuvers are comparatively long, and without control activity the aircraft will build up a considerable bank angle due to imperfect initial conditions and external disturbances, which can be avoided by using the wing-leveler capability. Again, this is necessary in order to keep the aircraft close to the desired trim point to be able to consider linearized models.

On the next level, signals can be injected as commands for the AFCS. This level is currently only used for features like automatic trim point capture or post maneuver continued flight and not for maneuver injection itself. Additionally, it can be used for altitude hold or heading hold during maneuvers with only one active test axis.

4.2 Maneuvers

Currently the FTMI module supports five different maneuvers, where special focus was laid on a highly modular approach. The interface to the FTMI module for execution and parametrization is the same for all maneuvers, thus allowing for quick and easy extension.

Most of the maneuvers, that have been implemented so far are mainly aimed at exciting the aircraft characteristic motion for parameter estimation and controller assessment purposes.

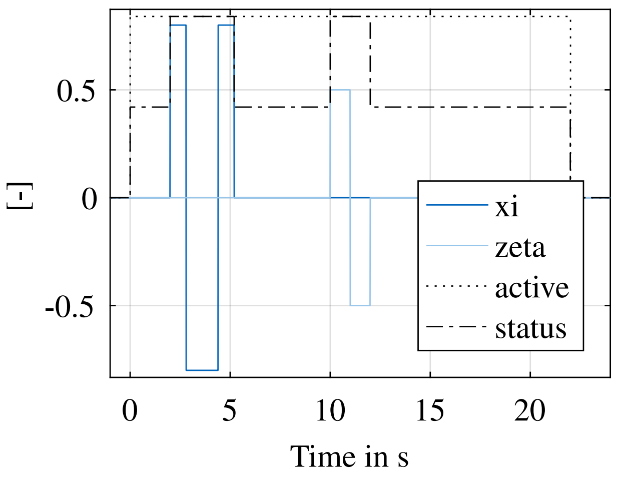


Fig. 4 Multi-Step Maneuver

However, they may also be used for automatic flight tests targeting model validation, or certification tests.

A classical approach to collect data for parameter estimation is to excite the system using step inputs with varying step times [7, 8]. Common examples for this are the 3-2-1-1, 2-1-1, or doublet maneuver, where the names indicate the relation of the step times to some base step width. Figure 4.2 shows an example for a 1-2-1 together with a time-skewed doublet in the lateral plane, which were designed using the FTMI and could e.g. be used to excite the roll mode and dutch roll motion. In addition to the signal shapes, figure 4.2 also shows values of the “active” flag command as received by the maneuver generation module and its internal “status”. The latter indicates wait-times with zero output, as well as active signal generation, which can be used to extract certain segments during post-processing of the data. The main parameters to be adjusted for multi-step maneuvers are the base step time, step sequence, and amplitude.

In some applications it may make sense to vary an input signal linearly, as is illustrated in figure 4.2 for the two thrust levers. If other controller loops are used to keep the altitude constant and wings level during this increase, it provides an effective means of implementing a level acceleration maneuver to investigate drag and lift

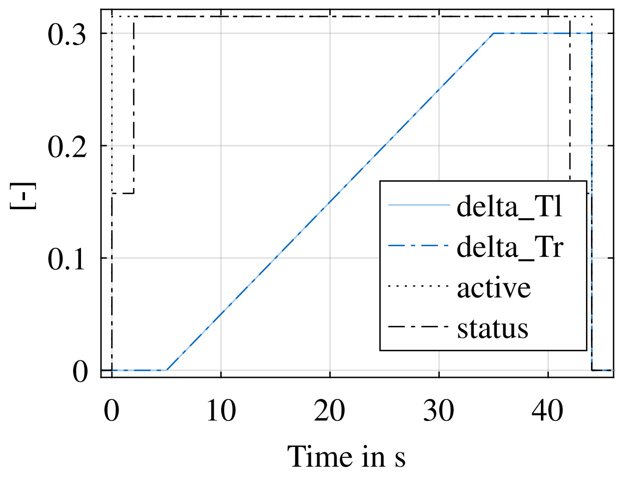


Fig. 5 Ramp Maneuver

characteristics over a range of velocities.

For both of the aforementioned maneuvers a repetition count can be used, to generate highly customizable saw-tooth or rectangular wave sequences.

Maneuvers that have been successfully applied in frequency domain parameter estimation are part of the FTMI module as well. An approach of using phase-optimized multi-sine signals as illustrated by Morelli et al. [9] is incorporated in the FTMI module. A multi-sine signal at one sampling time t_k for the j-th signal is

$$_j u_k = A_j \cdot \sum_{i \in I_j} \sin\left(\omega_i \cdot t_k + \phi_k\right). \tag{1}$$

Here, the frequency range $\left[\omega_1, \omega_{n_f}\right]$, number of excited harmonics n_f and amplitude per output A_j can be chosen freely. The mapping of frequencies to signals is defined by the user via the set of active frequency indices per output: e.g. $I_2 = \{1,3,5,7\}$ maps the frequencies ω_1, ω_3, ω_5, and ω_7 to the second signal. The module then optimizes the phase angles in order to minimize the relative peak factor [9]

$$RPF_j = \frac{\max_k\left({_j u_k}\right) - \min_k\left({_j u_k}\right)}{2\sqrt{\frac{2}{N}\sum_{k=1}^{N} {_j u_k}^2}}. \tag{2}$$

In this way, large peaks due to interference are avoided, while the desired frequency spectrum is kept constant. Figures 6(a) and 6(b) show an example time-series and power spectral density (PSD). This example maps frequencies from $0.5Hz$ to $2.6Hz$ to aileron and rudder controls, thus exciting the complete lateral dynamics simultaneously, but with perfectly uncorrelated signals [9].

Another approach for exciting a broad range of frequencies is via a sine-wave with exponentially increasing frequency [10]

$$\omega(t) = \omega_1 + \frac{\exp\left(\frac{C \cdot t}{t_{end}}\right) - 1}{\exp(C) - 1}\left(\omega_{n_f} - \omega_1\right) \tag{3}$$

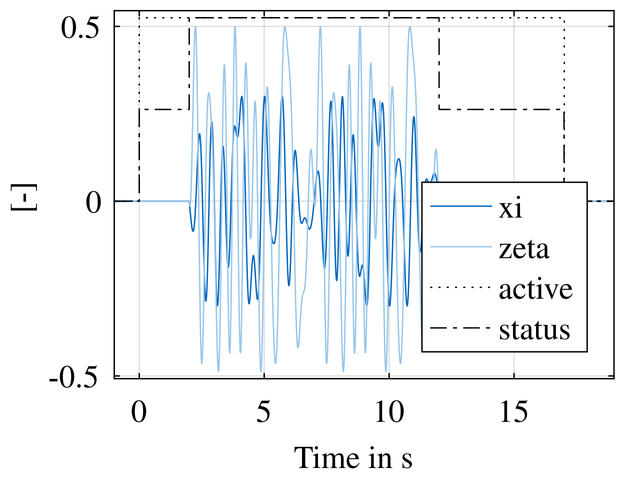
$$u_k = A \cdot \sin\left(\int_0^{t_k} \omega(\tau)\, d\tau\right) \tag{4}$$

where the constant C can be used to adjust the steepness of the frequency increase, for which Tischler proposes a value of $C = 4$. Sweeps are only implemented for one signal at a time, which is why the prescript j is dropped here. Configurable parameters include frequency range $\left[\omega_1, \omega_{n_f}\right]$, duration t_{end}, and amplitude A. Figure 7(a) shows an example time-series in the elevator control input, with the corresponding PSD in figure 7(b).

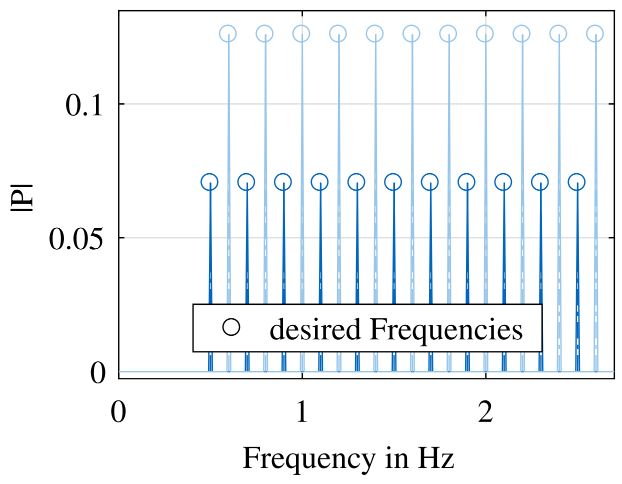
The last maneuver, currently implemented is a generic spline maneuver. By providing knots, spline coefficients and the spline order, the user is able to create arbitrary signal shapes, if something more complex than the aforementioned signals is desired. This spline maneuver can also be used as interface for other maneuver definitions, e.g. signals that were specifically designed to maximize the information content in recorded data: in [11, 12] Morelli illustrates an approach to maximize information using dynamic programming, which leads to input signals that can only attain discrete values. These can perfectly be represented using zero-order splines.

4.3 Trimpoint Detection and Capture

Each maneuver has a user defined trim point consisting of heading, speed and altitude, which can be individually activated. Additionally, values close to zero are required for the body rotational rates.

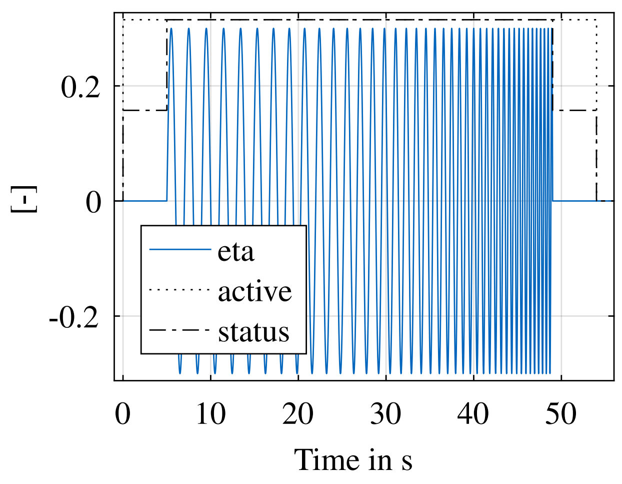


(a) multi-sine maneuver time series

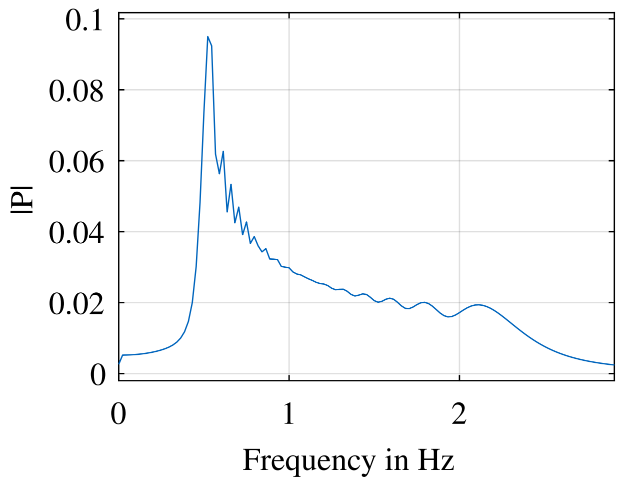


(b) corresponding PSD

Fig. 6 example multi-sine maneuver



(a) sweep maneuver time series



(b) corresponding PSD

Fig. 7 Example sweep maneuver

The Flight Test Maneuver Injection module is able to detect this trim point. If activated, it starts the execution of the maneuver only when the specific trim point is reached.

Furthermore it can utilize the AFCS to automatically capture the defined trim point. In this case the maneuver starts with the capturing of the new trim point. If this point is reached, the maneuver is automatically executed and afterwards the trim point is captured again.

4.4 Offline Design, Online Execution and Adjustment

One full maneuver definition consists of the corresponding signal type, its parameters, a trimpoint, the injection point, as well as the automatic trim-point capture, trim point detection, and protection settings.

To facilitate the definition of a full flight test program consisting of several signal shapes, injected at various points of the controller at different trim points, a graphical user interface has been developed. It allows for easy parameterization of the desired signals by providing graphical representations in time and frequency domain, similar to figures 6(a) and 6(b). Furthermore, injection, protection and trim points may be defined.

The resulting maneuver list is then stored directly within the FCC, which makes it possible to select individual maneuvers purely based on their list-index. This drastically reduces communication overhead.

Above approach limits the possibility of online adjusting signal shapes, which is alleviated by a set of generic scaling signals. Depending on the maneuver in question, these may be used to modify selected parameters online.

5 Conclusion and Outlook

This paper presents the development of the generic Flight Test Maneuver Injection software module. It allows the injection of various maneuvers into different control loops or directly to the actuators. Those maneuvers are used to identify or improve flight dynamic models, identify actuator dynamics or to evaluate controller performance. The implemented FTMI is used in multiple projects at the Institute of Flight System Dynamics and has already contributed to several improvements.

After a description of the system architecture of the FCC the architecture and interfaces of the FTMI are introduced. Following this insight to software architecture the features are described. This most importantly includes the available maneuvers, but also the injection points, trim point detection and capture and others. The paper is concluded by this section with flight test data and conclusion.

To demonstrate real-world applicability a 0.7 degree elevator doublet is shown in figure 8. In the upper part of the figure the command from the FCC, the actuator position, the surface position and the status of the respective electromagnetic clutch is presented. It can be seen that after activating the maneuver the current surface deflection (eta trim) is held for 5s. After that the 0.7 degree doublet is commanded, which is added incrementally to the trim deflection. The maneuver ends with a trim eta command for 20s.

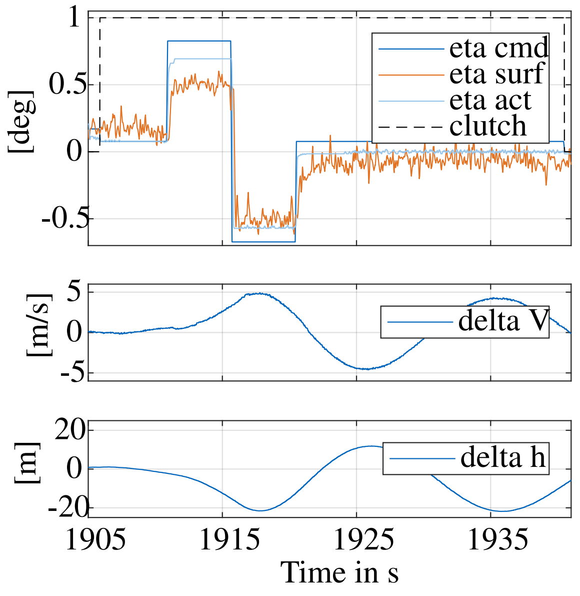


Fig. 8 Automatic Elevator Doublet

The signal was designed to stimulate the phugoid motion of the airplane, which can be seen in the lower part of the figure in deviations from trim speed (delta V) and altitude (delta h).

This paper gives a detailed description of the system architecture of the FTMI module, its components and interfaces, as well as its integration within the overall FCC. Additionally, the implemented features, such as various injection points, different maneuver signal shapes, and automatic or manual trim-point capture were discussed, together with an illustration of the work-flow necessary to design a complete flight test. Furthermore, the protections and use of automatic control functions to keep the aircraft close to the desired trim-point were shown. Lastly, example flight test results are presented to underline the real-life applicability of the system.

The combination of the illustrated maneuver suite, together with control functions and automatic trim-point capture allows for a wide variety of possible combinations: simple feed-forward signal augmentation such as the illustrated multisteps can be designed using the same toolchain as more complicated maneuvers like a Level-Turn or Level-Acceleration maneuver.

6 Acknowledgments

The presented work is partially funded by the German Federal Ministry for Economic Affairs and Energy on the basis of a decision by the German Bundestag.

Supported by:

Federal Ministry for Economic Affairs and Energy

on the basis of a decision by the German Bundestag

It is part of the project "TASC" which is administered by the VDI (16KN056520).

References

[1] Undertaking, S. J., “Demonstrating RPAS integration in the European aviation system,” Tech. Rep. 978-92-9216-068-5, SESAR Joint Undertaking, 2016, 2016.

[2] Kolberg, A., “German-Qatari joint development programme presents ground-breaking OPV,” "Website", 2016, "Safety & Security International (S&SI) | 2/2016".

[3] Hochstrasser, M., “Aspects of a Consistent Modeling Environment for DO-331 Design Model Development of Flight Control Algorithms,” *CEAS EuroGNC Conference*, in Press.

[4] Schneider, V., Mumm, N., and Holzapfel, F., “Trajectory generation for an integrated mission management system,” *Aerospace Electronics and Remote Sensing Technology (ICARES), 2014 IEEE International*, IEEE, 2015.

[5] Schatz, S. P. and Holzapfel, F., “Modular trajectory / path following controller using nonlinear error dynamics,” *Aerospace Electronics and Remote Sensing Technology (ICARES), 2014 IEEE International*, IEEE, 2014, pp. 157–163.

[6] Karlsson, E., Schatz, S. P., Holzapfel, F., Baier, T., Dörhöfer, C., Gabrys, A. C., Hochstrasser, M., Krause, C., Lauffs, P. J., Mumm, N. C., Nürnberger, K., Peter, L., Schneider, V., Spiegel, P., Steinert, L., and Zollitsch, A. W., “Development of an Automatic Flight Path Controller for a DA42 General Aviation Aircraft,” *CEAS EuroGNC*, 2017.

[7] Klein, V. and Morelli, E. A., *Aircraft system identification: Theory and practice*, AIAA education series, American Institute of Aeronautics and Astronautics, Reston, VA, 2006.

[8] Jategaonkar, R. V., *Flight vehicle system identification: A time domain methodology*, Vol. v. 216 of *Progress in astronautics and aeronautics*, American Institute of Aeronautics and Astronautics, Reston, Va., 2006.

[9] Morelli, E. A., “Real-Time Aerodynamic Parameter Estimation Without Air Flow Angle Measurements,” *Journal of Aircraft*, Vol. 49, No. 4, 2012, pp. 1064–1074.

[10] Tischler, M. B. and Remple, R. K., *Aircraft and rotorcraft system identification: Engineering methods with flight test examples*, AIAA education series, American Institute of Aeronautics and Astronautics, Reston, VA, 2nd ed., 2012.

[11] Morelli, E. A., *Practical Input Optimization for Aircraft Parameter Estimation Experiments*, Sc.d. dissertation, The George Washington University, Hampton, Virginia, 1990.

[12] Morelli, E. A. and Klein, V., “Optimal Input Design for Aircraft Parameter Estimation Using Dynamic Programming Principles,” *17th Atmospheric Flight Mechanics Conference*, American Institute of Aeronautics and Astronautics, Reston, Virigina, 1990.

7 Contact Author Email Address

mailto: christoph.krause@tum.de
mailto: christoph.goettlicher@tum.de

Copyright Statement