AdaptMC: A Control-Theoretic Approach for Achieving Resilience in Mixed-Criticality Systems

Alessandro Vittorio Papadopoulos, Enrico Bini, Sanjoy Baruah, and Alan Burns

March 24, 2018

1 Introduction

The following figures of the paper include a numerical result that can be reproduced by this artifact:

- Figure 2: Region of feasible control gains. The illustrated regions correspond to the values of $K_i \in \{0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.35\}$, respectively from the larger region to the smaller one. Black dots represent the gains of the controllers selected for the examples illustrated in Section 6.
- Figure 5: Supply functions for the considered set of control parameters.
- Figure 6: Effect of constant disturbances with various selection of K_{ij} , $i, j \in \{\mathrm{H}, \mathrm{L}\}$.
- Figure 7: Comparison between AdaptMC and PPA.

The remaining figure do not contain any numerical result, and they support the description of the paper as illustrations.

2 Requirements

We implemented the results presented in the paper by using Matlab, using mostly basic functions, and few functions included in the Control System Toolbox. We used Wolfram Mathematica to symbolically analyze, and compute the stability region presented in Figure 2. In particular, the following are the main requirements:

- Matlab R2016a or higher: We tested the codes on Matlab R2016a, R2016b, R2017a, but the lower is not confirmed.
- Control System Toolbox of Matlab.
- For Figure 2, the symbolic expressions are obtained in Matlab, but the regions have been obtained with Wolfram Mathematica. In order to reproduce such graph the following **alternatives** can be used:
  - Wolfram Mathamatica 11 (license needed).
  - Wolfram CDF player 11.3 (`https://www.wolfram.com/cdf-player/`), no need of any license.

3 List of Contents

Table 1: List of the functions included in the artifact, and their respective description.

Function	Figure	Description
`adaptMC_sim.m`	–	Simulate AdaptMC over time.
`computeConv.m`	–	Auxiliary function for computing the convolution of two signals.
`computeJI.m`	–	Auxiliary function for computing $\mathcal{J}(n)$ and $\mathcal{I}(n)$ terms.
`computeJInL.m`	–	Auxiliary function for computing $\mathcal{J}(n)$ and $\mathcal{I}(n)$ terms.
`computeNnL.m`	–	Auxiliary function for computing $\mathcal{N}(n)$ term.
`csvwrite_with_headers.m`	–	Function for saving the raw data in CSV format with a header.

generateFigure5.m	5	Generates Figure 5.
generateFigure6.m	6	Generates Figure 6.
generateFigure7.m	7	Generates Figure 7.
imp.m	–	Compute the impulse function at a generic time t.
jury.m	–	Generates the Jury table for the assessment of the stability of a Linear-Time Invariant System.
plottingResults_comparison.m	–	Auxiliary function for plotting results of Figure 7.
PPA_sim.m	–	Simulate PPA over time.
ram.m	–	Compute the ramp function at a generic time t.
rampResp.m	–	Compute the ramp response of for AdaptMC.
stability.m	–	Generates the stability conditions for AdaptMC.
StabilityRegion.cdf	2	Generates Figure 2 (Wolfram CDF player required).
StabilityRegion.nb	2	Generates Figure 2 (Wolfram Mathematica required).
stepResp.m	–	Compute the step response of for AdaptMC.
stp.m	–	Compute the step function at a generic time t.
supplyFunH.m	–	Compute the supply bound function for the HI-Criticality server.
supplyFunL.m	–	Compute the supply bound function for the LO-Criticality server.

4 Generate Figure 2

Typical time needed for the execution: about 1–2 seconds.

The generation of Figure 2 requires the execution of the following code, contained in the `stability.m` file.

```
clear; clc;
% Definition of symbolic variables
syms Khh Khl Kll Klh real
syms z

% Dynamic matrix of AdaptMC
A = [zeros(2,2), eye(2);
     -Khh, -Khl,   1, 0;
        0, -Kll,-Klh, 1];

% Computation of the characteristic polynomial
p = charpoly(A,z);

% Extraction of the coefficient of the characteristic polynomial
c1 = coeffs(p,z);
c2 = c1(end:-1:1);  % Reordering the coefficients

% Computation of the Jury chart
[J,C] = jury(c2)
```

The variable `C` contains the coefficients "of interest", i.e., the entries of the first column with uneven row-indices. `J` contains the whole Jury chart.

Theorem 4.1. *The eigenvalues of the matrix* `A` *lie in the unit circle if and only if the coefficients in* `C` *are greater than zero. Moreover, if non of these is zero, then the number of negative entries gives the number of roots outside the unit disc.*

The `jury.m` function is not part of the standard library of Matlab, or of the Control System Toolbox, but it can be found at `https://se.mathworks.com/matlabcentral/fileexchange/13904-jury`.

Plugging the coefficients of `C` in a symbolic manipulation tool, allow the calculation. In order to obtain Figure 2, we used Wolfram Mathematica, but a Computable Document Format is also provided.

5 Generate Figure 5

Typical time needed for the execution: about 2 minutes.

In order to generate Figure 5, execute the function `generateFigure5.m`. There is a toggle variable `SAVE`, that if set to 1, saves the data generated by the script[1].

6 Generate Figure 6

Typical time needed for the execution: about 1–2 seconds.

In order to generate Figure 6, execute the function `generateFigure6.m`. There is a toggle variable `SCENARIO` that can be used to evaluate the different systems under different disturbance scenarios:

- Scenario 0: $\varepsilon_{\mathrm{H}}(t) = \mathrm{imp}(t-10)$, $\varepsilon_{\mathrm{L}}(t) = -\,\mathrm{imp}(t-50)$
- Scenario 1: $\varepsilon_{\mathrm{H}}(t) = \mathrm{step}(t-10)$, $\varepsilon_{\mathrm{L}}(t) = -\,\mathrm{step}(t-50)$
- Scenario 2: $\varepsilon_{\mathrm{H}}(t) = \min(0.1\,\mathrm{ram}(t-10), 1)$, $\varepsilon_{\mathrm{L}}(t) = -\min(0.1\,\mathrm{ram}(t-50), 1)$
- Scenario 3: $\varepsilon_{\mathrm{H}}(t) = \mathrm{imp}(t-10) + 0.1\,\mathrm{ram}(t-60)(\mathrm{step}(t-60) - \mathrm{step}(t-80))$, $\varepsilon_{\mathrm{L}}(t) = -\,\mathrm{step}(t-40)$
- Scenario 4: $\varepsilon_{\mathrm{H}}(t) = \mathrm{imp}(t-10) + \mathrm{step}(t-30) - \mathrm{step}(t-50) + \min(0.2\,\mathrm{ram}(t-65), 3)$, $\varepsilon_{\mathrm{L}}(t) = 0$
- Scenario 5: $\varepsilon_{\mathrm{H}}(t) = (-1)^t\,\mathrm{step}(t-10)$, $\varepsilon_{\mathrm{L}}(t) = (-1)^{t+1}\,\mathrm{step}(t-10)$

Scenario 1 generates Figure 6.

There is also a toggle variable `SAVE`, that if set to 1, saves the data generated by the script.

7 Generate Figure 7

Typical time needed for the execution: about 1–2 seconds.

In order to generate Figure 7, execute the function `generateFigure7.m`. There is a toggle variable `SCENARIO` that can be used to evaluate the different systems under different disturbance scenarios, with the same meaning as for Figure 6.

There is also a toggle variable `SAVE`, that if set to 1, saves the data generated by the script.

Finally, there is a toggle variable `GAIN`, that decides what is the value of the set of gains to use:

- GAIN 1: $\mathcal{K}_1 = \{0.4, 0.1, 0.1, 0.35\}$
- GAIN 2: $\mathcal{K}_2 = \{0.15, 0.1, 0.1, 0.15\}$
- GAIN 3: $\mathcal{K}_3 = \{0.25, 0.1, 0.1, 0.25\}$
- GAIN 4: $\mathcal{K}_4 = \{0.5, 0.1, 0.1, 0.5\}$
- GAIN 5: $\mathcal{K}_5 = \{0.75, 0.1, 0.1, 0.75\}$

GAIN 1 generates what is presented in Figure 7, with SCENARIO 4.

8 Contacts

- Alessandro Papadopoulos alessandro.papadopoulos@mdh.se
- Enrico Bini bini@di.unito.it
- Sanjoy Baruah baruah@wustl.edu
- Alan Burns alan.burns@york.ac.uk

[1] The script makes use of the function `csvwrite_with_headers`, available at `https://se.mathworks.com/matlabcentral/fileexchange/29933-csv-with-column-headers`