

Verifying Switched System Stability With Logic (Repeatability Evaluation)

1 Introduction

This artifact is packaged as a script that sets up a server instance of the KeYmaera X theorem prover (and all of its prerequisites) within a Docker container. You will then interact with the KeYmaera X server’s user interface through a web browser on your host machine. You can also read up more about the KeYmaera X syntax and informal semantics, as well as about the main features of the KeYmaera X user interface in the KeYmaera X tutorial directly at

<https://keymaerax.org/Xtutorial.html>

The artifact is available at the following link:

<https://figshare.com/s/00b273eb0a5fc61c175d>

md5sum: c99ddc69b73cf44b4f8a03d775aade52 artifact.zip (updated 8 Feb 2022)

The following files and folders are included in this artifact:

- `dockersetup.sh` The setup script for creating a Docker container.
- `Dockerfile` The Dockerfile used by `dockersetup.sh`.
- `dockerrun.sh` The script for starting the KeYmaera X server inside Docker.
- `dockercheck.sh` The script for running a quick check of all proofs in this artifact.
- `paper.pdf` A copy of the submitted paper to HSCC’22 (henceforth referred to as “the paper”).
- `models/examples` The examples (`example1.kyx`–`example10.kyx`) used in Section 4 of the paper.
- `models/canonmaxsys` The case study in Section 5.1 of the paper.
- `models/cruisectl` The case study in Section 5.2 of the paper.
- `models/nonholo` The case study in Section 5.3 of the paper.

The artifact is designed for the reviewers to reproduce the claims in Sections 4 and 5 of the paper:

- Section 4.1: KeYmaera X modeling and proof interface for switched systems (§ 4).
- Section 4.2: Examples, specifically, Table 2 (§ 5).
- Section 5: Case studies (§ 6).

1.1 Updates

The following updates were made based on HSCC'22 Repeatability Evaluation anonymous reviewer feedback.

- There may be an intermittent problem downloading MATLAB. Unfortunately, this seems to be an issue with the MathWorks download servers and is out of our control. We suggest re-running the `dockersetup.sh` script to attempt the download again. If the download still fails, most of our package can be evaluated without the use of MATLAB. For this purpose, we have also updated the package file with an additional folder `no_matlab/` with files that will NOT attempt to download or install packages related to MATLAB. Please run the following commands in the unzipped folder:

```
mv no_matlab/Dockerfile ./Dockerfile
mv no_matlab/dockersetup.sh ./dockersetup.sh
... follow instructions for evaluation WITHOUT MATLAB ...
```

- We have now included a precompiled `keymaerax.jar` file in the package based on the commit used in the `Dockerfile` which can be used locally if needed (you will need to install WolframEngine locally). In addition, for the quick evaluation in § 3, `dockercheck.sh` is configured to use the Docker container by default so you will not be able to run it directly. To run the quick check, please remove the docker commands and instead run the local `keymaerax.jar` file directly, e.g.:

```
java -da -jar keymaerax.jar -prove models/examples/example1.kyx -out proof.kyp
```

2 Setup and Installation

The following instructions assume that Docker is installed on your machine and working correctly, see <https://docs.docker.com/get-docker/>. We recommend having at least 8–16 GB RAM on your machine to avoid compilation issues when building the Docker container. In addition, we recommend configuring Docker to create containers with at least 50 GB of disk space. The instructions have been tested on the following host systems:

Ubuntu 20.04, 16 GB RAM, Docker version 20.10.12, build e91ed57
(with MATLAB license)

Ubuntu 18.04, 16 GB RAM, Docker version 20.10.7, build 20.10.7-0ubuntu5~18.04.3
(without MATLAB license)

MacOS 11.6.1, 16 GB RAM, Docker version 20.10.12, build e91ed57
(with MATLAB license)

The installation script uses the following software versions:

WolframEngine 13.0
MATLAB r2021b
KeYmaera X 4.9.8 c877dd070548b2dec7ceb7c27a17d2e90f5ffe16

For a complete evaluation, our tool requires activated versions of MATLAB and WolframEngine. The license for WolframEngine can be obtained for free and you will be prompted during installation to do so. The setup script will attempt to reuse a MATLAB license belonging to you. **If you do not have a MATLAB license, most of the evaluation steps will work correctly, except those that use MATLAB to auto-generate Lyapunov functions. These are marked explicitly below with (Requires MATLAB) and you may safely skip them.**

2.1 Quick Installation Guide

The following guide gives minimal steps for the installation process. Refer to the full guide further below for detailed steps and troubleshooting instructions.

1. Find the path to your MATLAB license, see <https://www.mathworks.com/matlabcentral/answers/99147-where-are-the-license-files-for-matlab-located>
2. Run the setup script with the command (other options are detailed in the full instructions, including if you do not have a MATLAB license):

```
./dockersetup.sh -l /path/to/matlab.lic
```

Please note that the installation process may take a long time to complete because it installs both MATLAB and WolframEngine (even if you do not have a license for MATLAB).

3. During installation, you will see a dialog asking for one-time activation of WolframEngine. Please register for a free WolframEngine license:

The Wolfram Engine requires one-time activation on this computer.

Visit <https://wolfram.com/engine/free-license> to get your free license.

Wolfram ID:

We recommend you register for a license with a new Wolfram account (see full instructions for more details).

4. Run the following script to start the KeYmaera X docker container.

```
./dockerrun.sh
```

5. Open a browser on your **host machine** and browse to `http://localhost:8090`. Select **Industry Mode**, **Next**, and accept the KeYmaera X license.
6. You should now be on KeYmaera X's Models page and ready for evaluation. Click on the power button icon in the top-right corner to shut down KeYmaera X and the Docker container.

Please start with the instructions in §3 and §4 before trying the subsequent sections.

2.2 Full Installation Guide

1. Find the path to your MATLAB license, see <https://www.mathworks.com/matlabcentral/answers/99147-where-are-the-license-files-for-matlab-located>. Skip this step if you do not have a MATLAB license.
2. Run the setup script with the command (other options are detailed in `dockersetup.sh` and below):

```
./dockersetup.sh -l /path/to/matlab.lic
```

The step will attempt to guess a default username and network interface MAC address tied to your MATLAB license. If your license is tied to a different username or MAC address, please add it to the command manually as follows:

```
./dockersetup.sh -l /path/to/matlab.lic -u USERNAME -m MAC_ADDRESS
```

If you do not have a license for MATLAB, simply omit the `-l` option. The script will still download MATLAB but it will not be activated.

```
./dockersetup.sh -u USERNAME -m MAC_ADDRESS
```

Please note that the installation process may take a long time to complete because it installs both MATLAB and WolframEngine (even if you do not have a license for MATLAB).

3. You should see output like the following:

```

Running on Linux
Using Matlab license tied to:
YOUR_USERNAME
MAC_ADDRESS
/path/to/matlab.lic
kx
Sending build context to Docker daemon 44.54kB
Step 1/40 : ARG MATLAB_VERSION=r2021b
Step 2/40 : FROM mathworks/matlab-deps:${MATLAB_VERSION}
--> 2920e5c09e4b
...
Step 40/41 : ADD ./models/ models/
--> 7118828fb4e7
Step 41/41 : WORKDIR /home/${USER_NAME}/
--> Running in f8cded2a2754
Removing intermediate container f8cded2a2754
--> 9b3876795a0c
Successfully built 9b3876795a0c
Successfully tagged keymaerax:latest
82e73402a75d5ad2ef0d73adc2ef475c298572dc5964bb977711eefa5d6c8278
kx
...

```

4. During installation, you will see a dialog asking for one-time activation of WolframEngine:

The Wolfram Engine requires one-time activation on this computer.

Visit <https://wolfram.com/engine/free-license> to get your free license.

Wolfram ID:

Please follow the instructions to register a Wolfram account for a free WolframEngine license and then enter your Wolfram ID and password. The activation process requires that the Docker guest image have access to the host Licensing subfolder where you executed the setup script.

You can use an existing Wolfram account to register. However, please note that Wolfram may lock your WolframEngine license if it has been used to register too many different machines or Docker containers. Accordingly, if you have successfully activated WolframEngine before and you need to run the setup script again, please skip license activation by pressing Ctrl-D.

After activating WolframEngine, you should see the following final lines of output

```
KeYmaera X Prover 4.9.8
```

```
Use option -help for usage and license information
Initializing lemma cache...
...done
kx
```

If the script fails midway, please try to re-run it a few times to see if it makes further progress. Specifically, if the script fails in compilation with a Java Heap Space error, please try to increase memory for the Docker container and/or increase the SBT heap space in Dockerfile line 118 (-Xmx4G) and re-run. The artifact was tested with 16GB of memory available to the Docker container and 4GB of SBT heap space.

If the script fails when installing MATLAB or WolframEngine with error messages similar to “Error: Download failed. Check the network connection and retry.”, or “ZIP decompression failed”, please check your internet connection and/or try increasing the amount of disk space available to the Docker container and/or simply retry since the MATLAB and WolframEngine download sites may temporarily be unavailable.

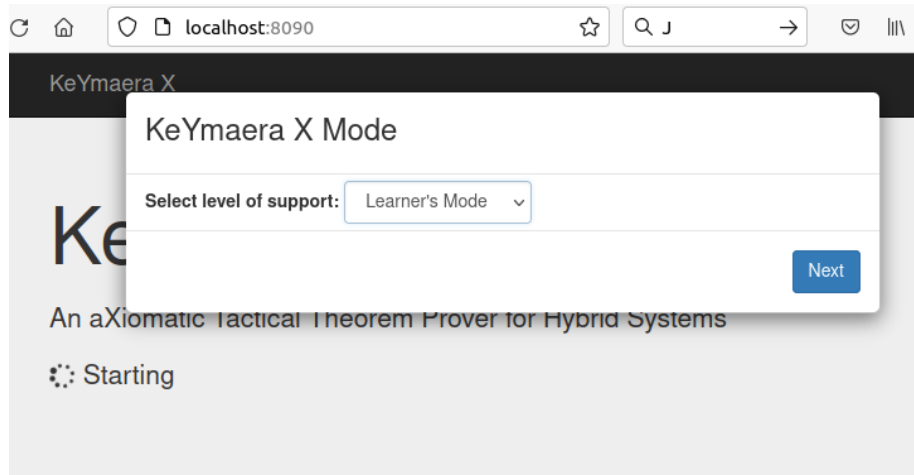
5. Run the following script to start the KeYmaera X docker container.

```
./dockerrun.sh
```

You should see output like the following:

```
kx
[launcher] Database version: 4.8.0
Point your browser to http://localhost:8090
```

6. Open a browser on your **host machine** and browse to `http://localhost:8090`. You should see KeYmaera X’s starting page (shown in the following screenshot). Select **Industry Mode**, **Next**, and accept the license.



7. You should now be on KeYmaera X's Models page and ready for evaluation. Click on the power button icon in the top-right corner to shut down KeYmaera X and the Docker container.

Please start with the instructions in §3 and §4 before trying the subsequent sections.

3 Evaluation: Quick Evaluation

For convenience, we have provided a simple script that checks the proofs in Sections 4 and 5 of the paper without using the KeYmaera X UI. We recommend running this script for a quick check before trying the more detailed evaluation steps.

1. Run the following command:

```
./dockercheck.sh
```

2. You should see output like the following:

```
kyn
...

Proving entries from 1 files
Proving .../example1.kyn#Examples/Example 1: stability ...
Done .../example1.kyn#Examples/Example 1: stability (proved)
Proving .../example1.kyn#Examples/Example 1: attractivity ...
Done .../example1.kyn#Examples/Example 1: attractivity (proved)
PROVED Examples/Example 1: stability: tactic=..., \
```

```
tacticsize=1,budget=0[s],duration=3914[ms],qe=782[ms],rcf=0,steps=3
PROVED Examples/Example 1: attractivity: tactic=..., \
tacticsize=1,budget=0[s],duration=3979[ms],qe=1898[ms],rcf=0,steps=5
...
```

The script should prove stability and attractivity for all examples and case studies (**PROVED** is shown in the output) except the following:

- Examples 5 and 10, which do not have a proof, see §5 for more details.
- Canonical max system case study, attractivity in case $f \neq 0$. This proof is slow and works better in the KeYmaera X UI, see §6 for instructions on running this proof in the UI.

4 Evaluation: Interface for Switched Systems

This section provides instructions for using the KeYmaera X switched systems modeling and proof interface. Follow the basic setup instructions (§2) to start KeYmaera X and open the UI in a browser.

1. On the Models page and click on:

```
Templates -> Switched Systems -> New Model
```

This will bring up a modeling interface similar to the one shown in Figure 3 of the paper.

2. The textbox under **Switching: Autonomous** allows users to write switching modes in a graph-like language. Modes are specified with the following format: `MODE_NAME ("ODE & DOMAIN")`. For example, try:

```
A("x'=-x^3 & x <= 5")
B("x'=-x")
A & B --> A & B
```

This yields a system that switches between modes A, B, where mode A has ODE $x' = -x^3$ in domain $x \leq 5$ and mode B has ODE $x' = -x$ with no domain constraint.

The last line `A & B --> A & B` is not strictly necessary for this example but it helps with visualization panel on the right.

3. In **Specification** there are three clickable tabs **Stability**, **Attractivity**, and **Custom**. These tabs autogenerate the relevant specifications from the switched systems. Uniform Global pre-Asymptotic Stability (UGpAS), as studied in the paper, is specified by selecting both **Stability** and **Attractivity** tabs. For example, the following abbreviated snippet is autogenerated for the above example after selecting **Stability** and **Attractivity**:

```

1 ArchiveEntry "New Entry: stability"
2 /*
3  * Generated from hybrid automaton
4  * A("x'=-x^3 & x <= 5")
5  * B("x'=-x")
6  */
7
8 ...
9
10 Problem
11 \forall eps
12 (
13   eps > 0 ->
14   \exists del
15   (
16     del > 0 &
17     \forall x (x^2 < del^2 -> [{x'=-x^3 & x <= 5} ++ {x'=-x}]*x^2 < eps^2)
18   )
19 )
20 End.
21
22 End.
23
24
25 ArchiveEntry "New Entry: attractivity"
26 ...
27 End.

```

The above stability specification is in KeYmaera X's ASCII syntax. A short guide to the syntax is available here: <https://keymaerax.org/KeYmaeraX-sheet.pdf>. Stability corresponds to $\text{UStab}(\alpha)$ and attractivity corresponds to $\text{UGpAttr}(\alpha)$ in the paper. Throughout this artifact, both stability *and* attractivity are proved, corresponding to UGpAS of the systems under consideration.

4. Users can modify the generated text to manually tweak specifications. For example, one could change the attractivity entry to use variable y instead of x as follows:

```

1 ...
2 ProgramVariables
3   Real y;
4 End.
5
6 Problem
7 \forall eps
8 (
9   eps > 0 ->
10   \forall del
11   (
12     del > 0 ->
13     \exists T_
14     (
15       T_ >= 0 &

```

```

16         \forall y
17         (
18             y^2 < del^2 ->
19             [t_:=0; { {t_:=1, y'=-y^3 & y <= 5} ++ {t_:=1, y'=-y}}*]
20             (t_ >= T_ -> y^2 < eps^2)
21         )
22     )
23 )
24 )
25 End.

```

Syntax errors in the modified entries will be shown on the interface with a red X. Hover over the button for the error message.

More complicated modifications are possible, follow the tutorial at <http://keymaerax.org/Xtutorial.html>.

- Click **Save** in the top right corner to save the models and click X to exit the modeling interface.
- Now, click on **New Entry: stability** on the **Models** page (or the corresponding entry name if you renamed it). This should bring up the autogenerated stability specification. Click **Start proof**.
- This entry can be proved using a Common Lyapunov Function (CLF). **(Requires MATLAB)** To automatically generate such a CLF, click on **Differential Equation -> Switched -> Stability CLF**. Click on the stability specification as shown below (make sure the entire formula is highlighted) and KeYmaera X will attempt to generate a CLF as described in Section 4.1, Table 1.

stabilityCLF

Provide tactic input

stab

$$\frac{\Gamma \vdash [\{x' = f_p(x) \& Q\}^*] (\vee)' <= 0 \quad \Gamma \vdash V(0)=0 \wedge (x|=0 \rightarrow V>0)}{\Gamma \vdash \forall \epsilon > 0 \exists \Delta > 0 \forall x^2 < \Delta^2 [\{x' = f_p(x) \& Q\}^*] x^2 < \epsilon^2, \Delta}$$

Select formula (hover and click to select typical formulas, press **option/alt** key and click to select any term or formula).

The screenshot shows the stabilityCLF interface. At the top, there's a header "stabilityCLF". Below it, a section titled "Provide tactic input" contains a proof goal labeled "stab". The goal is a complex logical statement involving quantifiers and functions. Below the goal, there's a section titled "Select formula (hover and click to select typical formulas, press option/alt key and click to select any term or formula)." This section displays a list of terms from the goal, each with a small icon and a label. A mouse cursor is hovering over the first item, which is labeled "Click to apply here". The items are:

- \vee eps
- {
- eps > 0 →
- ∃ del
- {
- del > 0
- $\forall x (x^2 < \text{del}^2 \rightarrow [\{x' = -x^3 \& x \leq 5\} \cup \{x' = -x\}]^* x^2 < \text{eps}^2)$
- }

The proof should complete successfully and you should see a page with the following output:

Proof: All goals closed

Provable(==> ... proved)

Tactic to Reproduce the Proof

stabilityCLF('R==...)

8. If you do not have MATLAB, follow Step 7 but instead input in **V** the CLF x^2 . Then, click on the stability specification to run the proof with this manually specified CLF. This should prove successfully as well.
9. Repeat the above steps for **New Entry: attractivity** to verify it using a CLF.
10. (Optional) In Step 7, choose **Differential Equation -> Switched -> Stability MLF** instead to attempt a Multiple Lyapunov Function (MLF) stability proof. **(Requires MATLAB)** An MLF can be automatically generated.
Alternatively, a manual proof can be done by giving as input to **Vp** the following two MLFs in list syntax $x^2 :: x^2 :: \text{nil}$.
11. (Optional) Try the above steps with other models, e.g., with **Timed**, **Guarded**, and **Generic** switched systems (see examples in §5).
12. Once you are down with evaluation, click on the power button icon in the top-right corner to shut down KeYmaera X and the Docker container.

5 Evaluation: Examples

The examples are all in `models/examples/` and can be evaluated similarly. The following instructions are for evaluating `example1.kyx`.

1. On the Models tab, click on:

Templates -> Plain -> New Model

2. Click on **Select file** and open `models/examples/example1.kyx`. This should load the proved model into the textbox below. Then, click on **Save** in the top-right corner.
3. Click on the **Examples** folder and select **Example 1:stability**. In the resulting dialog, click on **Run stored tactic**. This will run an existing proof with a known Lyapunov function for that example.
4. The proof should complete successfully and you should see a page with the following output:

Proof: All goals closed

Provable(==> ... proved)

Tactic to Reproduce the Proof

```
stabilityStateMLF("4331*x1^2/1000000+...", 'R==...)
```

5. To try the proof yourself, return to the **Models** page and open **Example 1:stability** again. Click on **Start proof**.
6. Similar to § 4, click on **Differential Equation -> Switched -> Stability MLF**. **(Requires MATLAB)** Run the tactic with no arguments so that KeYmaera X will automatically attempt to generate an MLF for this example. Examples with successful auto-generation of Lyapunov functions are labeled with ✓ in the last column Table 2 in the paper.
7. Alternatively, you can use the following MLF as a manual argument to **Stability MLF** in the **Vp** argument:

```
4331*x1^2/1000000+x1*x2/200000+87*x2^2/200000::  
217*x1^2/500000-x1*x2/200000+2161*x2^2/500000:nil
```

You can find suitable Lyapunov functions in the artifact .kx files.

8. Repeat the above steps for **Example 1:attractivity**.
9. The above steps should be repeated for all remaining examples to check the proofs. Please note the following (see Table 2 in the paper):
 - For Example 4, KeYmaera X is unable to automatically find a suitable Lyapunov function. However, **Run stored tactic** will use a known Lyapunov function provided with the artifact.
 - For Example 5, there is no proof, i.e., we do not have a suitable Lyapunov function.
 - For Example 6, you can use **StabilityCLF** (and **AttractivityCLF**) instead of MLF as a CLF suffices for proving stability of this system.
 - For Example 8–9, KeYmaera X does not support auto-generation of Lyapunov functions for time-dependent switching models. For these examples, **Run stored tactic** can be used to check the proofs.
 - For Example 10, the default approximation degree in the version of KeYmaera X in this artifact is set to a low number (in the interest of time). Thus, neither stability nor attractivity proof will succeed for this example.

6 Evaluation: Case Studies

The case studies can be evaluated similarly to the examples in § 5, i.e., load the relevant model then click **Run stored tactic** to check the accompanying proof for both stability and attractivity specifications. All proofs should complete successfully and result in a success page, e.g., for `canonmaxsys-stab.kyx`, this should be shown on completion:

Proof: All goals closed

```
Provable( ==> a()>0&b()>0&a()-f()>0&b()-g()>0&gam()<=0->
... proved)
Tactic to Reproduce the Proof
...
```

You can also check that the proved specifications correspond to those shown in Section 5 of the paper.

- **Canonical Max System.** The models are in `models/canonmaxsys/`, where `canonmaxsys-stab.kyx` proves stability, `canonmaxsys-attr-0.kyx` proves attractivity for $f = 0$ (see Footnote 5 in paper), and `canonmaxsys-attr-1.kyx` proves attractivity for $f \neq 0$. The model is shown on lines 1004–1015 in the paper. Note that `canonmaxsys-attr-1.kyx` contains a large proof and may take several minutes to run to completion.
- **Automated Cruise Control.** Both stability and attractivity specifications are proved in `models/cruiesctrl/cruise.kyx`. As explained in the paper, the cruise controller model is adapted from the Stabhyli example files obtained from <https://uol.de/svs/forschung/avacs/stabhyli>.
- **Brockett's Nonholonomic Integrator.** The models are in `models/nonholo/`.
 - `nonholoeventsimple.kyx` proves stability and attractivity for the simple event-triggered model given in lines 1193–1207 in the paper.
 - `nonholoevent.kyx` proves stability and attractivity for the event-triggered model given in lines 1227–1217 in the paper. Note that the attractivity proof is large and may take several minutes to run to completion.
 - `nonholotime.kyx` proves stability and attractivity for the time-triggered model given in lines 1260–1267 in the paper.