Problem Statement for Koh Young AI Competition 2024
: Swing-up and balancing control of a quadruple inverted pendulum

1. Control objective

The swing-up and balancing control problem of a quadruple inverted pendulum (QIP) is an important topic in control theory due to its complex dynamics and highly nonlinear characteristics. The challenge of controlling a QIP arises from its inherent instability and the intricate coordination required among its multiple segments. Therefore, the purpose of the competition is to quickly and accurately control the QIP system to achieve swing-up and balancing.

2. Assumptions and physical parameters

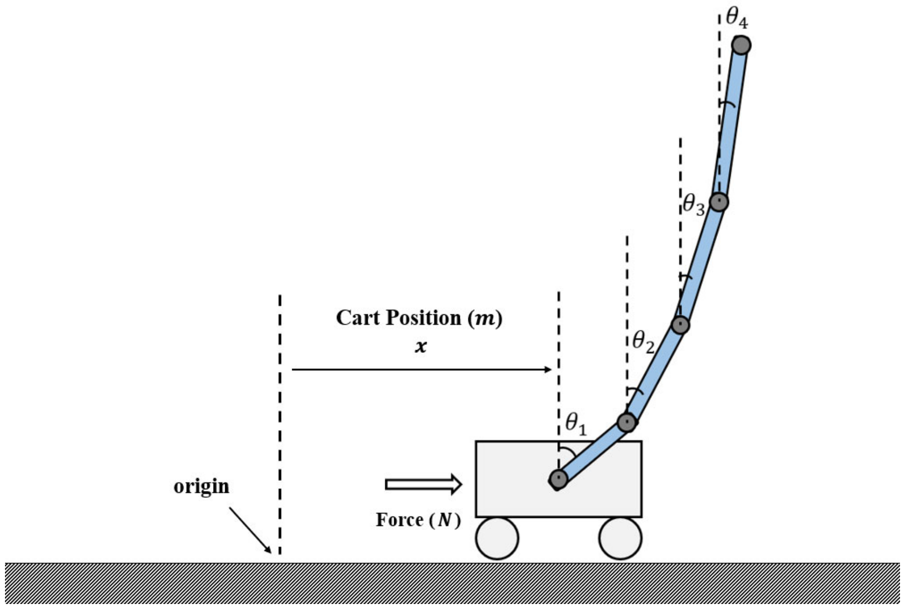

Figure 1. Diagram of the QIP system.

Physical parameters	Cart	Link 1	Link 2	Link 3	Link 4
Mass (kg)	1.3000	0.2000	0.2500	0.4300	*Not given*
Length (m)	-	0.1500	0.2000	0.4000	0.8000
*Distance from hinge to *CoG (m)*	-	0.0750	0.1000	0.2000	*Not given*
***MoI of rods about their CoG ($kg{\cdot}m^2$)*	-	0.0015	0.0033	0.0229	*Not given*

Table 1. Physical parameters of the cart and links.

**CoG: Center of Gravity*

***MoI: Momentum of Inertia*

Properties	Variables/Values
State variables	$[x,\dot{x},\theta_1,\dot{\theta}_1,\theta_2,\dot{\theta}_2,\theta_3,\dot{\theta}_3,\theta_4,\dot{\theta}_4]$
Initial State (downward)	[0, 0, π, 0, π, 0, π, 0, π, 0]
Goal State (upward)	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Maximum force bound	[-60N, 60N]

Table 2. Properties for QIP simulation.

Figure 1 provides an overview of the entire QIP system. The physical parameters and simulation properties are listed in Table 1 and Table 2, respectively. Additionally, we assume there is no friction in the cart, pole, or hinges.

3. Details

(1) Code Anaylsis

Participants are required to complete the **fill_the_code.py** file, which contains a user-specific **Controller** class. Specifically, they need to implement the **compute_force** method. This method takes the state ($[x,\dot{x},\theta_1,\dot{\theta}_1,\theta_2,\dot{\theta}_2,\theta_3,\dot{\theta}_3,\theta_4,\dot{\theta}_4]$) as input and returns the force as output. The input state must be of type Python list, and the output force must be of type Python float. The force is constrained to the range **[-60N, 60N]**. If the force exceeds this range, it will automatically be clipped to the maximum or minimum.

Participants should first run the simulation in a Linux based terminal and test their controller using both **example.py** and **main.py**. Additionally, we provide a Jupyter notebook with detailed descriptions for better understanding of both files.

- The **example.py** file helps participants understand how to interact with the simulation.
- The **main.py** file will be used for evaluation. Participants should ensure that their implementation of the **compute_force** method works correctly without any error.

The **main.py** script will also render a visualization of the simulation, showing the controller's behavior. After the visualization, the state transitions over the specified period will be stored as a text file; **transitions.txt**. The score will be measured based on the **transitions.txt**.

(2) User Manual

* All the script must be executed in the **/QIP** directory

Setup Environment:

```
python == 3.9
matplotlib == 3.9.2
numpy == 2.0.2
```

All submitted code will be tested in the setup described above, so please ensure that your code works properly under that environment.

Running the simulation:

```
$ ./simulator/base --ctrl_ms 10
```

A binary file located at `/simulator/base' is for QIP simulation. It sets up a network server and client using the `TCP/IP' protocol. You can specify a control period of the simulation as an argument; **ctrl_ms**.
(Note that the control period must be greater than or equal to 1 ms; otherwise, an error message will appear.)

Please refer to the Jupyter notebook for more details. (**example.ipynb**, **main.ipynb**)

Run the example:

```
$ python example.py
```

Render and save transitions:

```
$ python main.py
```

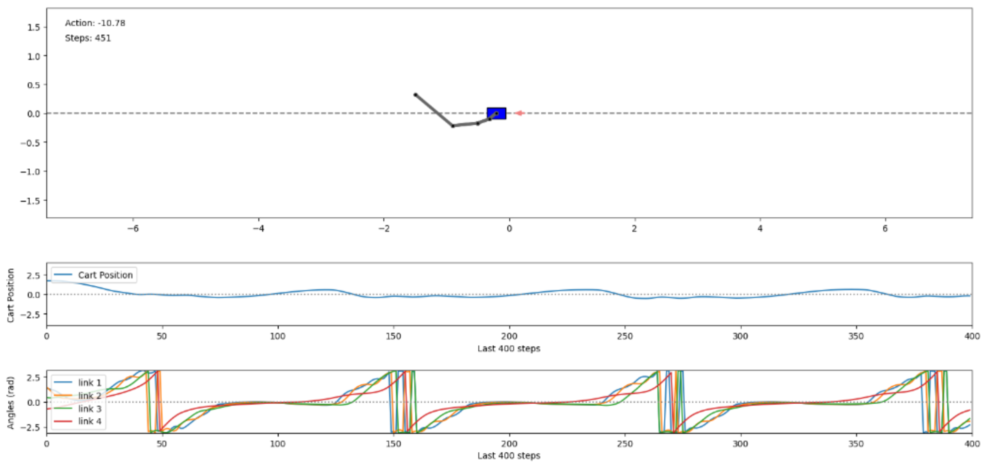

FIgure 3. Rendering a visualization of the resulting pendulum angles and cart position over time.

Before running **main.py**, participants should fill the **Controller** class in **fill_your_code.py**. **main.py** tests your controller and produce the corresponding output (**transitions.txt**). Participant's score will be calculated based on the **transtions.txt** file.

(3) Submission Format

Participants should submit a single **.zip** file that includes **fill_your_code.py** and **main.py.** If necessary, you may also include additional .py files or other types of files (e.g., .yaml, .pth, .h5). The **main.py** script must work perfectly with the given simulator and the files, as the evaluation will be based on the **transitions.txt** generated from running the **main.py** script.

4. Instructions

- Participants can submit results up to two times until the due date. The committee will choose the better one for the final evaluation. Your submission should be sent to

  sooheehan@postech.ac.kr and conference@icros.org
- If the provided QIP simulator is used for publication, please include the following acknowledgment:
  "The QIP simulator in this work is provided by Koh Young Technology, Inc."
- Stay up to date from http://KohYoungAI.iccas.org/.

5. Evaluation metric

- We evaluate the time it takes for a state variable to reach the following region:

$$-0.05[\mathrm{rad}] \le \theta_{\mathrm{i}} \le -0.05[\mathrm{rad}]$$
$$-0.05[\mathrm{rad/sec}] \le \dot{\theta}_i \le -0.05[\mathrm{rad/sec}],\ i=1,2,3,4$$
$$-0.5[\mathrm{m}] \le x \le 0.5[\mathrm{m}]$$

* For more than 5 seconds, the state variables must remain in the above region.

- Temporary uncertainties are applied to measure the robustness of the designed control algorithm.
- We also evaluate the novelty of the designed control algorithm by 40% of the total score.

6. References

- Chung Choo Chung, John Hauser, “Nonlinear control of a swinging pendulum”, Automatica, Volume 31, Issue 6, 1995.
- Tobias Glück, Andreas Eder, Andreas Kugi, “Swing-up control of a triple pendulum on a cart with experimental validation”, Automatica, Volume 49, Issue 3, 2013.
- Jongchan Baek, Changhyeon Lee, Young Sam Lee, Soo Jeon, Soohee Han,“Reinforcement learning to achieve real-time control of triple inverted pendulum‘, Engineering Applications of Artificial Intelligence, Volume 128.