DMAC: Deadline-Miss-Aware Control (Artifact)

Paolo Pazzaglia
Scuola Superiore Sant'Anna, Pisa, Italy
Department of Automatic Control, Lund University, Sweden
paolo.pazzaglia@sssup.it

Claudio Mandrioli
Department of Automatic Control, Lund University, Sweden
claudio.mandrioli@control.lth.se

Martina Maggio
Department of Automatic Control, Lund University, Sweden
martina.maggio@control.lth.se

Anton Cervin
Department of Automatic Control, Lund University, Sweden
anton.cervin@control.lth.se

Abstract

The real-time implementation of periodic controllers requires solving a co-design problem, in which the choice of the controller sampling period is a crucial element. Classic design techniques limit the period exploration to *safe* values, that guarantee the correct execution of the controller alongside the remaining real-time load, i.e., ensuring that the controller worst-case response time does not exceed its deadline. This paper presents the artifact linked to DMAC: the first formally-grounded controller design strategy that explores shorter periods, thus explicitly taking into account the possibility of missing deadlines. The experimental results obtained with this artifact show that the DMAC design proposal – i.e., exploring the space where deadlines can be missed and handled with different strategies – greatly outperforms classical control design techniques.

2012 ACM Subject Classification Computing methodologies → Computational control theory; Computer systems organization → Embedded software; Software and its engineering → Real-time systems software; Theory of computation → Stochastic control and optimization

Keywords and phrases Weakly-Hard Real-Time Systems, Deadline Miss Handling, Control Design

Digital Object Identifier 10.4230/DARTS.5.1.3

Funding This work was partially supported by the ELLIIT Strategic Research Area and by WASP (Wallenberg AI, Autonomous Systems and Software Program) funded by Knut and Alice Wallenberg Foundation.

Related Article Paolo Pazzaglia, Claudio Mandrioli, Martina Maggio, and Anton Cervin, "DMAC: Deadline-Miss-Aware Control", in 31st Euromicro Conference on Real-Time Systems (ECRTS 2019), LIPIcs, Vol. 133, pp. 1:1–1:24, 2019.
`https://dx.doi.org/10.4230/LIPIcs.ECRTS.2019.1`

Related Conference 31st Euromicro Conference on Real-Time Systems (ECRTS 2019), July 9–12, 2019, Stuttgart, Germany

1 Scope

The DMAC design leverages information about the probability that specific sub-sequences of deadline misses are experienced. The result is a *fixed* controller that on average works as the ideal clairvoyant time-varying controller that knows future deadline hits and misses. The design is based on a safe estimate of the hit and miss events, obtained using the *scenario theory*. This step allows us to provide probabilistic guarantees on the validity of the randomly generated simulations the design is based on. The related paper analyzes controllers implemented using the Logical

Execution Time paradigm and three different strategies to handle deadline miss events: killing the job, letting the job continue but skipping the next activation, and letting the job continue using a limited queue of jobs. The artifact provides the code necessary for the control design and comparison allowing the user to gather experimental evidence of the DMAC design soundness.

2 Content

The artifact package has the following tree structure:

- figures
  - fig7.tex
  - fig8.tex
  - fig9.tex
- src
  - matlab
  - scenario
  - sim
- tasksets
- generate_results.sh
- process_results.sh
- plot_results.sh
- README

The README file contains instructions for the execution of the artifact. Three scripts (generate_results.sh,process_results.sh, plot_results.sh) are used for data generation, processing, and plotting. The tasksets folder contains the set of tasks that are analyzed (plus additional ones that can be tested). The src folder contains the code for the artifact. The figures folder includes the tex files needed for the generation of the figures shown in the paper.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS).

4 Tested platforms

The code is better run in a Unix platform. Specifically, the code has been tested on both Linux (Fedora 25) and MAC OS X. It is possible to run the code on Windows machines, but the instructions included in the artifact do not apply (automation via shell scripting does not work, so the corresponding steps have to be manually executed).

Execution requirements: a C++ compiler, MATLAB, bash shell (for the automated execution).

The code has been tested with clang++ on Mac OS and g++ on Linux and with different versions of MATLAB (from R2012 to R2016a). In MATLAB, the code uses the Control Toolbox (which implements functions like 'ss'). Finally, the code requires a full version of latex installed for the generation of the figures (that uses tikz and pgfplots).

5 License

6 MD5 sum of the artifact

92ada2d930d64f5dc3e30e7aab4f7911

7 Size of the artifact

218016 bytes