COAST - A SIMULATION AND CONTROL FRAMEWORK TO SUPPORT MULTIDISCIPLINARY OPTIMIZATION AND AIRCRAFT DESIGN WITH CPACS

D. Kiehn, J. Autenrieb, N. Fezans - DLR Institute of Flight Systems

ICAS 2022, Stockholm, Sweden, September 5th 2022

Paper ID 0505

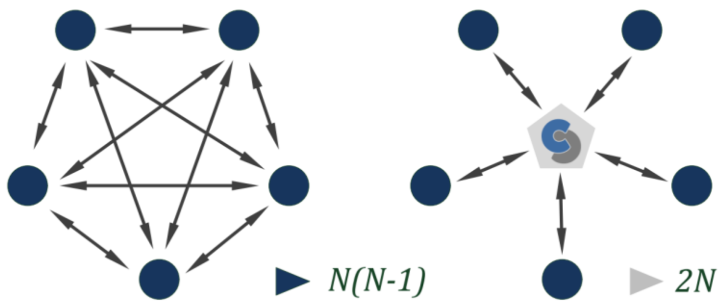
Overview

- The CPACS data exchange standard
- COAST: an aircraft simulation tailored to CPACS
- Full flight simulator AVES and its interface to COAST
- Flight control system design
- Application example
- Outlook

What is CPACS?

- <u>C</u>ommon <u>P</u>arametric <u>A</u>ircraft <u>C</u>onfiguration <u>S</u>chema
- Standardized data exchange format for MDO and aircraft design toolchains
- Focus on aircraft pre-design
- Based on XML (Extensible Markup Language)
- Developed mainly by DLR, but also international contributions
- Open source: www.cpacs.de
- Used internationally in the industry and in research

[1]



Tool to tool (decentralized) | With CPACS (centralized)

cpacs
A Common Language for Aircraft Design

[1] M. Alder, E. Moerland, J. Jepsen, B. Nagel: Recent Advances in Establishing a Common Language for Aircraft Design with CPACS. Aerospace Europe Conference, 25.-28.02.2020, Bordeaux, France.

Aircraft simulation in MDO toolchains

- Aircraft simulation in early design stages allows analysis of aircraft properties:
    - Stability and controllability
    - Flight performance, determination of the flight envelope
    - Handling qualities (MIL-STD-1797A, Cooper-Harper ratings via simulated flight tests)
    - Implications for flight control system design
    - …
- Results and insights can be fed back to the MDO design loop
- Changes of the aircraft configuration can be made (semi-) automatically
- DLR design projects also deal with unconventional aircraft and use CPACS

➔ Need for a CPACS-compatible aircraft simulation

COAST: an aircraft simulation tailored to CPACS

- CPACS-Oriented Aircraft Simulation Tool
- Aircraft simulation for configurations modelled in CPACS

- Three major components:

1) Import functions (“wrappers”): read data from CPACS, bring into required form
2) Core: 6-DoF rigid-body aircraft model
3) Toolboxes (e.g. for linearization) and interface to full flight simulator AVES

- Implementation: MATLAB/Simulink

Simulink implementation

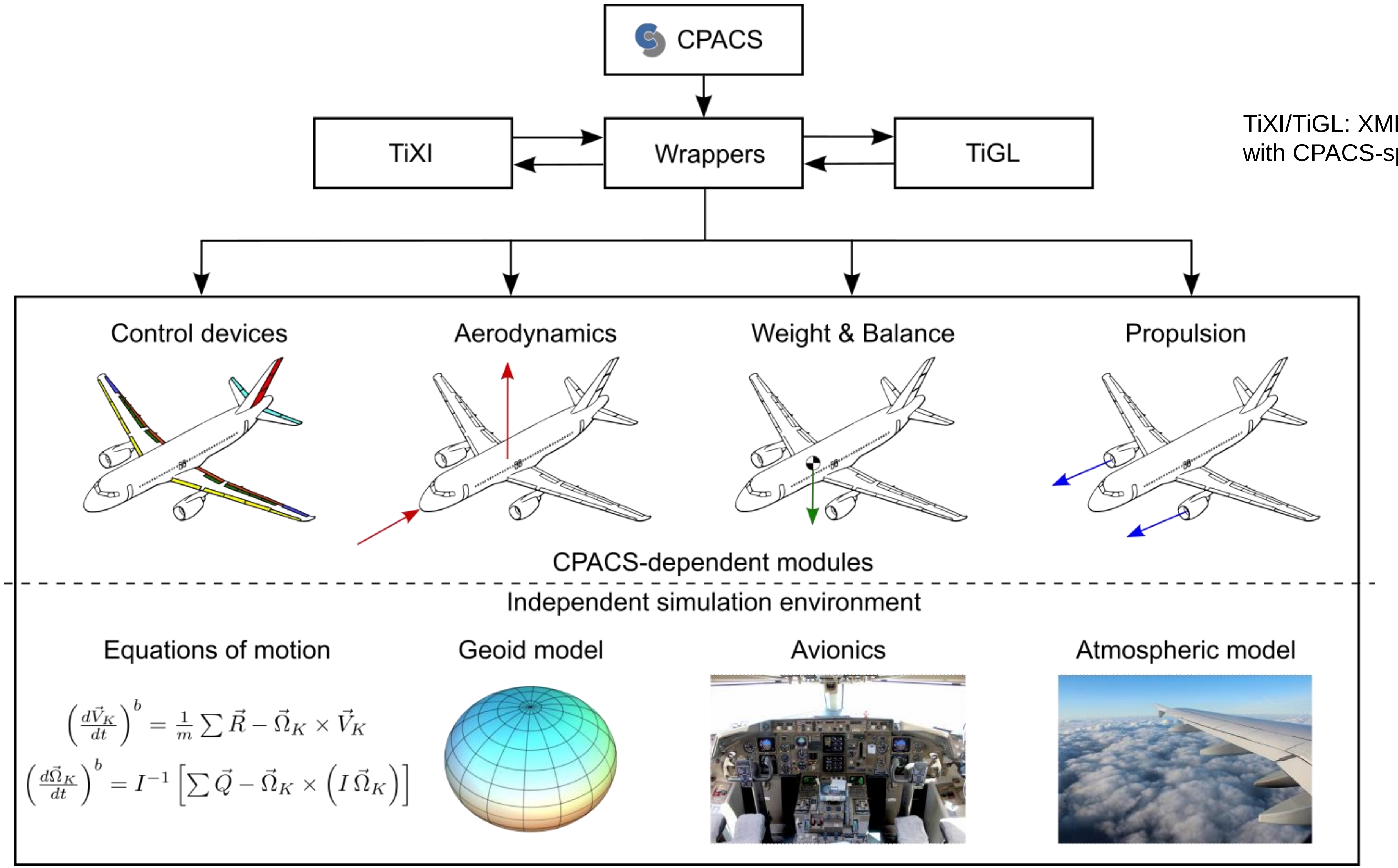
- Aircraft configurations might generally differ very significantly
  - E.g. different number of engines, number of control surfaces, …

- Implementation options in Simulink
  1) Use model with fixed structure, use available data and fill the rest with dummy data
     - No flexibility (upper limits are set), suboptimal computational performance

  2) Create model on-the-fly containing only the required components
     - Long and complex model creation process

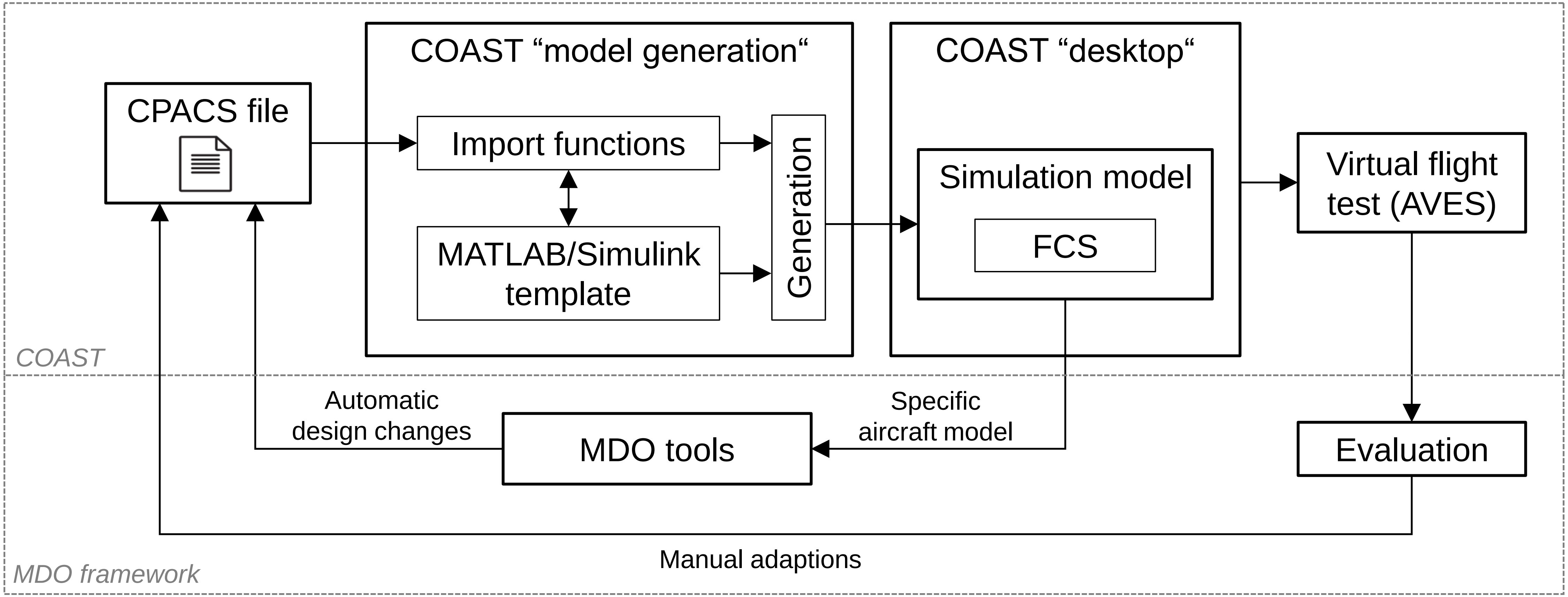
  3) Use Simulink‘s capability to provide user-written functions (so-called S-functions), in the case of COAST written in C++
     - Higher development effort, but very good flexibility and computational performance

COAST: structure and overview



TiXI/TiGL: XML parsers enhanced with CPACS-specific functionalities

COAST in an MDO toolchain



Flight control system

- COAST has to function for any given fixed-wing aircraft configuration
- DLR design projects typically consider more unconventional configurations
- Configurations generally differ significantly

➔ A universally applicable flight control system is needed to ensure basic stability and comparable handling characteristics for the pilot

Flight control system

- FCS design is based on the Normal Law of modern Airbus aircraft
- Nonlinear model following controller: response is shaped by 2nd order reference models → desired dynamics can be specified
- Pilot input commands
  - Roll axis: rate command/attitude hold (RCAH)
  - Pitch axis: vertical load factor n_z
  - Yaw axis: sideslip angle β
  - Thrust lever: speed command
- Control surface deflections are obtained by a control allocation module
  - Inversion of the nonlinear aircraft dynamics
  - Individual control characteristics are fully compensated for by the control allocation

  → Control system can be implemented almost independently of aircraft configuration

DLR’s full flight simulator AVES

- AVES = Air Vehicle Simulator
- Full flight simulator
- Electropneumatic hexapod motion system
- Motion and fixed platform with exchangeable cockpits
- Available cockpits: Airbus A320, Eurocopter EC135
- Visual dome with 240° x 95° field of view
- Located at DLR Braunschweig
- Essential platform for simulated flight tests

COAST interface to A320 cockpit

- Interface has been established between COAST and AVES with A320 cockpit
  - This allows simulated flight tests of configurations in early design stages

- CPACS configurations may represent very different aircraft, but simulator cockpit represents the A320

- Limited representation/compatibility of the pilot interface:
  - Simplified Primary Flight Display (PFD)
  - Simplified navigation page
  - Thrust levers control all left/right engines respectively (in direct law)
  - …

Example: simulated flight tests in the SynergIE project

[2]

- DLR project SynergIE: distributed electric propulsion
- Model based on COAST, slightly extended
- Simulated flight tests in AVES revealed [2]:
  - Strong effects of propeller slipstream on the lift distribution
  - Some “helicopter-like” flying characteristics: thrust increase leads to strong lift increase → to accelerate, thrust increase and significant pitch down (lower AoA) are needed
  - Effects were so severe that the aircraft was deemed almost uncontrollable without FCS, at least with full flaps

[2] D. Vechtel and J.-P. Buch: Aspects of Yaw Control Design of an Aircraft with Distributed Electric Propulsion. CEAS Aeronaut J (2022). doi: 10.1007/s13272-022-00595-1

Outlook

- Implementation of a generic but realistic landing gear model
  - Required for take-off and landing simulations
  - Trade-off: flexibility of implementation vs. degree of realism
- Improvement of the flight control system
  - More automatic tuning of gains
  - Better transition between Normal Law and Direct Law
- Implementation of the non-CPACS-dependent model components in C++ S-functions → performance gain on desktop computers

Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)
German Aerospace Center
Institute of Flight Systems | Department of Flight Dynamics & Simulation | Lilienthalplatz 7 | 38108 Braunschweig | Germany

Daniel Kiehn, M.Sc.
phone +49 531 295-2959
daniel.kiehn@dlr.de

THANK YOU VERY MUCH FOR YOUR ATTENTION!