Flap/Slat System

Astrid Rakow

July 21, 2011

Abstract

The Flap/Slat Controller benchmark is derived from a case study for Airbus [1]. During take-off (and landing) flaps and slats of the aircraft are extended to generate more lift at low velocity and have to be retracted in time as they are not robust enough for high velocities. Our system model is scalable in the number of flaps/slats, which allows us to study scalability of our methods on systems of concurrent components.

1 Application Domain

Flaps and slats are aerodynamic surfaces on respectively the trailing and leading edge of an aircraft's wings. During take-off and landing, these flaps and slats are extended to generate more lift at low velocity and have to be retracted in time as they are not robust enough for high velocities. They are under the primary control of the pilot via a Slat/Flap Lever, where lever positions correspond to flap/slat positions. A Flap/Slat Controller constantly monitors aircraft speed, whether the load of a flap/slat for the pilot commanded lever position is too high. The controller retracts flaps/slats from the extended position to the next lower lever setting, in case the aircraft speed exceeds a certain threshold speed for the selected configuration for more than a specified limit. If the aircraft speed falls below the threshold speed, the flap/slat position selected by the pilot will be automatically be commanded again.

2 HLang Models

In the following a family of HLANG models is described scalable in the number n of flaps/slats each having its own controller monitoring whether the respective flap/slat position needs to be corrected. This system is used to study scalability of our methods for systems of concurrent components.

2.1 Variables in the Model

The HLang description of our flap/slat system having k flaps and l slats has the following variables.

`boolean` C_flap_i	`true` if a correction has to be applied on flap i, $i \leq k$
`boolean` C_slat_i	`true` if a correction has to be applied on slat i, $i \leq l$
`boolean` pilot_done	`true` if the pilot has chosen a lever position
`boolean` automode_done	`true` when the controllers determined whether a correction has to be applied
`boolean` set_modes_done	`true` when for the flaps/slats the movement {KEEPING, INCREASING, DECREASING} has been determined.
`int` lever_pos	lever position
`int` md_flap_i_change	encoding the current movement of flap i; it represents the modes {KEEPING, INCREASING, DECREASING}
`int` md_slat_i_change	encoding the current movement of slat i; it represents the modes {KEEPING, INCREASING, DECREASING}
`real` plane_velocity	velocity of the aircraft
`real` flap_angle_i	angle of the i-th flap
`real` slat_angle_i	angle of the i-th slat

where $i <= k + l = n$.

2.2 Discrete Computations

The discrete computations are done in three consecutive steps:

1. The pilot may chose a new lever position.
2. The controllers decide on whether a correction has to be applied.
3. The appropriate movement for each flap/slat is determined.

Therefore boolean variables have been introduced, encoding the completion of the three steps. Each step entails parallel execution for flap/slat controllers.

2.3 Examined Safety Property

The safety property to be established for our model(s) is that the aircraft's velocity never exceeds the allowed velocities for the current flap/slat positions, respectively.

3 Experiments

To show that the examined model satisfies the safety property from above, we applied

1. the standard model checking technique of our tool FOMC (`fomc`) and
2. also with boolean state set abstraction (`bool abs`),
3. our novel technique targeting systems where many components evolve in parallel (`acc at once`) and
4. this technique combined with boolean state set abstraction (`bool abs + acc at once`).

Figure 1 shows the measured run times on the y-axis in logarithmic scale and on the x-axis model instances scaling up the number of flaps/slats to the right. On this model all approaches except our novel technique show an exponential increase in the runtime. The best results on this benchmark are achieved by the combination of boolean state set abstraction in combination with the new technique.

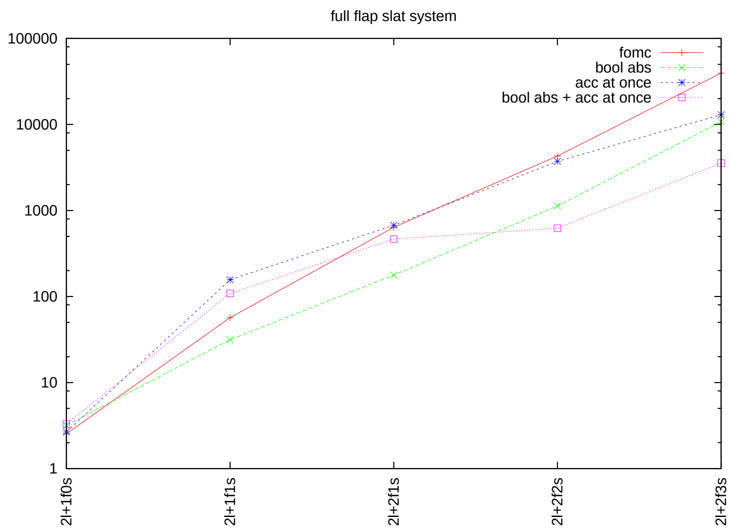

Figure 1: Run times measured in sec for scaled up instances of the Flap/Slat controller benchmark

References

[1] H3FOMCTeam, The flap controller description, 2010, `http://www.avacs.org/fileadmin/Benchmarks/Open/flapcontroller.pdf`.