FRICO: An AI-Enabled Friendly Cockpit Assistance System

Prakash Jamakatel, Sondes Morchedi, Jane Jean Kiam

Institute for Flight Systems
Universität der Bundeswehr München
Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany

Abstract

Single Pilot Operations (SPOs) is an emerging trend in aviation, with the potential to reduce crew and increase operational efficiency. However, SPOs present a higher risk due to the lack of redundancy; as demonstrated by historical accident statistics in single-pilot ultralight aircraft. To mitigate the risks associated with SPOs, we develop "FRICO", a FRIendly COckpit assistance system. FRICO is an AI-enabled assistance system that uses planning techniques to generate flight paths and provides guidance i.e. best course of actions to pilots while performing cockpit tasks in nominal situations, and even in non-nominal situations, while continuously monitoring the pilot, making SPOs safer.

Introduction

Accidents induced by human-factors occur more frequently in general aviation than in commercial or military aviation (De Voogt et al. 2018), primarily due to the less rigorous licensing procedures and limited technological assistance for pilots in the cockpit. To mitigate the pilot's errors by providing contextual help, future cockpits will be equipped with assistance systems, capable of complementing the pilot's skills according to the flight situation, by intervening if required. This will also pave the way for Single Pilot Operations (SPOs). Previous work in cockpit assistance systems such as CHAP-E (Benton et al. 2018) can provide guidance on what action to perform based on the aircraft's and automation state, however does not consider human factors.

In this paper, we introduce an assistance system FRICO, that uses AI-planning techniques to deliver reproducible, and explainable assistance, using open-source tools. Besides providing guidance to the (single) pilot, FRICO continuously monitors the pilot's actions, to account for human-factors. First, a brief overview of the system is showcased, followed by a short description of the techniques used in building the sub-functions for flight guidance, flight path planning and plan recognition.

FRICO System Overview

Our cockpit assistance system, FRICO, consists of three core modules (Figure 1): a Plan Generation Module (PGM), a Plan Recognition Module (PRM) and an Action Recognition Module (ARM). The PGM can be used to generate instructions (i.e., primitive tasks) to be executed by the pilot and to generate a flight trajectory between two designated coordinates. The ARM module monitors the pilot continuously to identify the action being performed, while the PRM recognises the pilot's intention (i.e. intended plan). The modules are described in the subsequent sections.

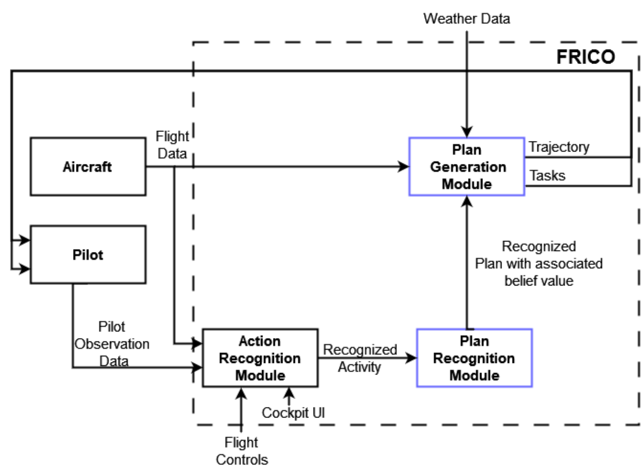

Figure 1: System Architecture of FRICO

Plan Generation Module

The instructions for performing flight procedures are hierarchical in nature and assume domain-level knowledge from the pilot.Hierarchical Domain Definition Language (HDDL) formalism from Geier and Bercher (2011) is used to model the cockpit-tasks as Hierarchical Task Network (HTN) using domain-knowledge from the pilot manuals, e.g. (SHARK 2017), while the PANDA framework (Höller et al. 2021) is used to generate list of partially or totaly ordered tasks to be performed by the pilot, upon request.

The PGM also contains the function to generate flight trajectories to guide the aircraft from a current position to a goal position while taking into consideration the weather information at planning time and the current state of the aircraft, including the health of the aircraft (e.g. fuel level, defects in the wings, etc.). Given that the flight trajectory planning problem is numeric in nature; the hybrid planner ENHSP (Scala et al. 2016) proves to be a suitable choice (León, Kiam, and Schulte 2020). The aircraft's physical model is encoded into the problem domain, while the weather conditions and the current state are encoded in the problem instance (Kiam et al. 2020). The generated plan is temporal, and provides control parameters such as turn rate and climb rate, while also considering standard patterns for different flight legs, and can be executed by the pilot or the autopilot(should one be onboard).

Plan Recognition Module

Hierarchical plan recognition as planning[1] (Höller et al. 2018) is used to implement the PRM, which takes a sequence of observations (or rather detected actions) as input and generates a single shortest-solution plan that best describes the given observations. Actions are low-level interactions between the pilot with the aircraft or the surrounding, which results in a state-change. Interactions like pilot's fixation points in cockpit, interaction with UI are combined for action inference. Dempster-Shafer Theory for action inference (Honecker and Schulte 2017) is used, and can account for human factors while performing action.

Technical Implementation

FRICO consists of several submodules, each of which has to process large amounts of data, and requires synchronized communication channels due to their interdependence. We use the Aerolite 103 aircraft model from X-Plane, which is a widely used flight simulator with realistic flight dynamics and an immersive simulation environment. AirManager [2] is used to render the flight instruments in FRICO's UI (Figure 2). FRICO takes into account weather information extracted from NOAA[3] when generating the flight trajectory. The map is rendered using ArcGIS, and contains topological information, aerial photographs, etc. that are available on BayernAtlas[4], an open-access source for geo-referenced data. The ARM module receives simulated flight data from X-Plane, which are communicated using ExtPlane[5]. The flight controls data, such as the throttle and yoke, are first sent to X-Plane and then read through ExtPlane. Pilot fixation data is generated using Pupil Core[6], and are communicated to the ARM module using a ZMQ publisher-subscriber pattern[7]. The PRM and ARM are integrated into a single C++ QT-program, the detected actions are communicated with the PRM using built-in signal and slots[8] mechanism. The same mechanism is used to relay the flight plan generated by the ENSHP planner to FRICO. The detected list of actions is saved in an external file and PRM invokes an external bash process to communicate with the PGR[1] to recognise the plans.

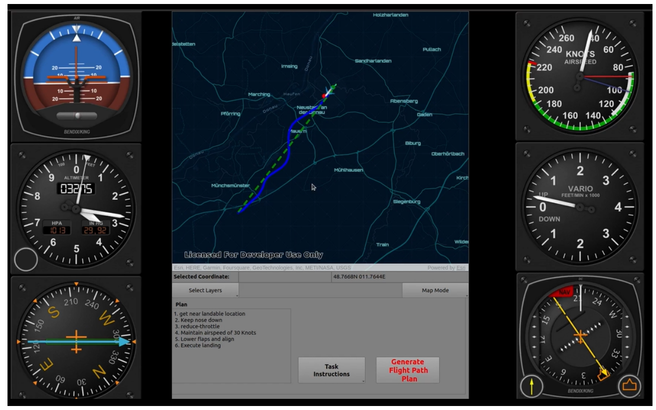

Figure 2: FRICO's UI with generated flight path and instructions

[1]https://github.com/panda-planner-dev/pandaPIpgrRepairVerify

[2]https://siminnovations.com/wiki/index.php

[3]https://www.noaa.gov/weather

[4]https://geoportal.bayern.de/bayernatlas/

[5]https://github.com/vranki/ExtPlane

[6]https://pupil-labs.com/products/core/

[7]https://zguide.zeromq.org/docs/chapter5/

[8]https://doc.qt.io/qt-6/signalsandslots.html

Acknowledgments

This work is funded by the German Federal Ministry of Economic Affairs and Climate Action (Project MOREALIS) and by Munich Aerospace (Research group "Intelligent Control of Highly Over-Actuated Flight Systems").

References

Benton, J.; Smith, D. E.; Kaneshige, J. T.; Keely, L.; and Stucky, T. 2018. CHAP-E: A Plan Execution Assistant for Pilots. In *ICAPS*.

De Voogt, A.; Chaves, F.; Harden, E.; Silvestre, M.; and Gamboa, P. 2018. Ultralight Accidents in the US, UK, and Portugal. *Safety*.

Geier, T.; and Bercher, P. 2011. On the decidability of HTN planning with task insertion. In *IJCAI*.

Höller, D.; Behnke, G.; Bercher, P.; and Biundo, S. 2018. Plan and goal recognition as HTN planning. In *ICTAI*.

Höller, D.; Behnke, G.; Bercher, P.; and Biundo, S. 2021. The PANDA framework for hierarchical planning. *KI-Künstliche Intelligenz*.

Honecker, F.; and Schulte, A. 2017. Automated online determination of pilot activity under uncertainty by using evidential reasoning. In *14th Engineering Psychology and Cognitive Ergonomics: Cognition and Design*.

Kiam, J. J.; Scala, E.; Javega, M. R.; and Schulte, A. 2020. An AI-based planning framework for HAPS in a time-varying environment. In *ICAPS*.

León, B. S.; Kiam, J. J.; and Schulte, A. 2020. Model-Based Automated Flight Path Planning for an Ultralight Aircraft. In *IPS-RCRA*, CEUR Workshop Proceedings.

Scala, E.; Haslum, P.; Thiébaux, S.; and Ramirez, M. 2016. Interval-based relaxation for general numeric planning. In *ECAI*.

SHARK. 2017. *Flight Manual: UL airplane*. SHARK.AERO CZ s.r.o.