Student research project

Adaptive solvers for differentiable predictive control framework

In model-based control frameworks, e.g. differentiable predictive control (DPC), solving ODEs quickly and accurately is essential in both training and simulation. Adaptive solvers adjust the step length based on the change in system dynamics, shown in Fig. 1. In case of slow dynamics, such solvers would increase step size, less steps taken, which would reduce number of evaluations. On the other side, fast state changes would result in shorter steps, hence more computations. This project investigates the use of adaptive solvers in Diffrax (a JAX-based differential equation solver library) to improve the computation efficiency of the learning process. One will explore different solvers, benchmak performance, and propose best practices for their use in real-world control pipelines.

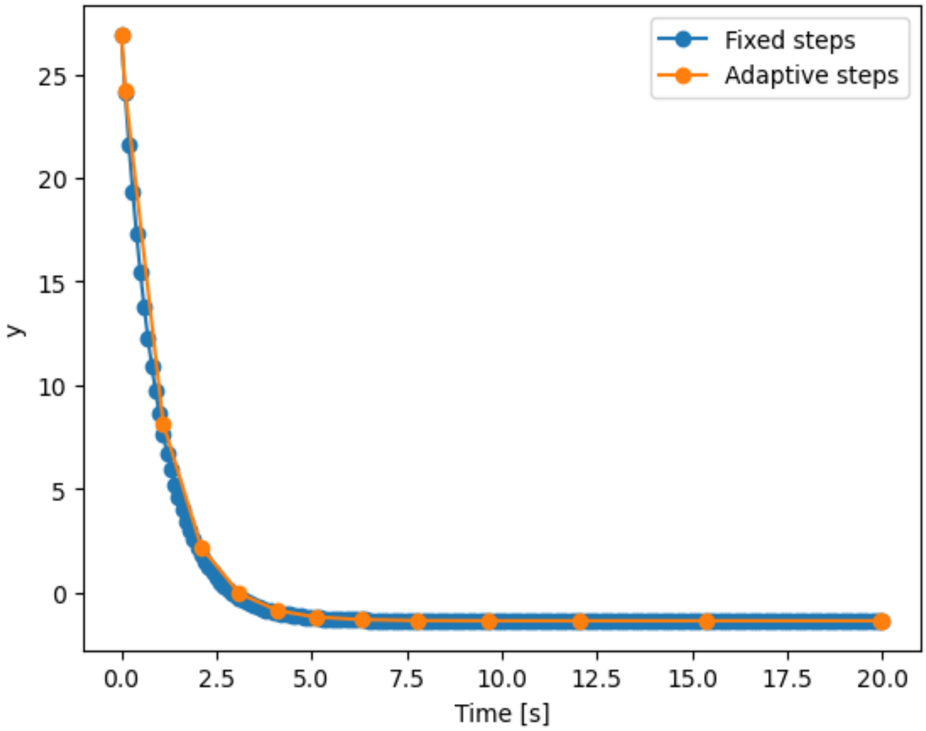

Figure 1: Adaptive-steps solver vs fixed-steps

Necessary requirements:

- Strong Python background
- Fundamental knowledge of control engineering
- Basic knowledge of machine learning
- Solid background in numerical methods and ODEs
- Familiarity with JAX or a willingness to learn quickly

WP 1: Familiarization with JAX and Diffrax **[3 weeks]**
The project begins with setting up the environment, studying how Diffrax solvers are used in JAX-based simulations, and building a simple training loop. By the end of this stage, one should be comfortable with modifying the solve behavior in code.

WP 2: Integration of adaptive solvers in the control framework **[2 weeks]**
Here, the training loop is modified by replacing the fixed-step ODE solver with different adaptive-step solvers in Diffrax. Compatibility with JAX's JIT and autodiff tools will be ensured, and correctness of learning will be validated.

WP 3: Solver behavior analysis **[3 weeks]**
In this phase, many experiments will be run, testing edge cases and documenting scenarios where adaptive solvers outperform or underperform fixed-step methods. Moreover, solver-specific hyperparameters, such as tolerances and step size bounds, should be tuned to find optimal configurations for speed and accuracy.

WP 4: Benchmarking training and inference performance **[2 weeks]**
In this stage, training time, inference time, solver steps, and control accuracy between the fixed-step and adaptive solvers will be compared, and results, plots and insights should be well-documented.

WP 5: Documentation **[2 weeks]**
Lastly, a final report and short presentation will be prepared, highlighting the findings and recommendations for future use.

Gantt chart

The planned timetable is shown in the Gantt diagram below.

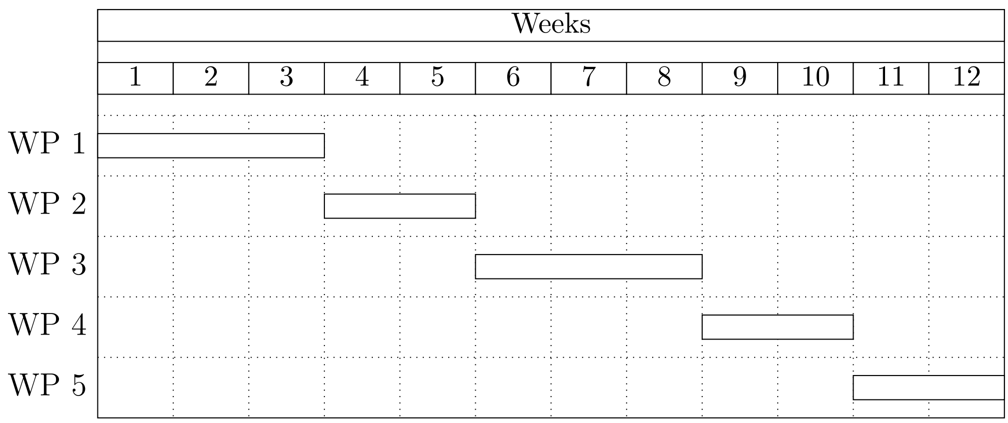

Figure 2: Gantt chart for the thesis.