A Total Capability Approach for Development of Safety-Critical Functions

David Löbl[1], Nils C. Mumm[1], and Florian Holzapfel[1]

[1] *Institute of Flight System Dynamics, Technical University of Munich, Garching, 85748, Germany, david.loebl@tum.de*

In civil aviation, but also in many other disciplines, safety-driven probabilistic requirements are imposed on systems. In conventional development approaches, a required functionality is developed in a non-probabilistic, conservative way with independent performance budgets for individual subfunctions, thereby sacrificing possible system performance. As opposed to this, this paper presents a novel, model-based probabilistic development approach. This approach utilizes the knowledge of desired system dynamics, known uncertainties, and disturbances to gradually develop implementation-independent simulation models that are used for derivation of lower-level requirements. These models are also used to validate the requirements before actual implementation, which reduces development risk. The actual implementation is supported and optimized by using executable requirements that allow for automatic proof of compliance during simulation. This minimizes risk that the actual implementation violates safety-critical requirements and hence safety can be increased. Verification of implementation against probabilistic requirements is accomplished by stochastic simulations, using sophisticated algorithms that allow for an efficient evaluation of small probabilities related to safety-critical events. By dynamic allocation of admissible performance budgets to individual subfunctions during operation, the availability of the implemented functions is increased. This paper gives an introduction to this total capability approach for development of safety-critical functions. Examples for the individual steps of the specific development process are given, which show promising results.

I. INTRODUCTION

In civil aviation, but also in many other disciplines, the operability of safety critical functions must be ensured with only very small admissible failure probabilities. During system design and verification, one must ensure that the limits specified in safety driven, probabilistic requirements are not exceeded with a probability higher than a certain threshold. For example, the admissible probability for the occurrence of a catastrophic event of a commercial aircraft must be lower than 10E-9 per flight hour.[1] These top-level requirements must be broken down in a structured manner to a level that allows for design, implementation, and verification. In conventional approaches, this breakdown is often carried out in a non-integrated manner where individual and independent performance budgets are allocated to different functions and subfunctions. Furthermore, these required budgets are often derived in a non-probabilistic, conservative way. Both approaches result in a reduced total system performance. With increasing computational power, model based development approaches gain in importance. This paper introduces, as opposed to the conventional approach, a model-based probabilistic development approach that promises increased safety and availability as well as a better utilization of the total available system performance.

The focus of this paper is on describing the concept of this novel development approach, called Total Capability Approach (TCA). The paper is organized as follows. In Section II, the principle idea of the TCA is presented. In Sections III-VI, the different steps of this approach are described in more detail. A sample application is given in Section VII. Section VIII concludes the paper.

II. TOTAL CAPABILITY APPROACH

II.A. Principal Idea

The overall system is required to deliver a certain performance given by quantified limit values and the allowable probability of their exceedance. The behavior of the integrated system in the real environment is driven by all contributing elements (e.g. sensors, actuators, computers, disturbances, etc.) and is therefore affected by all disciplines contributing to their implementation. Motivated by this fact, in the TCA the overall system is considered simultaneously instead of allocating hard tolerance budgets to the individual disciplines and components. This requires a radical design philosophy change: Instead of an independent subcomponent and subfunction design, the overall function and implementation is simultaneously designed and optimized. Although there are still requirements derived for the subfunctions, the respective admissible performance budgets are not static, but are dynamically allocated during operation. For example, if the performance of two subfunctions sums up to the overall performance, if at some instant of time one subfunction performs well, then the other must only provide a worse-than-normal performance to obtain an acceptable total system performance (Fig. 1). This also highlights the motivation for this approach: While for static budgets, one would either have a lower availability of the total system or would require more performant functions and components, the TCA allows a dynamic allocation of admissible performance budgets and hence overcomes the aforementioned drawbacks.

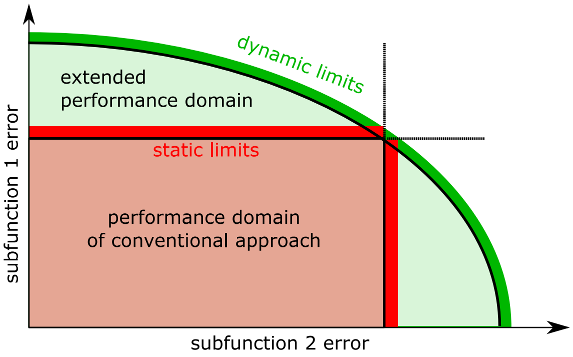


Fig. 1. Static vs. dynamic limits

Another major difference of the TCA is the consideration of statistical properties of all contributing variables, parameters and continuous processes during the design process. Up to now, often robust bounds are chosen to describe uncertainties. However, this either leads to an overly conservative systems using wide bounds or a rather unsafe system using relaxed bounds (Fig. 2). Using statistical properties to describe uncertainties allows a physically motivated inference on the contributions of individual uncertainties to critical events. For the case where an unlikely combination of uncertain parameters leads to an unfavorable system reaction, its probability is no longer overestimated as it would be the case using combinations of conservative parameter bounds. Furthermore, disturbances are often described by stochastic processes, with knowledge of intensities and according probability of occurrence.

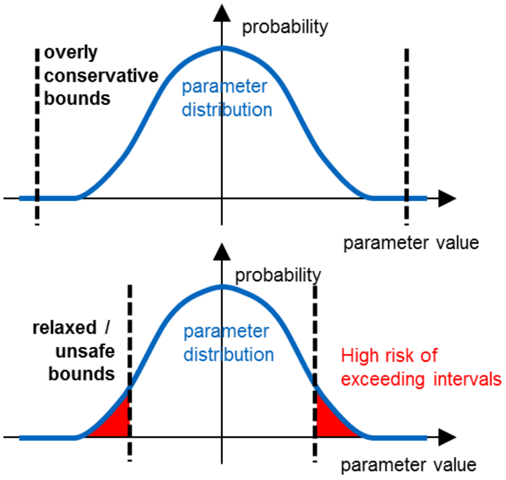


Fig. 2. Relaxed and overly conservative bounds compared to actual parameter distribution

II.B. Development Process

The TCA can be divided into the following main steps (see also Fig. 3):

- **Mission-driven derivation of lower level requirements**: Requirements are still formulated with respect to classical metrics, however linked to probabilities. Simulation models are built utilizing preknowledge about the system. Stochastic simulations are conducted to quantify lower-level requirements. Since considered probabilities are in general very low, enhanced stochastic algorithms are applied. Outputs of this step are system bounds for which compliance with top level requirements is guaranteed with a certain probability.
- **Model-based validation of requirements before system design**: Executable requirements are implemented into the simulation models. The models represent the system state during the respective development stage. Thorough analysis of system responses allows proof of completeness and consistency of specified requirements.
- **Design system according to requirements**: The actual system design process is conducted in a conventional manner. Utilizing the executable requirements, system design parameters (e.g. controller gains) are optimized to minimize probability of exceeding thresholds.
- **Verification of implementation against requirements**: Using the knowledge about possible disturbances and failures together with their probabilities of occurrence as well as system uncertainties allows a detailed stochastic evaluation of the probabilistic requirements.
- **Online monitoring**: Model-based online monitoring is applied to ensure compliance and hence safety not only during system certification but also during daily operation.

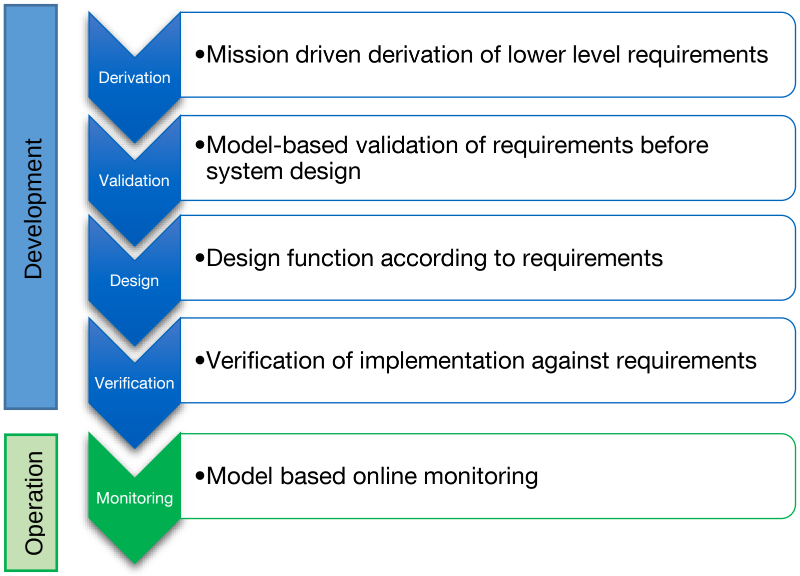

Development
Operation
Derivation
•Mission driven derivation of lower level requirements
Validation
•Model-based validation of requirements before system design
Design
•Design function according to requirements
Verification
•Verification of implementation against requirements
Monitoring
•Model based online monitoring

Fig. 3. Steps of the TCA

III. REQUIREMENTS DERIVATION

III.A. Motivation and Shift in Paradigm

For development of safety-critical functions, often only high level requirements are available and no or only unreasonable lower-level requirements are specified by certification authorities and customers. For example, although there are a lot of specific requirements provided by certification authorities to ensure safe flight of manned aircraft, these requirements are not necessarily suitable for unmanned aircraft. However, due to lack of alternatives, often such inappropriate requirements are still used as common denominator. This directly influences the achievable performance of a system, i.e. it is already limited from beginning by possibly unappropriate requirements. To overcome the disadvantages linked to this approach, the TCA follows a model-based breakdown of top-level requirements to lower-level requirements.

An important difference to conventional requirement derivation approaches is the consideration of stochastic processes, uncertainties, and disturbances already during system specification. While requirements are still defined by classic metrics, they are also linked to acceptable probabilities of exceedance. This is motivated by the fact that – when using a physically motivated description of uncertain parameters with distributions and disturbances with stochastic processes – there is always a certain probability of violating a requirement.

III.B. Model-Based Breakdown of Requirements

The TCA follows a top-down model-based requirements derivation approach where the formalization of requirements is driven by simulation. For that, adequate simulation models must be available. The models must provide the following functionalities:

- The simulation models must correctly represent the system dynamics which cannot be modified: Many systems for which a certain functionality must be developed have dynamics, that are inherent to the principal system and which cannot be altered by any inputs and hence cannot be influenced by any implemented function. This is for example the case for system kinematics. Also dynamics of subsystems that are not in the reach of the envisaged development, e.g. off the shelf components, must be correctly modeled, including uncertainties.
- The simulation model must provide means to adjust the relevant dynamics part of the model that can be influenced. However, no assumptions on possible or already envisaged implementations must be made during this step since this would reduce the possible solution space to solutions which use the envisaged implementation.
- The simulation model must reproduce the effects of disturbances and uncertainties in a manner that allows for an evaluation of its influences on system response. Often, adequate disturbance models and admissible system responses are defined by certification authorities or can be obtained from experiments. Still, one must adequately model the effects of disturbances on system dynamics.
- The simulation models must provide interfaces for relevant outputs, inputs, and for variation of uncertain parameters. Relevant signals in this context are signals that are required for requirements evaluation.

Using these simulation models, scenarios can be simulated according to requirements specification. The formulated requirements are translated to executable requirements, which allow for determination of compliance of requirements based on system states and pre-defined simulation scenarios. The procedure for model-based requirements derivation is the following:

1. Break down top-level requirements in a conventional manner, i.e. establish requirements for sub-functions, however, with placeholders for quantitative values, and describe the conditions under which the requirements must be fulfilled.
2. Set up simulation scenarios according to the conditions specified in the requirements. This includes for example initial conditions and disturbances.
3. Conduct simulations to find the boundary surface in the influenceable dynamics space for which the requirement is just exactly fulfilled, i.e. where the probability of violating the requirement is exactly the probability specified in the requirement.
4. Select an adequate subspace within the boundary surface for quantification of the requirements specified in step 1.

A schematic structure of the requirements derivation environment is shown in Fig. 4. There are two major challenges linked to this procedure: First, the determination of the boundary surface is a very challenging task, especially since the probabilities that are linked to safety-critical functions are often very small and stochastic processes for uncertainties and disturbances add up to a high number of uncertain parameters for discrete simulation. Hence, conventional Monte Carlo algorithms for probability estimation are no viable option. Several solutions to this problem are conceivable. In the simplest case, requirement, stochastic processes, and plant are linear and an analytical solution is possible. Since this is often not the case, more sophisticated simulation methods (e.g. Subset simulation[2], importance sampling[3]) are a viable option. Second, the dimension of the solution space – i.e. the order of influenceable dynamics - can be relatively large for common systems. Although the high-dimensional solution space could be described by complex numerical formulations in the requirement, this is no viable option in the context of actual development. Instead, the solution space can be split up in several one and / or two dimensional subspaces for which a reasonable system design is possible. However, this requires algorithms that find the best combination of subspaces that are a subset of the solution space, see for example Ref. 4.

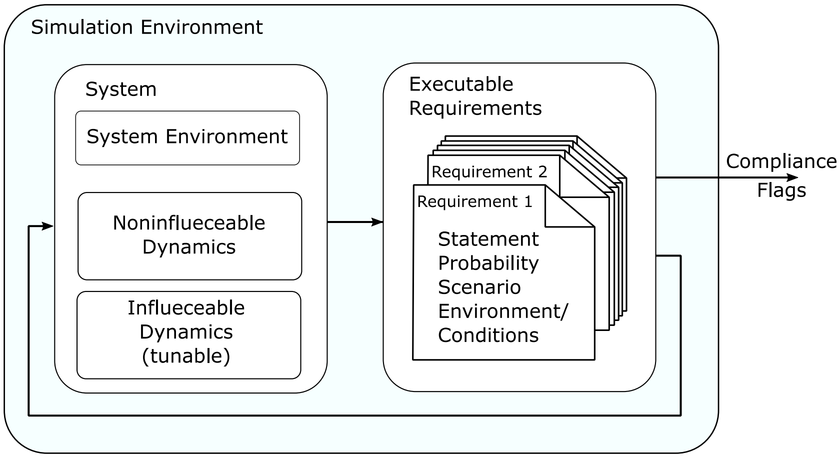


Fig. 4. Simulation environment for requirements derivation

IV. REQUIREMENTS VALIDATION

Validation is the assurance that a specified functionality meets the needs defined by customers and stakeholders.[6] This is ideally conducted prior to actual system design. However, often this is done in parallel to system design which could lead to changes of or supplements to system specifications and hence changes of design. To overcome these disadvantages, the models developed during requirements derivation can also be used to validate specifications prior to actual function design. The model-based validation process in the scope of the TCA especially ensures two major properties of the requirements set: completeness and consistency.

IV.A. Completeness

Completeness means that the set of requirements is sufficient and necessary to perform the tasks specified by the stakeholders. Sufficiency ensures that the function performs as intended. Necessity means that not more requirements than necessary are specified, i.e. that no functions are specified that are not required by stakeholders or for providing the intended function. In conventional approaches, proof of completeness is often done by engineering judgement, i.e. experts review a hard copy of the requirements and decide whether they are sufficient or not. This validation process can be significantly enhanced by using the simulation and requirement models developed during requirements derivation. Using these models, one can simulate the intended functionality. If the simulation passes through from begin to end without unexpected behavior and under all conditions envisaged during system specification, the set of requirements is sufficient to describe the intended functionality. Recall that at this stage the simulation models do not include any knowledge of an actual implementation and simulations are performed before actual system design.

IV.B. Consistency

In general, consistency means that a requirement does not contradict any other requirement. In the context of TCA, one can distinguish between internal and external consistency. Internal consistency ensures that the requirements specified for a certain functionality do not contradict each other. This is automatically given when following the derivation process for the TCA described in chapter III, where requirements are specified in a way that they do not infringe each other. External consistency means that the specifications of a certain functionality are in no contradiction to specifications of other involved systems that are not in the scope of the actual development. When having a multi-domain simulation environment, it is imaginable that such evaluations could be also done partly automatically. However, expert knowledge is still required to identify possible interferences between different systems.

V. DESIGN AND VERIFICATION

In this chapter, the procedure for design, implementation, and verification in the scope of the TCA is discussed.

V.A. Design and Implementation

As mentioned in chapter II, requirements are still formulated with respect to classical metrics, i.e. system design can be accomplished in a conventional manner. During requirements derivation the desired closed-loop behavior of the system is modeled without any knowledge of the actual implementation. During functional design and implementation, the universally formulated desired closed-loop behavior is replaced by the actual implementation of the function and its interaction with the original plant dynamics (Fig. 5). The framework and the executable requirement models developed and implemented during requirements derivation can be directly reused to determine compliant system behavior but also for design optimization. For example, if the decision is made for a certain control structure, the tunable parameters of the control law can be optimized to best fulfill all requirements.

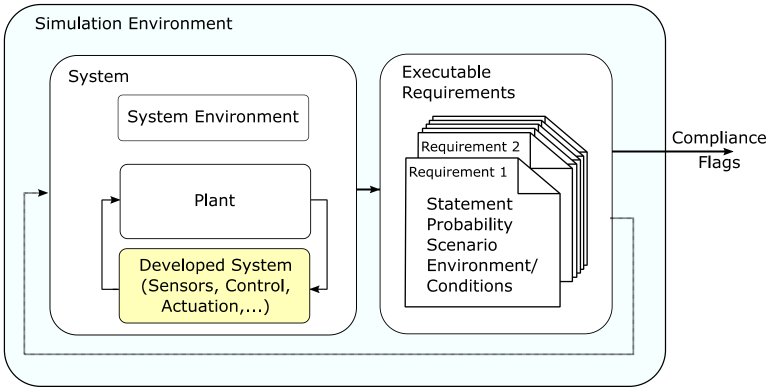


Fig. 5. Simulation environment for design and verification

V.B. Verification

Verification is the assurance that the implemented functionality meets the specified requirements. Theoretically, when the function is designed, tuned, and verified according to the derived requirements, all low-level requirements and consequently the top-level requirements are fulfilled. However, not all influences and dynamics can be considered during requirements derivation. Hence, additionally to the proof of compliance of the low-level requirements, a direct evaluation of high-level requirements is possible. For that, the knowledge of all uncertainties, disturbances, and failures stemming from system design, architecture, and contributing dynamics is used to evaluate the probability of critical events by stochastic simulations. This is done with similar algorithms as for requirements derivation, but with the actual implementation and additional uncertainties stemming from system design instead of the artificial behavior models. For reasonable simulation of the actual implementation, often highly complex and computationally intensive simulation models with sub-models for different aspects like actuation, sensors, and data buses are used. Each of these sub-models adds additional uncertainties to the system which must be considered during stochastic evaluation. This underlines the necessity for highly-efficient stochastic algorithms.

VI. ONLINE MONITORING

VI.A. Online Monitoring Method

Online monitoring is – in contrast to the steps described before – no part of the system development process. However, it is a vital component of the TCA since it enables dynamic allocation of performance budgets during daily operation. This leads to increased availability since a worse performance of one subsystem can be compensated by a better performance of another subsystem. Therefore it is required to know the current system performance and how it will develop during a certain short time horizon. Having this knowledge, compliance of the implemented function with its requirements can be ensured and adequate measures can be triggered in case that requirements are violated. Figure 6 shows the principle idea of a dual-step online monitoring approach:

1. Low-level requirements are monitored on their compliance. During requirements derivation, requirements are put on the admissible dynamic response. Based on this admissible behavior, uncertainty bounds are propagated within which the system response lies if it complies with its specification. Admissible disturbances and uncertainties specified in the requirements are incorporated into the calculation of the bounds, leading to wider uncertainty intervals. In the presence of stochastic disturbances, the uncertainty bounds get a probabilistic nature, i.e. there is a certain probability of exceeding these bounds although the system behaves according to specification. This is caused by the fact that for stochastic disturbances there is always a small probability for greater-than-average disturbances. For a possible implementation of such a monitoring algorithm see Ref. 6. The referred monitoring algorithm can also be used for modeling of the executable requirements that are required for system validation and design with the major difference that when using simulation models, all states are directly accessible in contrast to online monitoring where only a smaller number of states distorted by measurement uncertainties is available.
2. Often, the total performance of a system is provided by different functions with different uncertainties. Fig. 6 shows the case where the total performance is provided by two subsystems with individual uncertainty bounds. This could be for example the performance of a controlled system, with control accuracy on the one hand and measurement uncertainties on the other hand. Assuming independence of the uncertainties, the overall variance bounds can be calculated as the sum of variances. This overall variance bound is linked to a certain probability. This means that by using multiples of the variance bounds, a qualitative and often also quantitative statement about the probability of exceeding limits (indicated by the red limit bounds in Fig. 6) can be given. The limits and related probabilities can be directly taken from higher-level requirements.

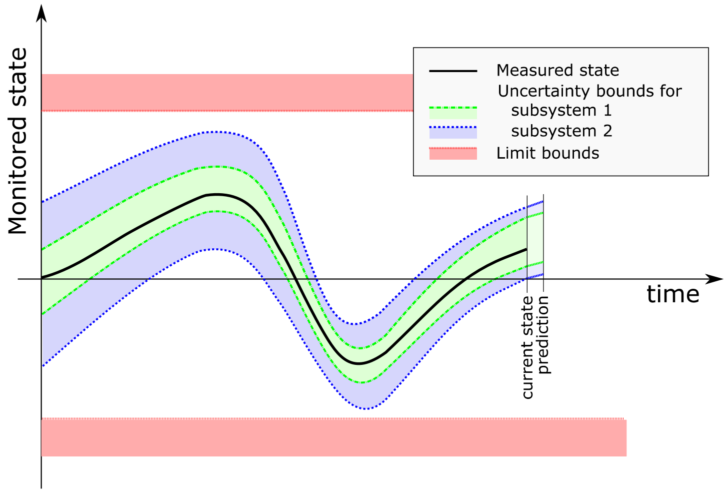


Fig. 6. Online monitoring

Using this intertwined, dual-step approach, it is ensured that the system behaves according to requirements. Disturbances and failures leading to non-conformal system behavior are detected and dynamic allocation of budgets is enabled. Furthermore, using the knowledge about the acceptable dynamics of the individual subsystems, uncertainty predication enables the identification of critical situations before they occur. Further enhancements of this monitoring approach can be for example the usage of disturbance observers instead of propagating maximum admissible disturbances as specified in requirements. By that, the uncertainty bounds could become even tighter, further increasing the overall availability.

VII. EXAMPLE

VII.A. System Description

The development of an automatic formation flight control system for large transport aircraft is exemplarily used to demonstrate the individual steps of the TCA. In the specific example, it is assumed – opposed to conventional military aerial refueling – that a tanker aircraft is controlled to maintain an adequate relative position with respect to a receiver, while the receiver aircraft flies straight and level utilizing standard autopilot functionalities. The objective is to develop an autopilot for the tanker aircraft that is capable of maintaining a certain desired relative position. Open loop system dynamics of the tanker is known, same as closed loop dynamics of the receiver with its baseline autopilot.

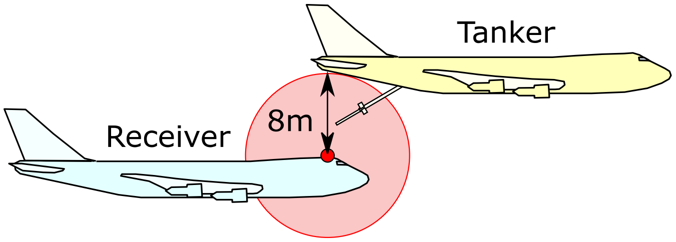


Fig. 7. Formation flight configuration

VII.B. Requirements Derivation

The top-level safety requirement for the considered application is that the probability of a collision between the aircraft must be smaller than 10E-9 per flight hour. This requirement can be broken down to subfunctions, e.g. lateral control, longitudinal control, thrust control. For demonstration of the TCA for requirements derivation, the following requirement is looked at:

“The probability for a vertical position error of more than 8 meters towards the receiver aircraft must be less than 10E-9 per flight hour.”

Environmental conditions under which this requirement must be fulfilled are specified for example in Ref. 7. Here, turbulences with moderate intensity according to the Dryden wind turbulence model are considered, which have a probability of occurrence of 10E-3, leading to the requirement, that the given threshold must not be exceeded with a probability higher than 10E-6 in the presence of moderate turbulences. Figure 8 shows the structure of the simulation model used for requirements derivation. For the tanker aircraft, a full state feedback is used to modify the closed-behavior of the tanker. Note that no controller structure is assumed, only the full state vector is used to shift the poles of the plant to possible desired positions. For the receiver, the closed-loop dynamics of the altitude hold autopilot is modeled. The Dryden wind turbulence model is used for generation of turbulence histories. This model is a linear filter which takes Gaussian white noise as input. The same turbulence velocity acts on both aircraft with only a small time delay similar to the longitudinal separation between the aircraft divided by the airspeed. This is a reasonable assumption especially concerning the low-frequent part of the turbulence excitation which is crucial for aircraft reaction. The disturbance acts via two channels on the relative dynamics between the aircraft: First, directly via the aerodynamics of the tanker aircraft (plant disturbance), secondly via the receiver dynamics, which causes an error in relative position and velocity (output disturbance). Hence, the receiver can be considered as additional source of disturbance in this development task.

The afore introduced requirement can be further broken down to specific requirements that can be directly used for system design. One of these requirements is for example concerning the desired closed loop dynamics of the tanker during formation flight. When only having a look at the vertical motion and neglecting actuator dynamics, the transfer function from the control input (elevator) to the vertical position is of forth order. Physically described, this is the second-order rotational motion of the aircraft from pitch moment input to an angle of attack, causing a vertical acceleration, which then leads to the vertical velocity and position after one and two integrations respectively. Hence, the dynamics range of a fourth-order system is looked for, for which the above stated requirement is fulfilled. The forth-order dynamics can be described by two second-order dynamics for the rotational and translation motion respectively. For each of these second order systems, damping and eigenfrequency are varied independently.

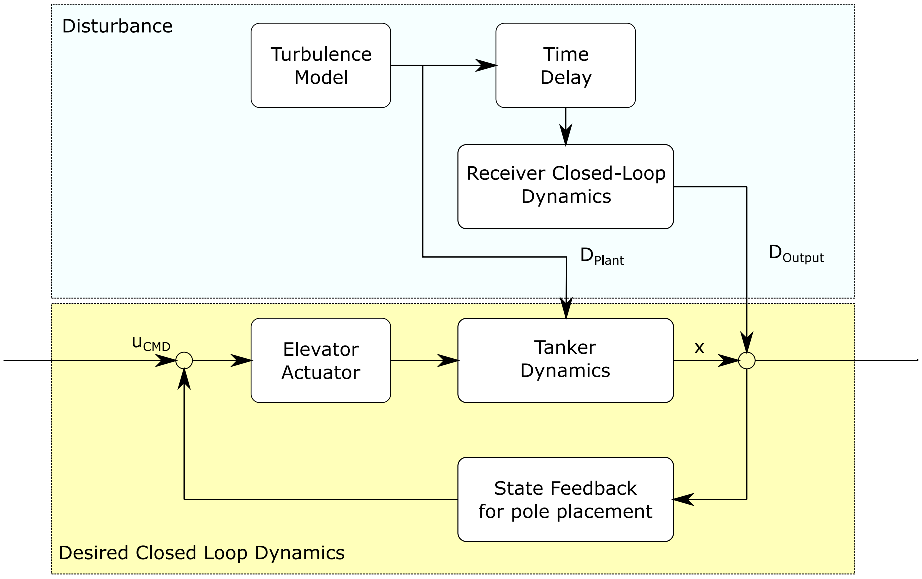
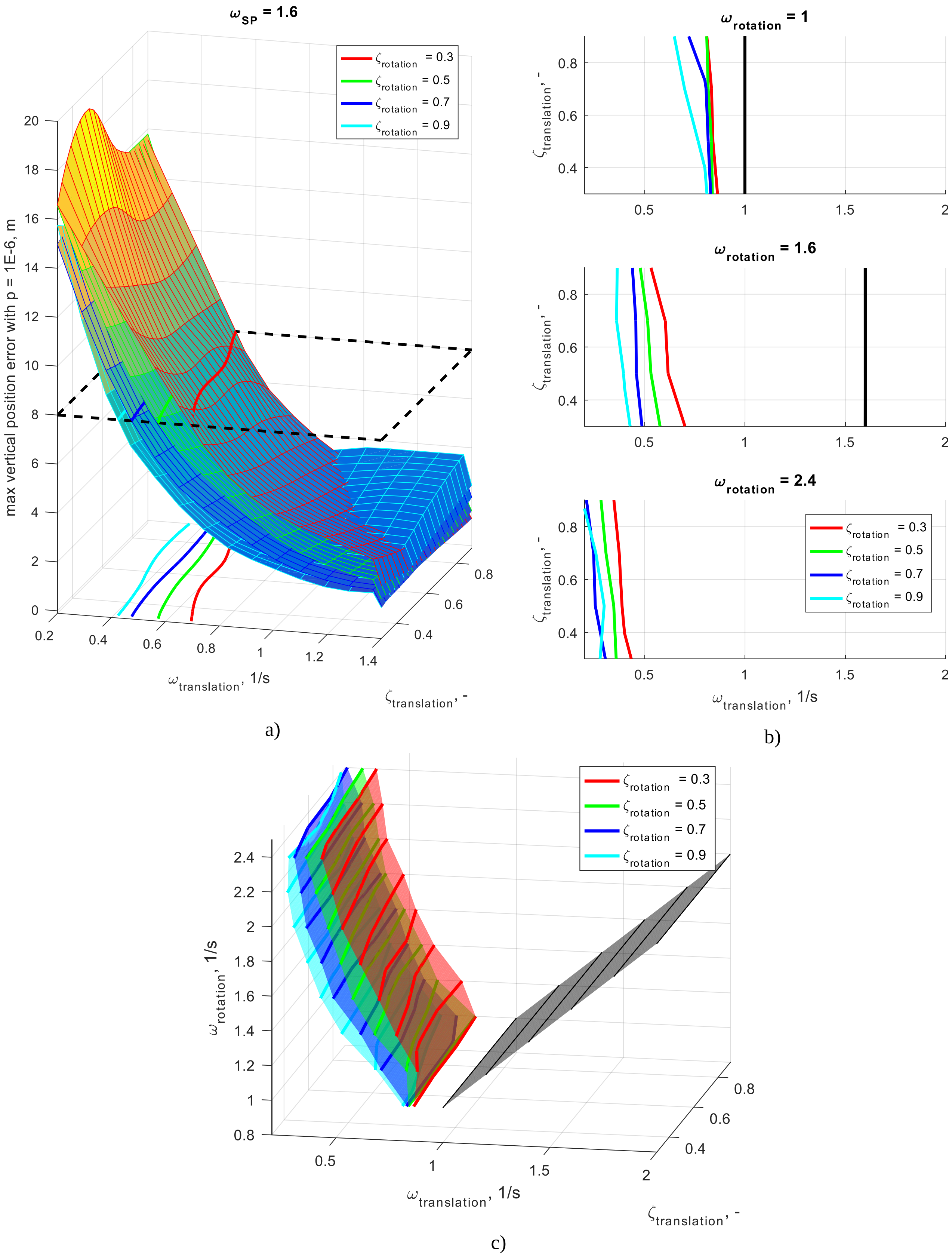


Fig. 8. Simulation model structure for requirements derivation

The objective of this task is to determine the acceptable combinations of the four parameters. The solution requires a two-step approach: First, for each contemplable parameter combination, the threshold is evaluated which is not exceeded with a probability higher than 10E-6 in the presence of disturbances and uncertainties. Second, the boundary parameter surface is looked for where the threshold of 8m is theoretically exceeded with a probability of exactly 10E-6. For the first, Markov Chain Monte Carlo simulations with a modified Metropolis Hastings sampler are used. [2] For the sake of simplicity of the example, the second is accomplished using a 4-dimensional grid for the parameter and interpolation between grid points.

Figure 9 presents exemplary results for this process. Each 3D surface displayed in subfigure a) gives the position error that is not exceeded with a probability higher than 10E-6 for different combinations of eigenfrequency and damping of the translational motion. The individual surfaces stands for different damping ratios of rotational motion. This plot is drawn for a single eigenfrequency of the rotational motion. The black dashed line highlights the vertical position limit that must not be exceeded with the specified probability. The intersections between the 3D surfaces and the requirement limit lead to boundaries that separates the parameter combinations into acceptable and non-acceptable regions. Subfigure b) gives these limit lines for different eigenfrequencies of the rotational motion and c) visualizes the admissible solution space in 3D. The black lines and surface in b) and c) respectively are the limit arising from the fact that translational motion cannot be faster than rotational motion for conventional aircraft configurations using only the elevator for altitude control. The last step during requirement derivation is to find a box within the admissible solution space. Actually, the admissible solution space is an intersection of the individual solution spaces of different requirements. For example, the boundaries for maximum load factors would limit the solution space towards higher eigenfrequencies. For system design, acceptable parameter intervals are required that do not or not too much depend on other parameters. A tradeoff must be made to find the best acceptable solution space, i.e. box in the solution space. When only considering the given requirement and the derivation results in Fig. 9. c), a higher range of admissible eigenfrequencies of rotational motion comes along with a decreased admissible range of eigenfrequencies for translational motion. Note that the different lines and surfaces for different damping ratios of rotational motion are only a mean to visualize the higher-dimensional solution space. The visualization shall only highlight the process of deriving limits for lower level requirements based on higher level requirements. For more than four design parameters or more complex geometries of the solution space caused by multiple requirements or more complex system dynamics, the best possible solution space must be found computationally. This is currently a topic highly researched on, see e.g. Ref. 4.



a)

b)

c)

Fig. 9. Requirements derivation results

VII.C. Validation

This section gives a brief description of how validation for the given example would look like. Due to the complexity of this process, no specific results can be given in the context of this paper. Completeness of requirements for the formation flight example is shown by the ability of the specified autopilot to maintain relative position under certain specified environmental conditions during simulation analysis. Consistency includes the interaction of the specified system with other subsystems. Examples include the evaluation of interference between the flight control system and mechanical design. By using aeroelastic models for the aircraft structure, it is possible to verify that the specified closed-loop system dynamics do not excite critical structural modes, which could lead to structural damages.

VII.D. Design and Verification

For design and verification of the formation flight autopilot, a sophisticated simulation environment is used. Detailed nonlinear aircraft simulation models of four-engined civil transport aircraft with a wingspan of almost 60m and a maximum take-off weight of 330 tons are used, including engine and actuator dynamics and nonlinear aerodynamic models.[8] Relative position and velocity estimation is done by a sensor suite, where radio frequency ranging transponders, optical sensors, as well as inertial and GPS measurements are used.[9] Relevant sensor errors are modeled as stochastic processes reproducing realistic error modes. Data buses between sensors, sensor data fusion, and control and guidance are modeled considering quantization, discretization and latency effects. Sensor data fusion and control are discretely implemented.

The relative position between the refueling receptacle of the trailing receiver aircraft and a reference point of the leading tanker aircraft is controlled by a nonlinear dynamic inversion based controller, where the relative kinematics between the two mentioned points is inverted to obtain acceleration inputs for the innerloop controller of the tanker.[10] Similar as for the requirements derivation, Markov Chain Monte Carlo simulations are performed to proof compliance with the stochastic requirements. Opposed to the requirements derivation process, only the implementation with its resulting closed-loop dynamics is evaluated instead of a whole range of possible plant dynamics. However, these simulations are computationally much more expensive due to the high complexity of the simulation model and the even higher number of uncertainty parameters. The Gaussian noise input for the turbulence excitation already increases the number of uncertain parameters by the number of discrete simulation steps per disturbance channel. Also sensor noise and uncertainties of the contributing systems, e.g. aircraft and actuator dynamics, can and must be considered. The consideration of statistical properties of all contributing variables, parameters and continuous processes during the design and especially verification process is required to obtain acceptable verification results for certification authorities. See Ref. 11 for more details on stochastic evaluation of low probabilities for this example.

VII.E. Online Monitoring

The dual-step approach for online-monitoring described in chapter VI is demonstrated based on the high level requirement for relative position control accuracy and the exemplarily derived lower-level requirement on acceptable rotational and translational dynamics. The compliance of the lower-level requirement is monitored using incrementally propagated bounds. The idea behind the approach is to use the knowledge of the acceptable system behavior and the plant inputs to propagate the acceptable state boundaries and compare these to the available measurements. Since this would lead to a divergence of propagated bounds, measurements and the knowledge about their uncertainties are used to update the estimated bounds. The incremental limits are probabilistic bounds since the uncertainties considered during propagation are also of stochastic nature. Hence, there is always a certain probability that the boundaries are violated during operation. This can be compensated by using a multiple of the propagated variance bounds for notifications. By that the probability of fault negatives can be reduced. Figure 10 exemplarily shows the results of monitoring the relative position control performance for an acceptable and a non-conformal plant dynamics on the left and right hand respectively. See Ref. 6 for more details on the online monitoring algorithm. For the second stage of this online monitoring approach, the propagated uncertainty bounds for the relative position are compared to the limit values specified in the requirement, for the requirement introduced above this is 8m. As already mentioned, by scaling the multiplier for the variance of the propagated bounds, one can control the probability that the system response lies outside of these bounds. A probability of 10E-6 for exceeding these bounds can be obtained by choosing the right multiplier. So both, closed-loop system dynamics referring to lower level requirements and the compliance of higher level requirements, e.g. on relative position, can be monitored.

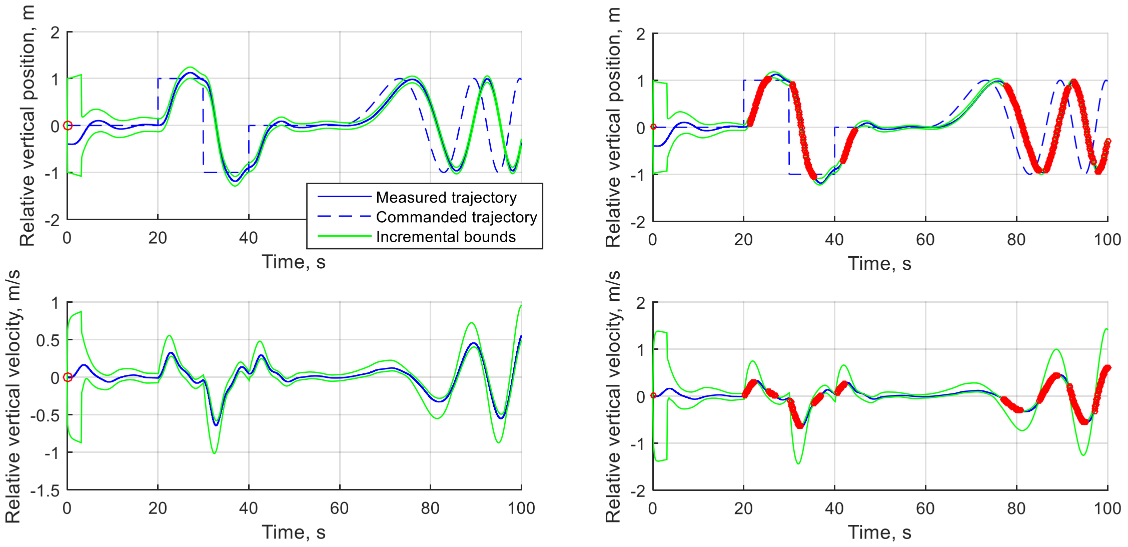


Fig. 10. Exemplary online monitoring results[6]

VIII. CONCLUSIONS

This paper presented a total system capability approach for model-based system development that utilizes all available knowledge about a functionality to be developed, including system dynamics and uncertainties, throughout the whole development process. Following this approach, a system is designed less for best behavior during nominal operation but rather for safest operation, i.e. to best fulfill safety-related probabilistic requirements. The principal idea for the different development steps were introduced and examples and references given where applicable. Currently, work is done towards a proof in practice, developing a real automatic landing system for a twin-engine fixed-wing aircraft.

ACKNOWLEDGMENTS

This research was supported by a Grant from the GIF, the German-Israeli Foundation for Scientific Research and Development.

REFERENCES

1. Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes CS-25", European Aviation Safety Agency, Cologne, Germany, AMC25.1309, 22 Jun. 2016.
2. Au, S. K., Beck, J. L., "Estimation of small failure probabilities in high dimensions by subset simulation," *Probabilistic Engineering Mechanics*, **16**.4, pp. 263-277 (2001).
3. Au, S. K., Beck, J. L., „First excursion probabilities for linear systems by very efficient importance sampling", *Probabilistic Engineering Mechanics*, **16**, pp. 193-207, (2001).
4. Zimmermann, M., Edler von Hoessle, J., "Computing solution spaces for robust design," *International Journal for Numerical Methods in Engineering*, **94**:290-307, (2001).
5. *Project Management Body of Knowledge (PMBOK Guide)*, Project Management Institute, Newtown Square, PA. (2013).
6. Löbl, D., Holzapfel, F., "Model Based Online Monitoring of Uncertain Plants Subject to Stochastic Disturbances", *AIAA Atmospheric Flight Mechanics Conference 2016*, Washington, DC (2016).
7. U.S. Military Handbook MIL-HDBK-1797 – *Flying Qualities of Piloted Aircraft* (1997).

8. Hanke, C. R., Nordwall, D. R., „The Simulation of a Jumbo Jet Transport Aircraft, Volume II: Modeling Data," D6-30643, The Boeing Company, KS (1970).
9. Löbl, D., Holzapfel, F., "Simulation Analysis of a Sensor Data Fusion for Close Formation Flight," *AIAA Guidanec, Navigation, and Control Conference,* National Harbor, MD (2014).
10. Wang, J., Löbl, D., Raffler, T., Holzapfel, F., "Kinematic Modeling and Control Design for an Aerial Refueling Task," *RAeS Applied Aerodynamics Conference*, Bristol, UK (2014).
11. Löbl, D., Holzapfel, F., "Subset Simulation for Estimating Small Failure Probabilities of an Aerial System Subject to Atmospheric Turbulences," *AIAA Atmospheric Flight Mechanics Conference 2015*, Kissimmee, FL (2015).