

The FlightControl library for aircraft control design applications

Reiko Müller¹

¹DLR Institute of Flight Systems, Oberpfaffenhofen, Germany
reiko.mueller@dlr.de

Abstract

Controller development of aerial vehicles is a long-standing task during aircraft design and has a large impact on the resulting systems performance. In today's integrated design loops, ideally all components of the aircraft must be considered in simulation and testing in order to develop complex system architectures and meet performance requirements. One of the tools that are suited for modeling and simulation of a multidisciplinary aircraft and systems assembly is DLR's FLIGHTDYNAMICS Library, which takes advantage of the capabilities of the MODELICA modeling language. In this work, a new augmentation to the library is discussed, which implements a range of common aircraft control concepts which can be used for design of new controllers, closed-loop simulation and experimental testing via code-export. The library is set up in a modular way, so that flight guidance and flight control systems can be developed for multiple aerial platforms, including manned/unmanned fixed- and rotary-wing aircraft. For this paper, a simulation example is provided by means of an autoland controller that shall be designed for a high-fidelity 6-DoF fixed-wing aircraft model in MODELICA. The combination of aircraft and controller models is subjected to a three step synthesis process, which yields a controller that is robust against external and internal disturbances.

Keywords: *Flight control, flight dynamics, design optimization*

1 Introduction

As was laid out in previous publications (Looye 2008; Looye et al. 2014), the DLR FLIGHTDYNAMICS library is a well established tool for aircraft system design and simulation in the language MODELICA. It can be employed in a multitude of applications, spanning from systems design / Multidisciplinary Design Optimization (MDO) (Liersch and Hepperle 2011), trajectory optimization (Müller 2021), mission simulation as well as reliability and validation studies. Many of these setups require some sort of flight controller and / or flight guidance functions in order to adhere to the respective simulation scenarios. Therefore, an effort was made to bring flight controllers and flight guidance functionality to the library / to MODELICA, where previous approaches relied upon model export (i.e., via Functional Mockup Interface (FMI)) and usage of other tools for the controller design. Hence, this paper

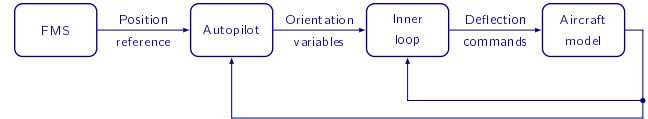


Figure 2. Standard cascaded flight controller layout: with flight guidance/trajectory generation (FMS) → Trajectory tracking (autopilot) → Tracking of aircraft orientation (Inner loop control).

exemplifies the usage of the flight controller sub-library to relevant aerospace applications.

In Section 2, the overall structure of the sub-library is laid out. Section 3 covers the flight guidance functions (e.g. Flight Management System (FMS)). The actual flight controllers are discussed in Section 4, separated by the respective control task and design methodology. The application example covers the robust design of an autoland controller and is laid out in Section 5.

2 Library structure

The FLIGHTCONTROL library was structured according to the classical perception of fixed-wing aircraft flight control where the control variables are associated with cascaded layers of tasks. This is possible due to the time-scale separation principle: the inner layer has the fastest dynamics, which allows to assume constant command inputs from the outer layer, where in turn, from the outer layer perspective, the reaction of the inner layer to commands are assumed to be instantaneous, see Figure 2. This separation is reflected in the library with subpackages for guidance (FMS), autopilot (shown in Figure 1) and inner loop controllers. Another distinction is made between classical flight control concepts relying on linear PID control and approaches containing more model information (like Nonlinear Dynamic Inversion (NDI) controllers) which are both collected in separate subpackages. The examples shown in the paper are of the first type, as they are better explainable with regards to flight physics as some of the more elaborated counterparts.

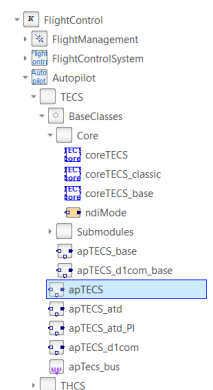
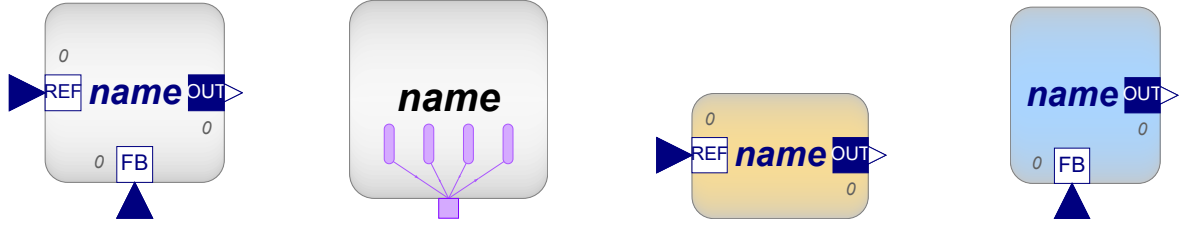


Figure 1. Opened autopilot package.



(a) The partial controller class. (b) The partial controller class using expandable bus inputs/outputs. (c) Partial guidance base class. (d) The partial computation base class.

Figure 3. Partial models that are used to discern between blocks with/without reference/feedback inputs, all blocks have an optional boolean input for integrator reset (not shown).

2.1 Interconnection with other libraries

As mentioned before, the FLIGHTCONTROL library lends itself to be combined with the DLR FLIGHTDYNAMICS library for easy integration with aircraft models or derivation of model-based controllers. With a clear I/O concept based on standard connectors and buses and a variable naming based on common standards (e.g. LN 9300, ISO 1151-1), it is also easy to integrate the flight controller library with other MODELICA based libraries for aircraft simulation. The base class models for controller blocks are shown in Figure 3. In addition, usage in form of exported models is possible, for example as Functional Mockup Unit (FMU) that can nowadays be integrated in a multitude of tools and languages (for example MATLAB®/SIMULINK®, PYTHON, etc.).

3 Guidance functions

The guidance module or often called FMS provides reference values for the downstream Flight Control System (FCS). While not being dependent on feedback in the sense of a controller, the FMS still needs information from the aircraft, for example to calculate a trajectory based on the current aircraft state or to provide switching of controller / autopilot modes. This is also reflected in the library, where models exist for 4-D continuous trajectory generation, and also a mode-based FMS relying on a state machine formulation with MODELICA STATEGRAPH2. Apart from this, the guidance focuses on the system kinematics, which are model - independent. For example the reference velocity $\vec{V}_k$ can be calculated from a trajectory given in ellipsoidal coordinates (φ, λ, h) by:

$$\vec{V}_k = [v_N, v_E, v_D]^T = \begin{bmatrix} (R_n + h) \cdot \dot{\varphi} \\ (R_e + h) \cdot \dot{\lambda} \cos \varphi \\ -\dot{h} \end{bmatrix}, \quad (1)$$

with R_e, R_n as Earth's radii of curvature. From this the usual reference variables V_k (total inertial speed), γ_k (trajectory pitch angle) and χ_k (course angle) can be calculated. If there is feedback from the aircraft, the inputs to e.g., 4-D trajectory tracking can also be provided, which would be the Along Track Distance (ATD) (for controlling time), as well as Cross-Track Distance (XTD) and

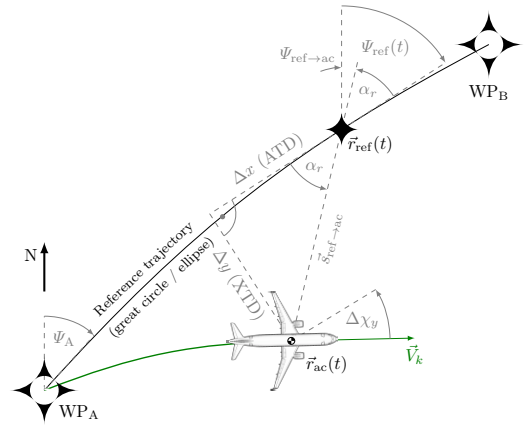


Figure 4. Variables for lateral tracking of a 4-D trajectory.

altitude (for control of lateral / vertical offset), see Figure 4. The FMS can also be a part of a larger block involving an optimization to provide optimal trajectory guidance. In this case, the trajectory is parameterized (for example by polynomials/splines) and models a position (and time) trajectory that can be adjusted by an optimization algorithm to fulfill constraints and / or minimize objective functions (for details see (Müller 2020; Müller 2021)).

4 Flight control system

As mentioned in the introduction, for a cascaded flight controller, a separation can be made with regards to the time-scales of the control variables, and to the degrees of freedom of the aircraft.

4.1 Autopilot

As the outer loop controller, the autopilot keeps the aircraft on the desired trajectory specified by the FMS, usually in the variables altitude h , velocity V , heading angle Ψ as well as track length or time reference (sometimes sideslip angle β is also directly controlled, e.g. to facilitate cross-wind landings.). A well known and widely adopted autopilot controller for the longitudinal motion of the aircraft is the Total Energy Control System (TECS), see (Lambregts 2013a) for details. Considering the equilibrium of potential and kinetic energy of the aircraft,

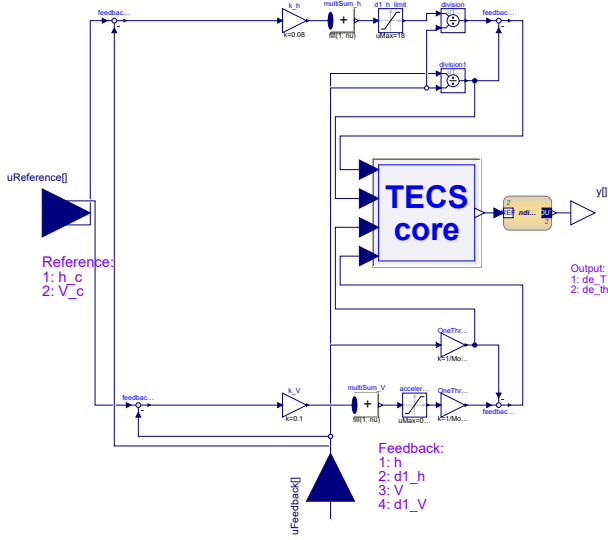


Figure 5. Realization of the Total Energy Control System (TECS). For 4-D trajectory tracking this controller serves as base class and is augmented with control of the ATD. By keeping ATD to zero, it is ensured that the aircraft will not fly ahead or behind the reference time-wise.

it builds upon the longitudinal force equation of a mass point:

$$m\dot{V}_k = T - D - mg \cdot \sin \gamma_k. \quad (2)$$

When assuming trimmed horizontal flight, errors in $\dot{V}_k$ and γ_k are cancelled with thrust and pitch commands δT and $\Delta\Theta$, working as energy generating and energy distributing control respectively. This coupling hence allows to control both channels in one block. An implementation using the FLIGHTCONTROL library is shown in Figure 5. A similar concept for the lateral aircraft motion was devised in the form of Total Heading Control System (THCS) (Lambregts 2013b), which is also implemented in the library but not shown here.

4.2 Inner loop

For the inner loop controller (sometimes denoted as Stability and Control Augmentation (SCA)), a separation into the roll/pitch/yaw channels (and thrust) can be made for fixed wing aircraft, as their longitudinal and lateral movement are usually not tightly coupled. The task of an inner loop controller is to increase mode damping and stabilization by generating appropriate control surface deflection commands (aileron, elevator, rudder) out of the reference inputs (e.g., commanded roll-, pitch- and yaw rates p_c, q_c, r_c) received from the autopilot. A number of controller architectures are possible here, implementing for example classical linear PID controllers (see e.g., (Brockhaus, Alles, and Luckner 2011; Stevens, Lewis, and Johnson 2015)) and model-based (e.g., dynamic inversion) approaches.

As a concrete example for an inner loop controller implemented in the library, see Figure 6. It represents a pitch

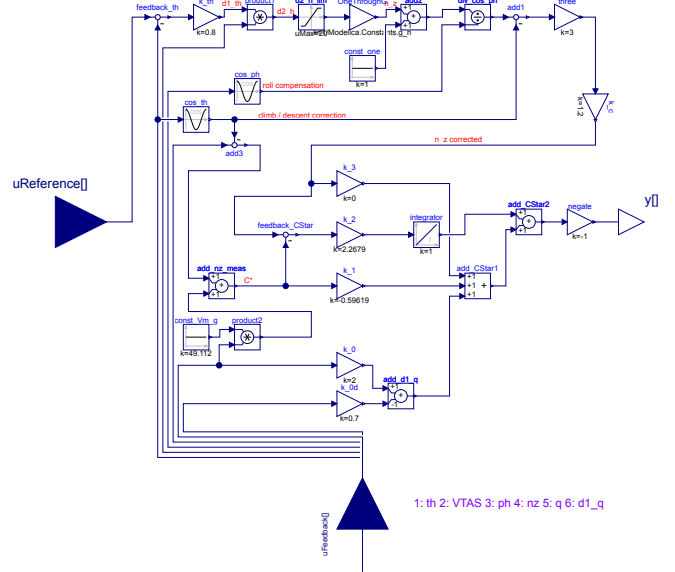


Figure 6. Aircraft pitch controller implementing the C^* - criterion for control performance and passenger comfort.

controller, implementing the well-known C^* - criterion, where:

$$C^* = \frac{1}{g} (-\ddot{z} + x_{\text{pilot}} \dot{q} + V_m q) \quad (3)$$

is a weighting between vertical acceleration at the pilot position $(-\ddot{z} + x_{\text{pilot}} \dot{q})$ and pitch rate q . Since pilots judge aircraft reaction at high speeds depending on the former and at low speeds on the latter, this variable is well suited as pitch control variable throughout the flight envelope. The upper part converts pitch- to vertical load factor commands (also corrected for turning flight), which is then compared with the measured acceleration. Adaptation to an aircraft model can then be achieved by adjusting the five gains aligned in the middle, e.g. via pole placement and using aircraft derivatives obtained from linearization (which usually has to be done for a multitude of operating points in the flight envelope). An alternative to this for robust controller synthesis is an iterative process consisting of worst-case search across the envelope, multi-case optimization and Monte-Carlo analysis, which is shown in Section 5.

In addition to these structured flight control concepts, Multiple-Input Multiple-Output (MIMO) controllers can also be easily integrated, for example by using appropriate classes from the LINEARSYSTEMS2 library. Owing to the acausal modeling feature of MODELICA, a long tradition of using inverse models within the FLIGHTDYNAMICS library is preserved here in several controller implementations using concepts like NDI (Enns et al. 1994). All of the mentioned approaches however require model information, which is not included in the FLIGHTCONTROL library and has to be provided separately.

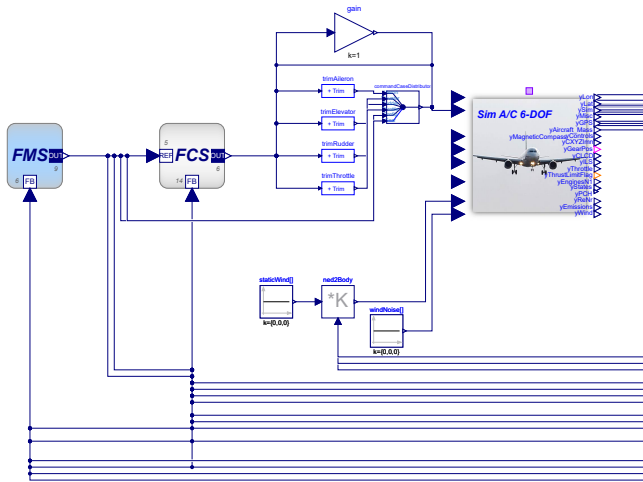


Figure 7. Assembly of FMS and Flight Control System (FCS) with aircraft model.

4.3 Special control functions

The library also contains functions that are not part of primary flight control. This includes for example controllers for lateral and vertical gust load alleviation, which were deployed in (Müller and Ritter 2017). Controllers implementing Pseudo Control Hedging are for example important in trajectory optimization where non-feasible/flyable trajectories must be hedged with respect to different variables (mostly thrust). This is also crucial for simulation of High Altitude Pseudo Satellite (HAPS) aircraft with their very narrow velocity envelope, as was shown in (Müller, Kiam, and Mothes 2018). Related to landing, there are controllers for high-lift systems, which are based on various available specifications by manufacturers. Regarding autoland, the library also contains controllers to follow the glideslope and glidepath, for managing flare, ground-roll and taxi. In the following example scenario, a respective autoland controller is designed.

5 Simulation example

An application of the FlightController library in conjunction with the FLIGHTDYNAMICS library as well as an optimization tool is shown in this section. The goal of this scenario is to investigate effects of sensor malfunction and disturbances on safe landing performance of an aircraft using GPS- and ILS-aided autoland systems. The scenario is based on the description given in the Civil Aircraft Landing Challenge (CALC) proposed by Airbus and Onera in 2016 (AIRBUS 2016).

5.1 Modeling and scenario description

The 6-Degree of Freedom (DoF) aircraft and systems model is provided by the FLIGHTDYNAMICS library, as is shown in Figure 7. The GPS-, glideslope- and localizer sensors are the relevant systems in this setup and are subjected to disturbances listed further below. The trajectory is part of the NAMUG 26R RNAV approach to Munich

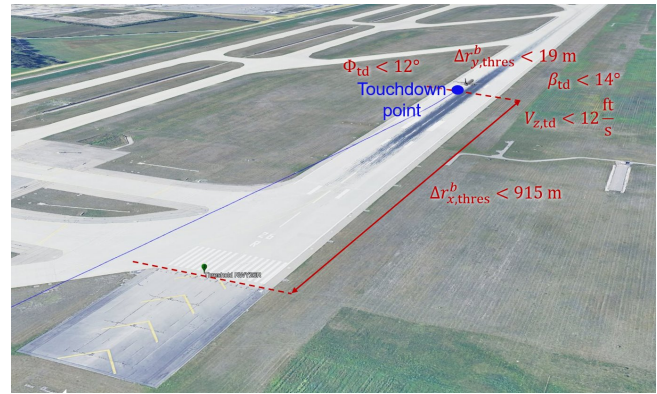


Figure 8. Touchdown criteria as specified by the CALC document.

airport, starting from waypoint DM428 and landing on RWY26R, where the aircraft has to conduct a right turn. For this scenario, the 4-D FMS was not activated, since the focus was not to model aircraft scheduling but instead the adherence to the autoland criteria specified by the CACS challenge. The main objectives the aircraft has to fulfill when touching down are:

1. Absolute bank angle Φ_{td} lower than 12° ,
2. Absolute sideslip angle β_{td} lower than 14° ,
3. Lateral distance from the centerline $\Delta r_{y,thres}^b$ smaller than 19 m,
4. Longitudinal distance from the threshold $\Delta r_{x,thres}^b$ smaller than 915 m and
5. Vertical velocity $V_{z,td}$ at touchdown smaller than $12 \frac{ft}{s}$.

The geometry of the constraints can also be seen in Figure 8. Disturbances are considered in the following systems and inputs:

1. A maximum of 100 ms in each of the roll/pitch/yaw and throttle channels of the controller, modeling worst-case cumulative time-delays in the whole controller.
2. Sensor noise in the GPS receiver with a maximum angle error corresponding to 5 m of position deviation.
3. Sensor noise in the glideslope and localizer sensors with a maximum signal boundary deviation of $\pm 10\%$ from the reference.
4. Winds with maximum values of 25/10/15 kts for head-/tail-/crosswinds. These values are translated in the model to wind strength and direction.

In order perform the autoland, a combined FMS and FCS controller was used which either tracks the 3-D GPS

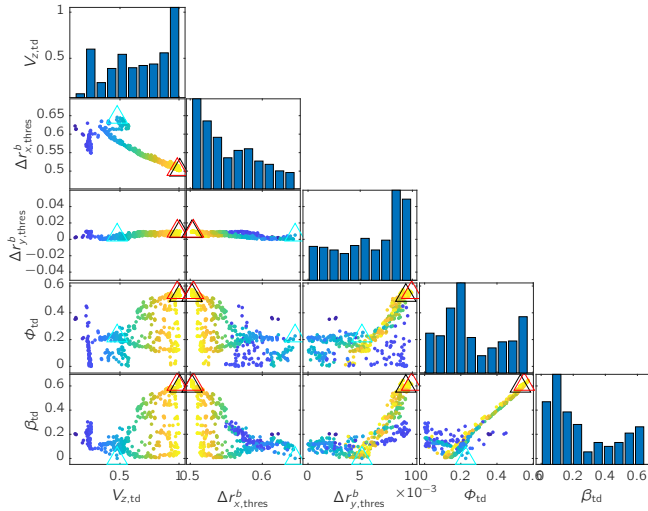
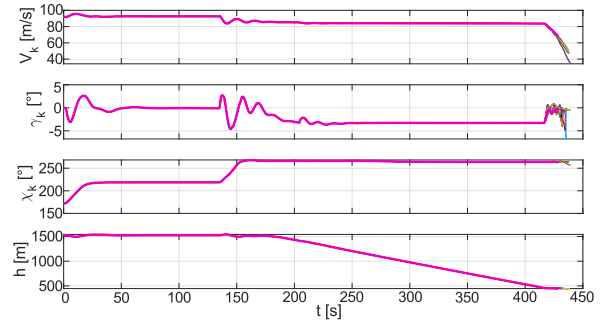


Figure 9. Anti-optimization results after 100 generations with values normalized to the CALC constraints (same colored data points belong together).

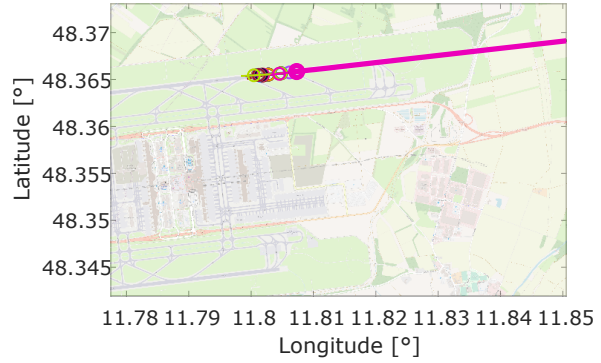
trajectory or controls lateral/vertical deviations from the localizer and glideslope signals to zero. For the flare and touchdown, a variable- τ law controller was implemented (for details see e.g. (Looye 2007)). In accordance to the CALC challenge specifications, the simulation is stopped after all three gears have weight on the wheels and commence braking, therefore excluding ground roll and taxi from the scenario. According controllers are however available in the library, for simulation examples (rejected takeoff, mission simulation) see (May, Müller, and Looye 2021).

5.2 Simulation and optimization

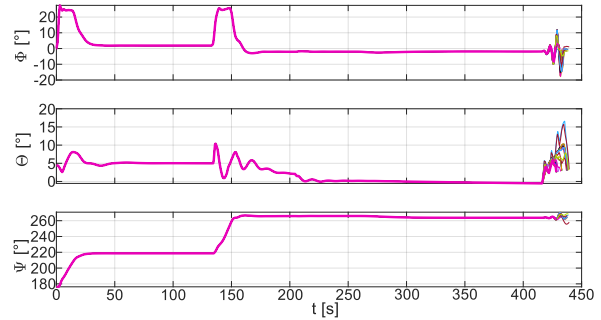
One way of obtaining a robust controller is an iterative process where in a first step, a worst-case search / anti-optimization was conducted in order to find the cases where criteria assume their worst value. This was realized in this scenario using a genetic algorithm optimizer (either e.g. from the OPTIMIZATION library (Pfeiffer 2012) or the DLR tool Multi Objective Parameter Synthesis (MOPS) (Joos 2016; Ossmann and Joos 2019)) with a population of 20 individuals and 100 generations. The parameters for the anti-optimization are the controller time delays, the sensor noises and boundary deviation as well as the wind disturbance mentioned in the last section. This yields the results shown in Figure 9, which represents 2-D cuts (one for each combination of the five criteria) through the 5-dimensional solution space generated by the optimization, along with histograms of occurrence. All the feasible, pareto-optimal solutions of the 100 consecutive generations are plotted (the colormap does not represent a value but shows which solutions belong together), and allows insight to possible criterion dependencies (for example, longitudinal deviation seems to decrease for higher touchdown speeds). It was found that only $V_{z,td}$ (marked with black triangles) generated critical results that are close or



(a) Kinematics variables.



(b) Runway map with final trajectory phase, touchdown points are marked as circles.



(c) Orientation in Euler angles.

Figure 10. Results of the controller optimization for the Instrument Landing System (ILS) guided approach.

higher than 1 (value is normalized with maximum). In the second step, the controller gains and time constants for path tracking / glideslope capturing and the flare law are optimized for worst-cases in a common optimization setup. This is achieved by consecutive evaluation of each cases and min-max weighting of the resulting criterion values (the optimizer tries to minimize the current highest normalized criterion value) until all constraints are satisfied or the optimizer cannot find any better solution. Due to the non-convex nature and complexity of the model and setup, a gradient-free optimization algorithm (pattern-search, (Hooke and Jeeves 1961)) was employed. Also since only $V_{z,td}$ exceeded its admissible value (larger than one), the other criteria were considered active during the optimization, but were not as critical as the velocity in the min-max sense. The results of this optimization is shown in Figure 10, with the optimal solution colored in magenta.

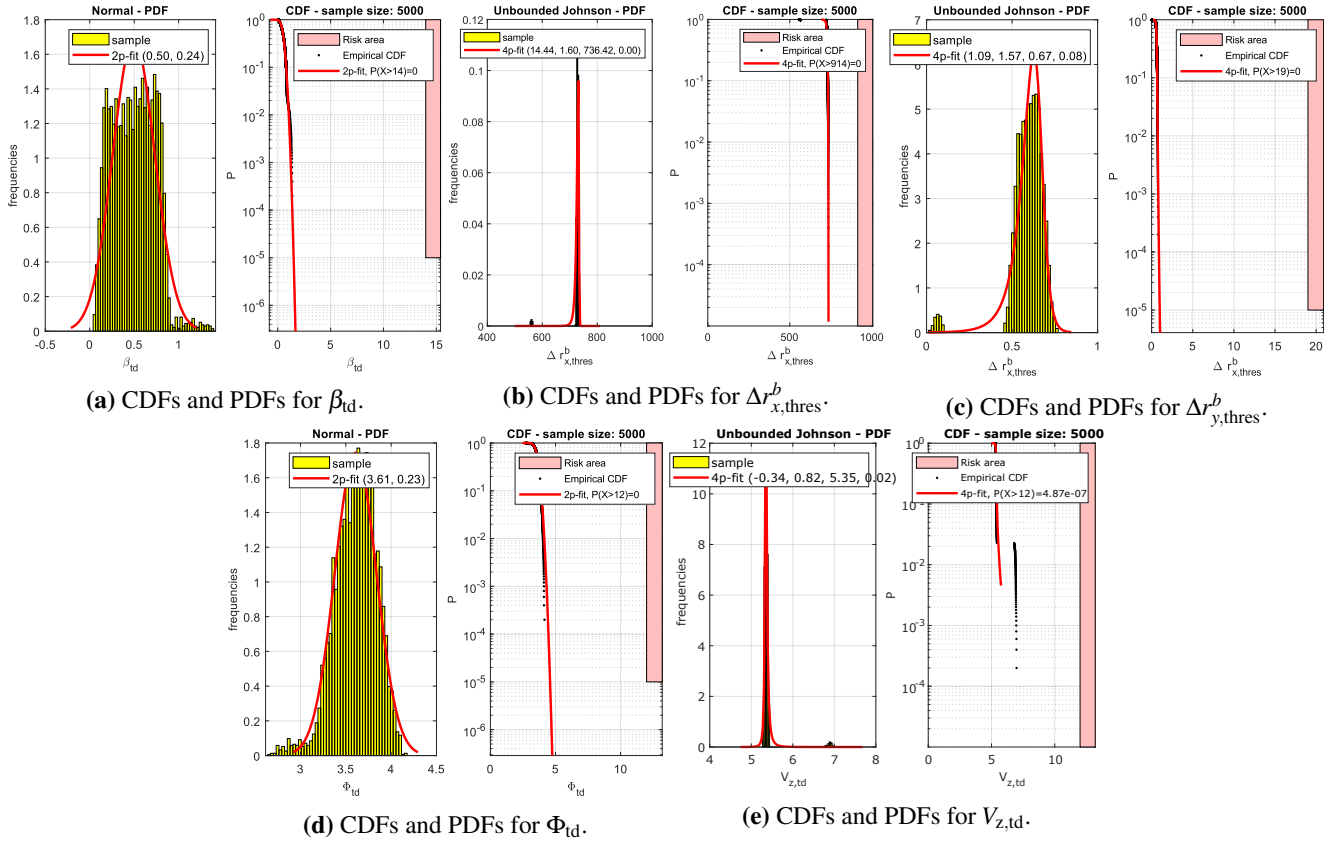


Figure 11. Monte-Carlo analysis of the found optimal controller subjected to random disturbances as specified in Section 5.1.

In the kinematics- (Figure 10a) and orientation (Figure 10c) plots, it can be seen that initial solutions touch down at slower speeds and higher incidence, which also leads to oscillations in the pitch axis due to the landing gear getting airborne again. Also cross-wind has a high effect during touchdown as the aircraft has not much control authority left to counter it. This leads to larger roll angles, which can also exceed the tolerances there. Furthermore there is a lateral deviation also caused through controller time delay and wind disturbance, creating an offset to the touchdown point and increasing deviation in the following roll phase (a roll controller is not implemented here). The ILS sensor noise however does not contribute to increased oscillations in the state variables which is probably due to its low magnitude and the damping of the controller. The final magenta controller however manages to fulfill all the requirements of the CALC including the longitudinal threshold deviation, which can be seen in the trajectory plot in Figure 10b.

In the third step, the obtained controller must be verified with respect to uncertainties and disturbances that might not have been present during the synthesis process. This is achieved in this case by a Monte-Carlo analysis taking into account the uncertain parameters listed in Section 5.1. The results of this process is shown in Figure 11, where 5000 simulations were conducted in an initial investigation. The failure probabilities postulated by CALC are shown as light red squares in the plots and may not

be exceeded if the controller shall be considered verified¹. For the criteria histograms, respective Probability Density Functions (PDFs) were fitted, leading to Cumulative Distribution Functions (CDFs) that mostly fit the obtained samples very well. One exception is the vertical velocity where small secondary occurrences are present around $7 \frac{ft}{s}$ which are not covered by the unbounded Johnson PDF. This would make a sample-based analysis necessary in order to state the risk. Apart from this, all samples stay clear of the risk area with good margins, which allows the preliminary statement that the controller is safe in this exemplary simulation scenario. If that would not be the case, the three step process would have to be repeated iteratively, until the third verification step is successful.

6 Conclusion and outlook

In this paper, an overview of the FLIGHTCONTROL library was given, which was developed as an addon to the well-established DLR FLIGHTDYNAMICS library. The layout of the library and the main contents were described, along with examples of the main building blocks of an aircraft flight controller. Although these examples are mainly related to fixed-wing aircraft control, the FLIGHT-

¹However, due to the small probabilities involved, the sample size should be much higher to ensure this with a certain confidence level. As this was considered out of scope for this exemplary simulation study and due to the long expected computation time, such a study was not performed here.

CONTROL library also allows to integrate rotary wing, drone and other aircraft concepts, due to its modularity and straightforward application with other libraries (like the LINEARSYSTEMS2 and MSL).

A high-fidelity simulation study of an autoland approach trajectory was selected to showcase the controller synthesis capabilities of the library. For this scenario, an autoland controller was developed and tuned which behaves robustly under a wide range of external and internal disturbances. This was verified by an anti-optimization with subsequent multi-case optimization for controller gain tuning and finally a Monte-Carlo analysis to determine the safety of the found controller.

In the future, the FLIGHTCONTROL library will be further extended with means to automatically determine a controller's performance (e.g. via optimization and integration of common performance and handling quality measures). Furthermore, the support for model-based design approaches shall be extended and possibly concepts from optimal control (e.g. Model Predictive Control (MPC)) be included.

References

- AIRBUS, ONERA (2016). "A Civilian Aircraft Landing Challenge". In.
- Brockhaus, Rudolf, Wolfgang Alles, and Robert Luckner (2011). *Flugregelung*. Springer. ISBN: 9783642014437. URL: <http://books.google.de/books?id=2IKXH3skXBwC>.
- Enns, Dale et al. (1994). "Dynamic inversion: an evolving methodology for flight control design". In: *International Journal of control* 59.1, pp. 71–91.
- Hooke, R. and T.A. Jeeves (1961). "Direct Search Solution of Numerical and Statistical Problems". In: *Journal of Association for Computing Machinery (JACM)* 8.1, pp. 212–229.
- Joos, Hans-Dieter (2016-06). "MOPS - Multi-Objective Parameter Synthesis, User's Guide V6.6". URL: <https://elib.dlr.de/104761/>.
- Lambregts, Antonius A. (2013a). "TECS Generalized Airplane Control System Design - An Update". In: *Advances in Aerospace Guidance, Navigation and Control*. Springer, pp. 503–534.
- Lambregts, Antonius A. (2013b). "THCS generalized airplane control system design". In: *2013 CEAS Conference on Guidance, Navigation and Control, Delft, The Netherlands*.
- Liersch, Carsten M. and Martin Hepperle (2011-12). "A distributed toolbox for multidisciplinary preliminary aircraft design". In: *CEAS Aeronautical Journal*. CEAS Aeronautical Journal Volume 2.Number 1-4, pp. 57–68. URL: <https://elib.dlr.de/74509/>.
- Looye, Gertjan (2007). *An Integrated Approach to Aircraft Modelling and Flight Control Law Design*. ISBN: 9789053351482.
- Looye, Gertjan (2008). "The new DLR flight dynamics library". In: *Proceedings of the 6th International Modelica Conference*. Vol. 1, pp. 193–202.
- Looye, Gertjan et al. (2014-09). "Object-Oriented Aircraft Modelling with the DLR FlightDynamics Library". In: 63. *Deutscher Luft- und Raumfahrtkongress*. Augsburg, Germany: Deutsche Gesellschaft für Luft- und Raumfahrt.
- May, Marc, Reiko Müller, and Gertjan Looye (2021). "Aircraft Mission Simulation with the updated FlightDynamics Library". In: *Modelica Conferences*, pp. 155–162.
- Müller, Reiko (2020). "The Gnomonic Projection for B-Spline parameterized 4-D Trajectory Optimization Problems". In: *AIAA AVIATION 2020 FORUM*. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2020-3205. eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.2020-3205>. URL: <https://arc.aiaa.org/doi/abs/10.2514/6.2020-3205>.
- Müller, Reiko (2021). *Four-dimensional multi-objective trajectory optimization using high-fidelity aircraft models and a projected parameterization : = Vierdimensionale multikriterielle Bahnoptimierung unter Verwendung hochgenauer Flugzeugmodelle und einer projizierten Parametrierungsmethode / vorgelegt von Reiko Müller*. eng. Aachen.
- Müller, Reiko, Jane Jean Kiam, and Federico Mothes (2018). "Multiphysical Simulation of a Semi-Autonomous Solar Powered High Altitude Pseudo-Satellite". In: *Aerospace Conference, 2018 IEEE, Big Sky, MT, USA*. Institute of Electrical and Electronics Engineers, pp. 1–16.
- Müller, Reiko and Markus Ritter (2017). "Virtual flight testing of a controller for gust load alleviation using FMI for cosimulation". In: *Proceedings of the 12th International Modelica Conference*. Vol. 132. Modelica Association and Linköping University Electronic Press, pp. 921–928.
- Ossmann, Daniel and Hans-Dieter Joos (2019). "Multiobjective optimization-based fault-tolerant flight control system design". In: *International Journal of Robust and Nonlinear Control* 29.16, pp. 5341–5355. DOI: <https://doi.org/10.1002/rnc.3955>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rnc.3955>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.3955>.
- Pfeiffer, Andreas (2012). "Optimization library for interactive multi-criteria optimization tasks". In: *9th International Modelica Conference*.
- Stevens, Brian L., Frank L. Lewis, and Eric N. Johnson (2015). *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons.