Safe Learning Documentation

Tianpeng Zhang

January 15, 2022

1 Introduction

This is the documentation of the Python implementation of the Safe Learning algorithm and its related simulations based on [1]. The code can be found in the following online GitHub repository: `https://github.com/lina-robotics-lab/SafeLearning`.

See Section 2 for a mathematical description about the system model we used in the simulations.

See Section 3 for a quick guide to run the simulation.

Section 4 contains detailed documentations for each component of the implementation.

2 System Model

We consider a dampened spring-mass system in the simulation. The system contains a mass with weight $m > 0$ attached to a spring with stiffness constant $k > 0$. The mass is restricted to move along a one-dimensional surface perpendicular to the gravity direction. We assume no friction between the mass and the surface, but there is a drag negatively proportional to the velocity of the mass, with a drag constant $\lambda > 0$. We can apply an external force u on the mass along the trajectory of its movement to control its state.

Let $x \in \mathbb{R}$ be the location of the mass, the system equation in continuous time can be defined as

$$\ddot{x} = (-kx - \lambda\dot{x} + u)/m \tag{1}$$

In the implementation, we approximate (1) using a discrete-time system defined by

$$s_t := \begin{bmatrix} x_t \\ v_t \end{bmatrix} = (\begin{bmatrix} 0 & \Delta \\ -\frac{k\Delta}{m} & -\frac{\lambda\Delta}{m} \end{bmatrix} + I) \begin{bmatrix} x_{t-1} \\ v_{t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\Delta}{m} \end{bmatrix} u_{t-1} \tag{2}$$

where s_t is the system state at time step t, and $\Delta > 0$ is a small constant representing the sampling time interval.

3 How to run the simulations

Step 1: clone the git repository to a local folder by

```
git clone https://github.com/lina-robotics-lab/SafeLearning
```

Step 2: Install the Python packages required by the simulation. Especially, jupyterlab, matplotlib, scipy, and cvxpy.

I recommend installing the Python environment manager called *conda* `https://docs.conda.io/en/latest/miniconda.html`, and follow `https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html` to create a new Python environment. This ensures the new packages do not interfere with the original Python on your computer which serves crucial purposes such as rendering your desktop and windows.

A typical flow of operations after installing *conda* is

```
conda create -n SafeLearningEnv python=3.9 scipy jupyterlab matplotlib
conda activate SafeLearningEnv
pip install cvxpy
```

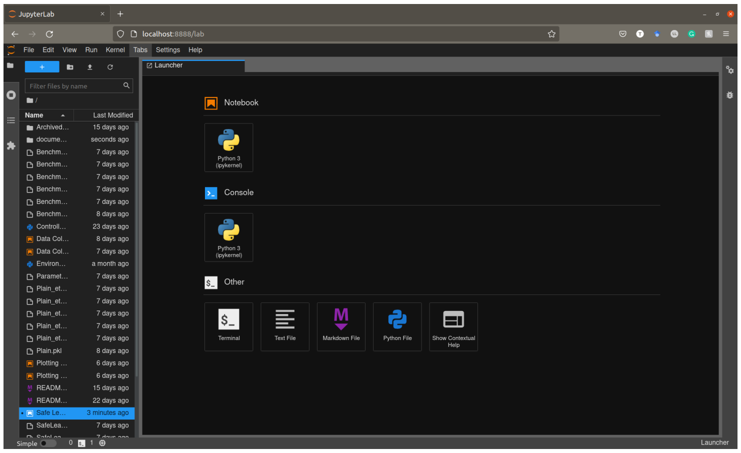
Step 3: Ensure the Python environment we created in Step 2 in command prompt by

```
conda activte SafeLearningEnv
```

Navigate to the SafeLearning folder in command prompt. Then start the jupyter lab by running

```
jupyter lab
```

The default browser of your computer should pop up and show a tab loading the jupyter lab. Eventually you will see something like the screenshot in Figure 1.



Name	Last Modified
Archived…	15 days ago
docume…	seconds ago
Benchm…	7 days ago
Benchm…	7 days ago
Benchm…	7 days ago
Benchm…	7 days ago
Benchm…	7 days ago
Benchm…	8 days ago
Controll…	23 days ago
Data Col…	8 days ago
Data Col…	7 days ago
Environ…	a month ago
Paramet…	7 days ago
Plain_et…	7 days ago
Plain_et…	7 days ago
Plain_et…	7 days ago
Plain_et…	7 days ago
Plain_et…	7 days ago
Plain.pkl	8 days ago
Plotting …	6 days ago
Plotting …	6 days ago
READM…	15 days ago
READM…	22 days ago
Safe Le…	3 minutes ago
SafeLea…	7 days ago

Figure 1

Step 4: in the jupyter lab window, navigate to *notebooks* foler, double click to open the *[Safe Learning Simulation.ipynb]* notebook.

Press *Esc* to enter command mode.

Press *Ctrl+a* to select all cells.

Finally, press *Ctrl + enter* to run the entire notebook from start to end.

The Python notebook works similarly as the live scripts/run by section utility in Matlab. Watch this video for a 30-min beginner's tutorial on Python notebooks:`https://www.youtube.com/watch?v=HW29067qVWk`.

4 Detailed Documentations

4.1 File Structure

The repository is structured in the following manner:

```
Repository Root
- Scripts(.py files) and README
- documentation
  - Latex files
- notebooks
  - Demonstration notebook(SafeLearningSimulation.ipynb)
  - Devel
    - Notebooks under development
    - The data folder
      - Data files(.pkl)
  - Folders containing specific experiments
    - The simulation notebook
    - The plotting notebook
    - The data folder
      - Data files(.pkl)
```

We ensure the data files are encapsulated in the inner-most layer and the scripts are exposed in the outer-most layer, and that folders for individual experiments follow the template specified above.

4.2 Environment

4.3 Controllers

4.4 Subroutines

4.5 Simulations

References

[1] Y. Li, S. Das, J. Shamma, and N. Li, "Safe adaptive learning-based control for constrained linear quadratic regulators with regret guarantees," *arXiv preprint arXiv:2111.00411*, 2021.