Model Predictive Control using Quantum Long-Short Term Memory in Reinforcement Learning Environments

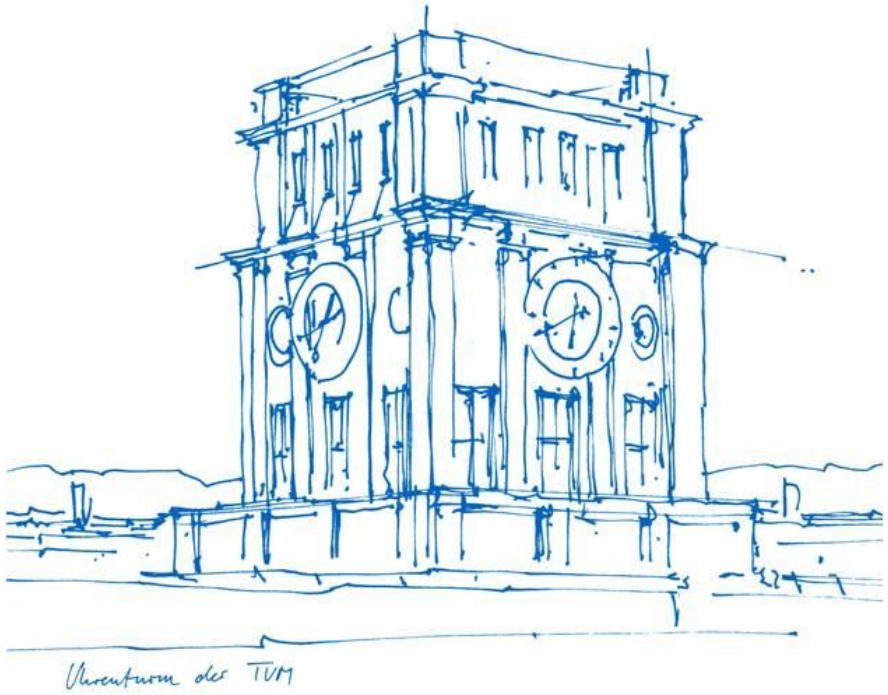
Siva Karthikeya Mandarapu

Supervisor(s): Univ.- Prof. Dr. Christian B. Mendl and PD. Dr. habil. Jeanette Miriam Lorenz

Advisor(s): Dr. Daniel Hein and Dr. Steffen Udluft (Siemens)

TUM School of CIT, Chair of SCCS

Munich 28.11.2024



Agenda

- Background and Motivation
- Objectives
- Implementation Details
- Results and Discussion
- Conclusion and Outlook

Background and Motivation

Model Predictive Control

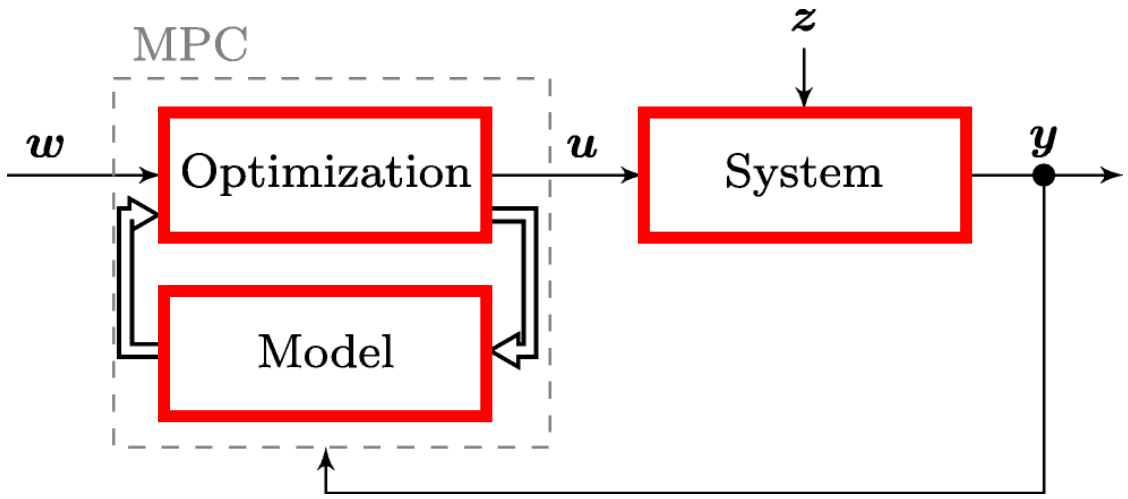


Figure 1: Model Predictive Control [1]

- Model predictive control (MPC) - a set of advanced control methods, which make use of a process model to predict and optimize the future behavior of the controlled system
- By solving a—potentially constrained—optimization problem, MPC determines the control law implicitly.
- This shifts the effort for the design of a controller towards *modeling of the to-be-controlled process.*
- Widely used in industries incl. Aerospace, Automotive, and Chemical processes, due to its ability to handle multivariable control problems and constraints effectively.

Long-Short Term Memory

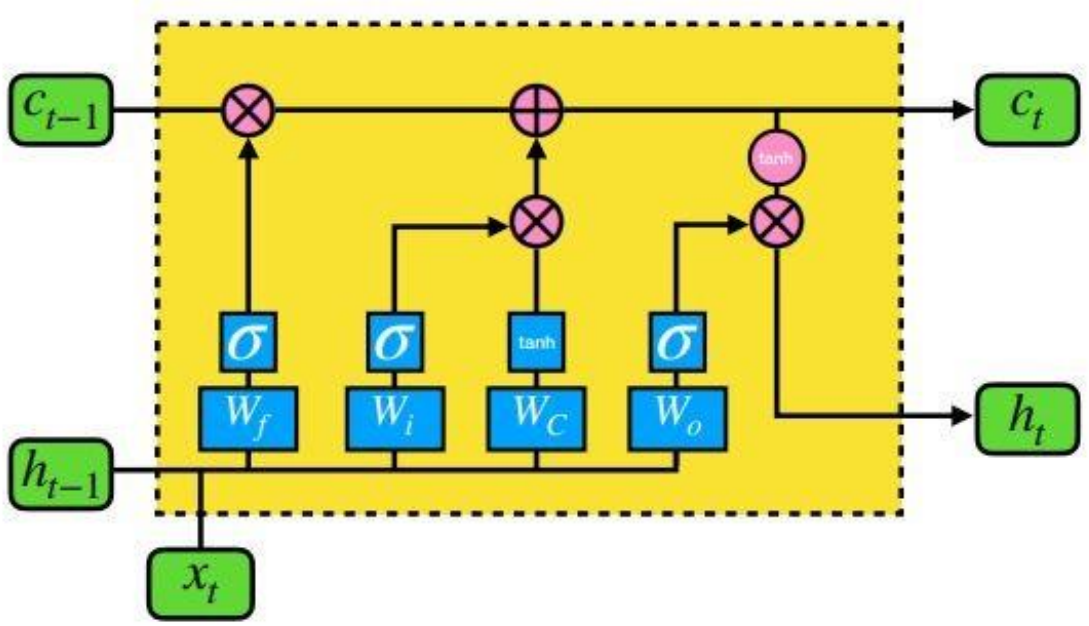


Figure 2: Long short-term memory [2]

- A type of recurrent neural network (RNN) designed to model sequence and temporal dependency data.
- LSTMs are effective in capturing long-term dependencies in time-series data.

$$
\begin{aligned}
\textbf{Input Gate:} \quad & \mathbf{i}_t = \sigma\left(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i\right), \\
\textbf{Forget Gate:} \quad & \mathbf{f}_t = \sigma\left(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f\right), \\
\textbf{Output Gate:} \quad & \mathbf{o}_t = \sigma\left(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o\right), \\
\textbf{Cell State Update:} \quad & \tilde{\mathbf{c}}_t = \tanh\left(\mathbf{W}_c \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c\right), \\
\textbf{Cell State:} \quad & \mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \\
\textbf{Hidden State:} \quad & \mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t).
\end{aligned}
$$

Variational Quntutm Circuits

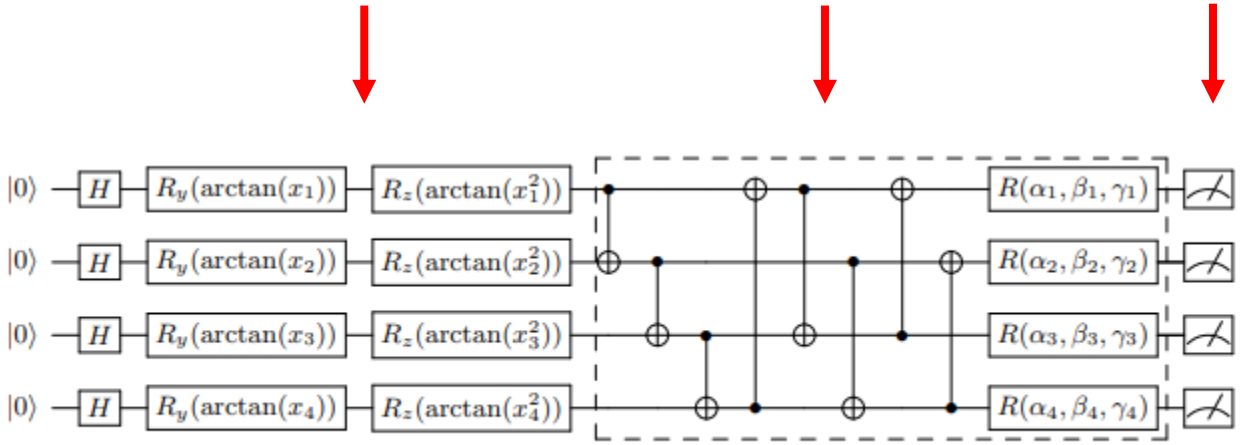


Figure 3: Variational Quantum Circuit [2]

- Extension of classical LSTM into the quantum realm by replacing the classical neural networks in the LSTM cells with VQCs.

The VQC is composed of three major parts:

- Data encoding - transform the classical vector (input) into a quantum state.
- Variational layer - actual learnable components, with circuit parameters updated via gradient descent algorithms.
- Quantum measurements - to retrieve the values for further processing.

Quantum Long-Short Term Memory

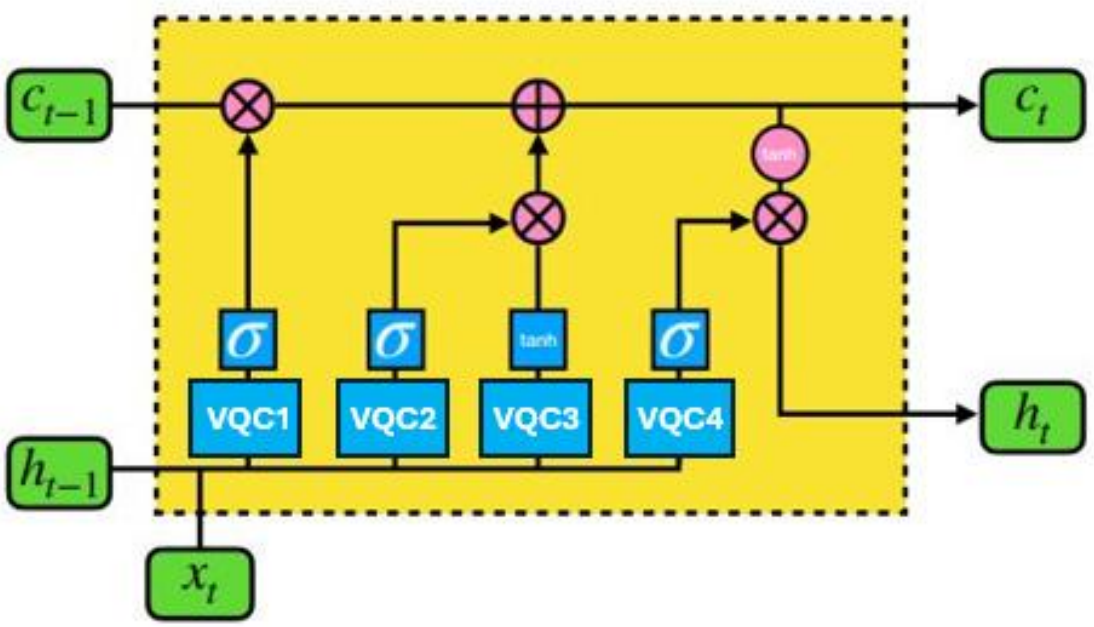


Figure 4: Quantum long short-term memory

- In quantum LSTM, the classical LSTM gates (input, forget, and output), and cell state update are replaced with variational quantum circuits.

$$
\begin{aligned}
\textbf{Forget Gate:} \quad & f_t = \sigma\left(\mathbf{VQC}_1(\mathbf{v}_t)\right), \\
\textbf{Input Gate:} \quad & i_t = \sigma\left(\mathbf{VQC}_2(\mathbf{v}_t)\right), \\
\textbf{Cell State Update:} \quad & \tilde{C}_t = \tanh\left(\mathbf{VQC}_3(\mathbf{v}_t)\right), \\
\textbf{Cell State:} \quad & C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \\
\textbf{Output Gate:} \quad & o_t = \sigma\left(\mathbf{VQC}_4(\mathbf{v}_t)\right), \\
\textbf{Hidden State:} \quad & h_t = o_t \odot \tanh(C_t).
\end{aligned}
$$

Literature Review – Quantum LSTM

- Previous works on quantum LSTMs have shown that they learn more information than their classical counterparts in fewer epochs, and they are better at learning local features than LSTMs when there is a temporal structure involved in the data [2].

- Quantum LSTMs have also been used as Q-function approximators to realize quantum deep recurrent Q-function learning in [3]. They have shown higher stability and average scores compared to their classical counterparts.

- FedQLSTM framework used in [4], operating on temporal data, achieves faster convergence (25-33% fewer communication rounds) and reduced computations compared to classical FL frameworks, highlighting its potential for quantum sensor-inspired applications.

- The proposed QRNN-based framework in [5] utilizes a hybrid QLSTM reservoir for quantum A3C, demonstrating stability and comparable performance to fully-trained models, paving the way for efficient quantum reinforcement learning with recurrence.

Objectives

1. Implement a quantum LSTM using VQCs from literature
2. Compare quantum LSTM with classical LSTM and MLP
3. Evaluate the feasibility of quantum LSTMs in MPC
4. Explore the applicability of quantum LSTM for complex control problems

Implementation Details

Cart-pole Benchmark

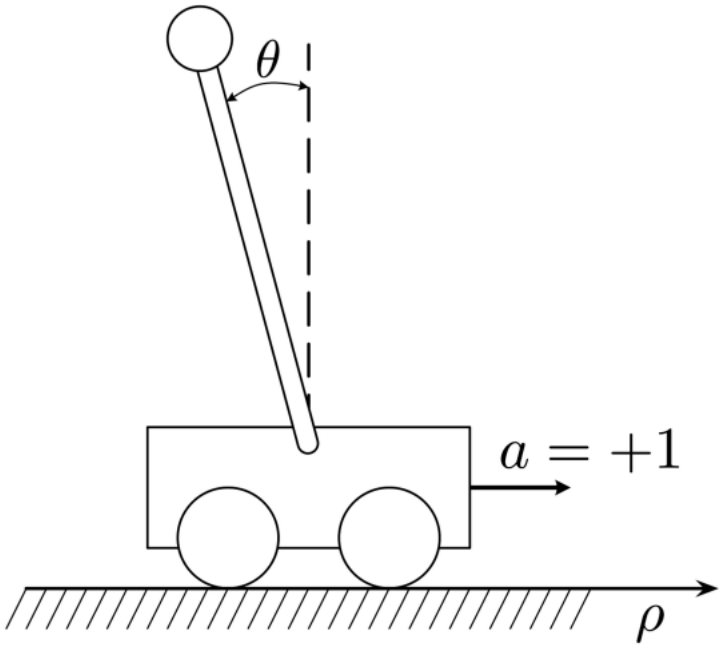


Figure 5: Cart-pole system [6]

Parameter	Lower Bound	Upper Bound
Cart Position	-2.4	2.4
Cart Velocity	$-\infty$	∞
Pole Angle	$-\frac{\pi}{15} \approx -0.2095$	$\frac{\pi}{15} \approx 0.2095$
Pole Angular Velocity	$-\infty$	∞

Table 1: Bounds for the cart-pole environment.

- Fully observable cart-pole benchmark.
- Partially observable cart-pole benchmark

System identification

- Data:
  - Generated 10,000 samples (state variables, actions, and delta states) using a random policy on the cart-pole system.
  - Data scaled for numerical stability.
- Models:
  - Classical LSTM: 5 hidden units.
  - Quantum LSTM: 16 hidden units, 4 qubits with quantum gates.
  - MLP: 10 hidden units.
- Training:
  - Used MSE loss, with Adam optimizer for Classical LSTM and Adagrad for Quantum LSTM.
  - Dataset: 8,000 training, 2,000 testing samples.
- Rollout Evaluation:
  - Predicted delta states iteratively for 50 steps.
    Compared predictions to ground truth.

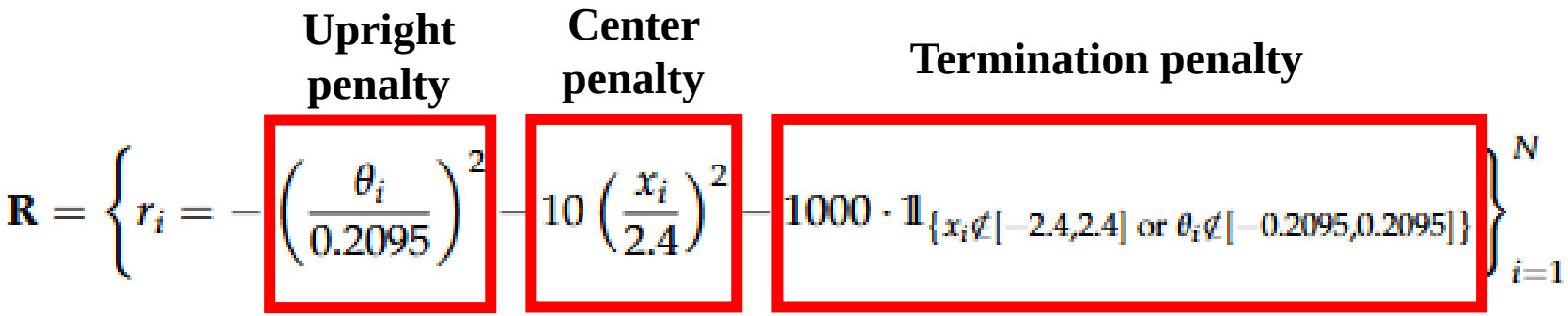
Model Predictive Control Framework

- Initialize: Set current state $s \leftarrow s_0$.

- Repeat Until Termination Conditions Are Met:
  - Perform PSO Procedure to find the best action sequence $x^{\wedge}$:
  - a. Initialize a population of particles with random positions x_i and velocities v_i.
  - b. Evaluate the fitness of each particle based on model predictions $f(x_i)$
  - c. Update particle positions y_i and best neighbourhood positions y_g .
  - d. Adjust velocities v_i and positions x_i according to PSO rules.
  - e. Repeat until convergence or reaching the maximum number of iterations.
  - f. Select the best action sequence $x^{\wedge}$.
- Extract the first action $a^{\wedge}$ from $x^{\wedge}$.
- Apply $a^{\wedge}$ to the system and update s based on the response.

- Output: Termination occurs when the planning horizon T or task success is achieved.

Reward Function



$\boldsymbol{R}$: Reward vector containing rewards $\boldsymbol{r_i}$ for each state i in a batch of size N.

$\boldsymbol{r_i}$: Reward for the i-th state, combining:

- Upright Penalty: Deviation of the pole angle ($\boldsymbol{\theta_i}$) from vertical.
- Center Penalty: Deviation of the cart position ($\boldsymbol{x_i}$) from the center.
- Termination Penalty: Applied if $\boldsymbol{x_i}$ or $\boldsymbol{\theta_i}$ go out of bounds.

$\boldsymbol{\theta_i}$: Pole angle (in radians); θ = 0 is vertical.

$\boldsymbol{x_i}$: Cart position (in meters); x = 0 is the track center.

Results and Discussion

Prediction of Sine Function

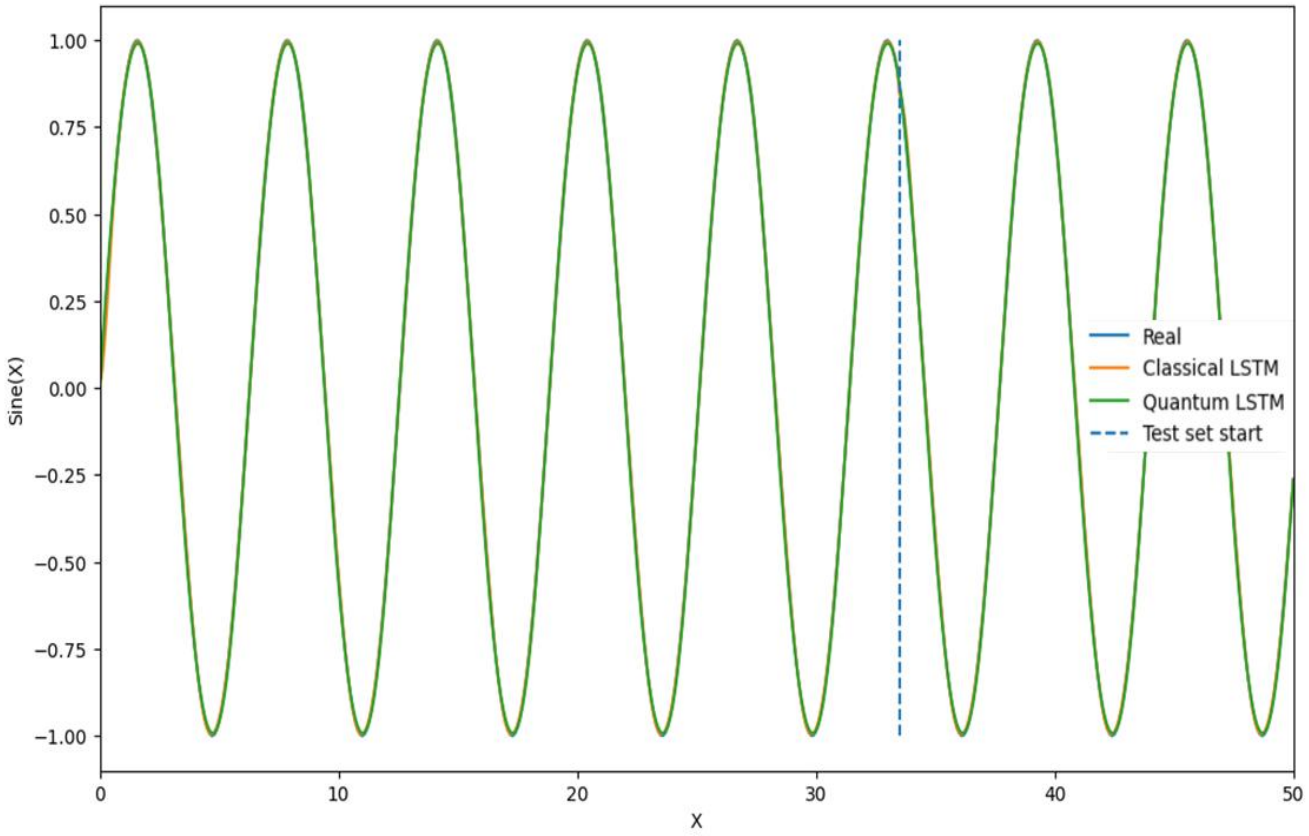


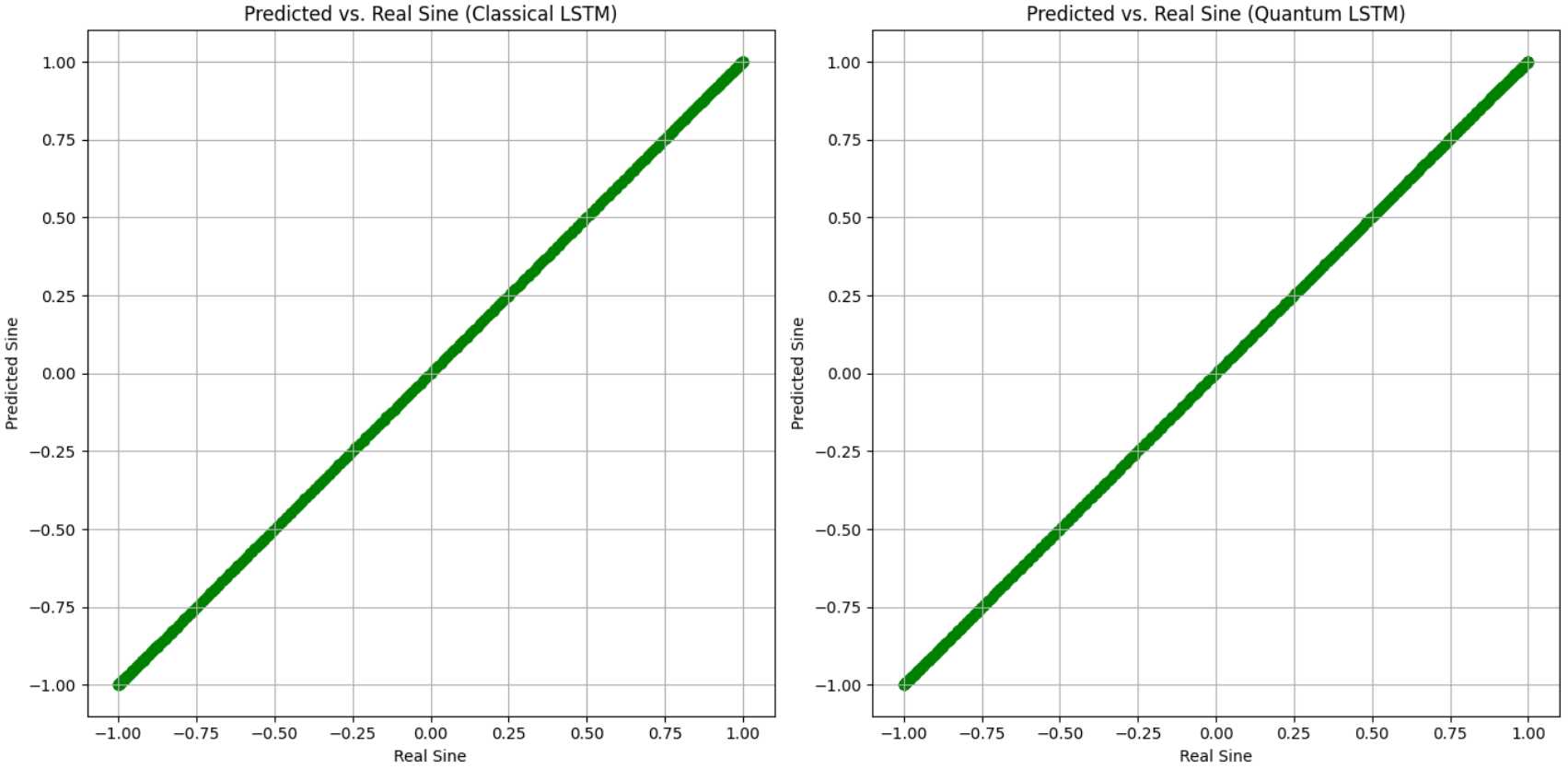
Figure 6: Sine function approximation

- The sine function $y=sin(x)$ was sampled over the range $x \in [0,50]$ with 501 points.
- Models used the previous five $sin(x)$ values as inputs to predict the next value.
- Both Quantum LSTM and Classical LSTM accurately modelled the sine function, demonstrating their ability to approximate smooth periodic signals.

	Training Loss	Testing Loss
Quantum LSTM	4.56×10^{-5}	3.45×10^{-5}
Classical LSTM	4.98×10^{-5}	3.12×10^{-5}

Table 2: Average training and testing loss comparison for five training repetitions

Prediction of Sine Function



- Scatter plots show predictions closely align with actual values, with points concentrated near the bisector, confirming accuracy.

Figure 7: Scatter plot with the data on the x-axis and the model predictions on the y-axis for sine function

Prediction of Damped Harmonic Oscillator (DHO)

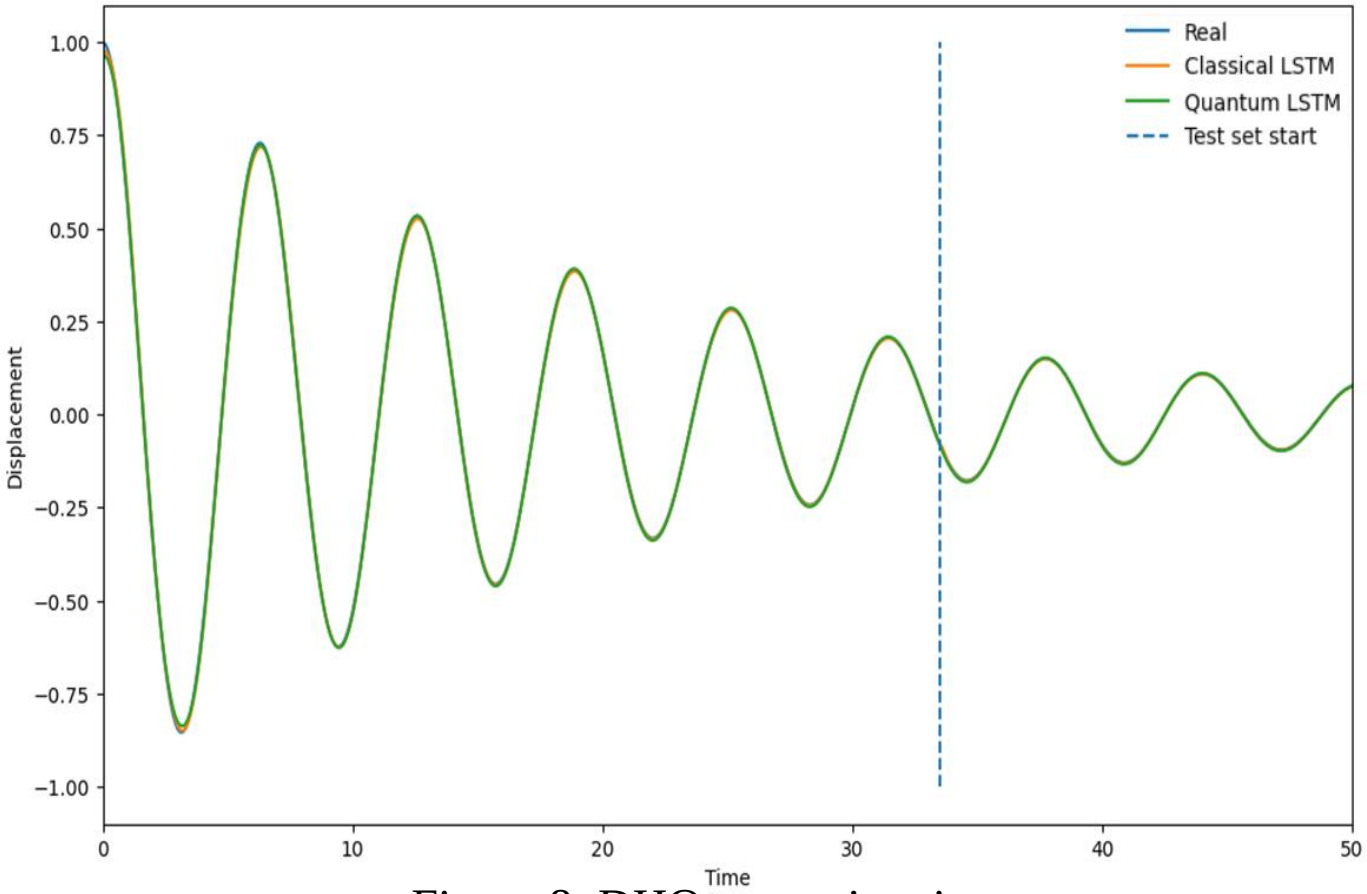


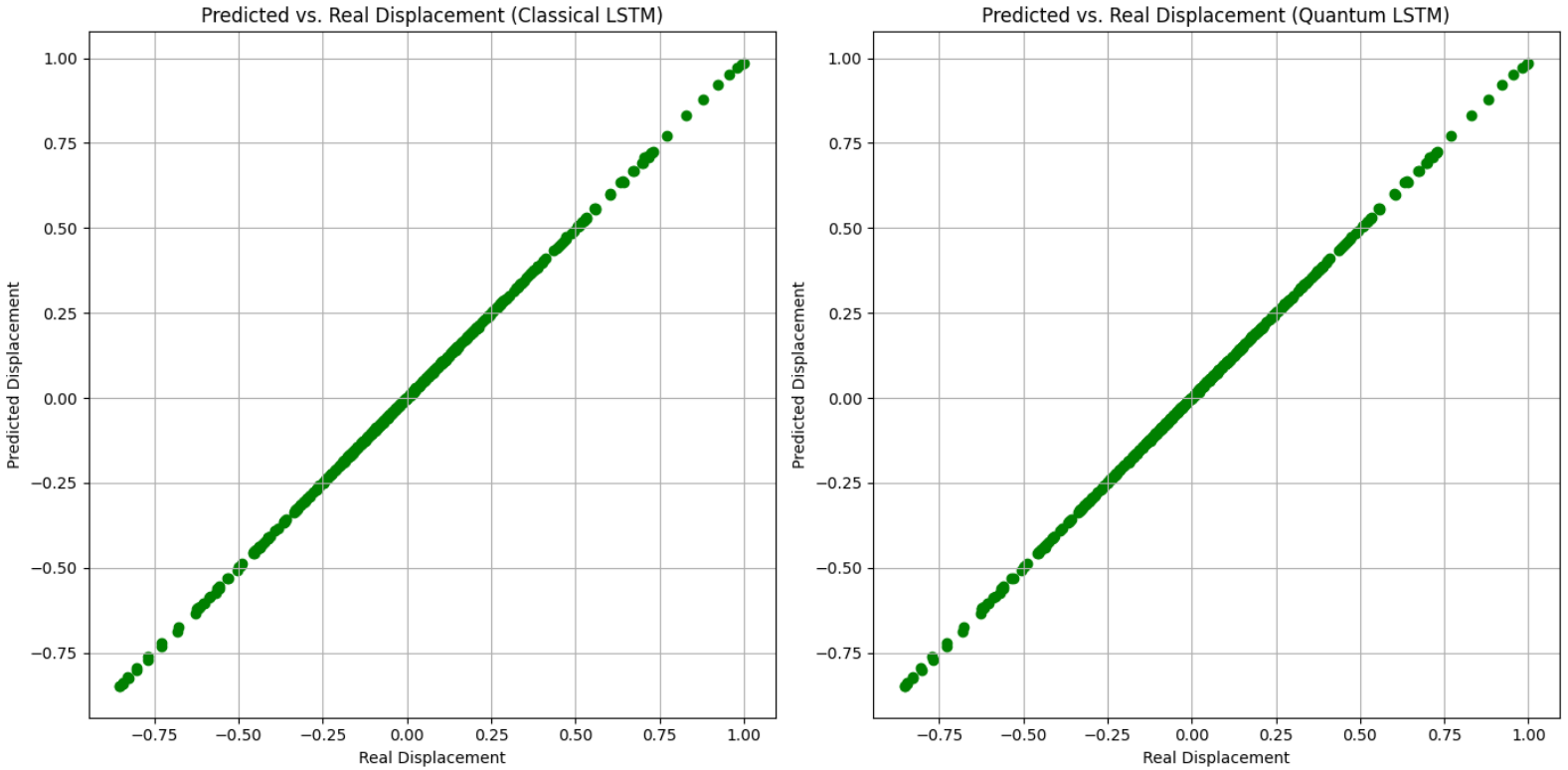
Figure 8: DHO approximation

- Synthetic data was generated using the DHO model.
- The dataset includes displacement $x(t)$ as the target variable, with the previous five displacement values as model inputs.
- Both Quantum LSTM and Classical LSTM accurately approximated the DHO function.

	Training Loss	Testing Loss
Quantum LSTM	5.70×10^{-5}	5.48×10^{-6}
Classical LSTM	3.31×10^{-5}	2.81×10^{-6}

Table 3: Average training and testing loss comparison for five training repetitions

Prediction of Damped Harmonic Oscillator (DHO)



- Scatter plots show predicted values closely match actual data, with points concentrated on the bisector.

Figure 9: Scatter plot with the data on the x-axis and the model predictions on the y-axis for sine function

Prediction of Bessel functions

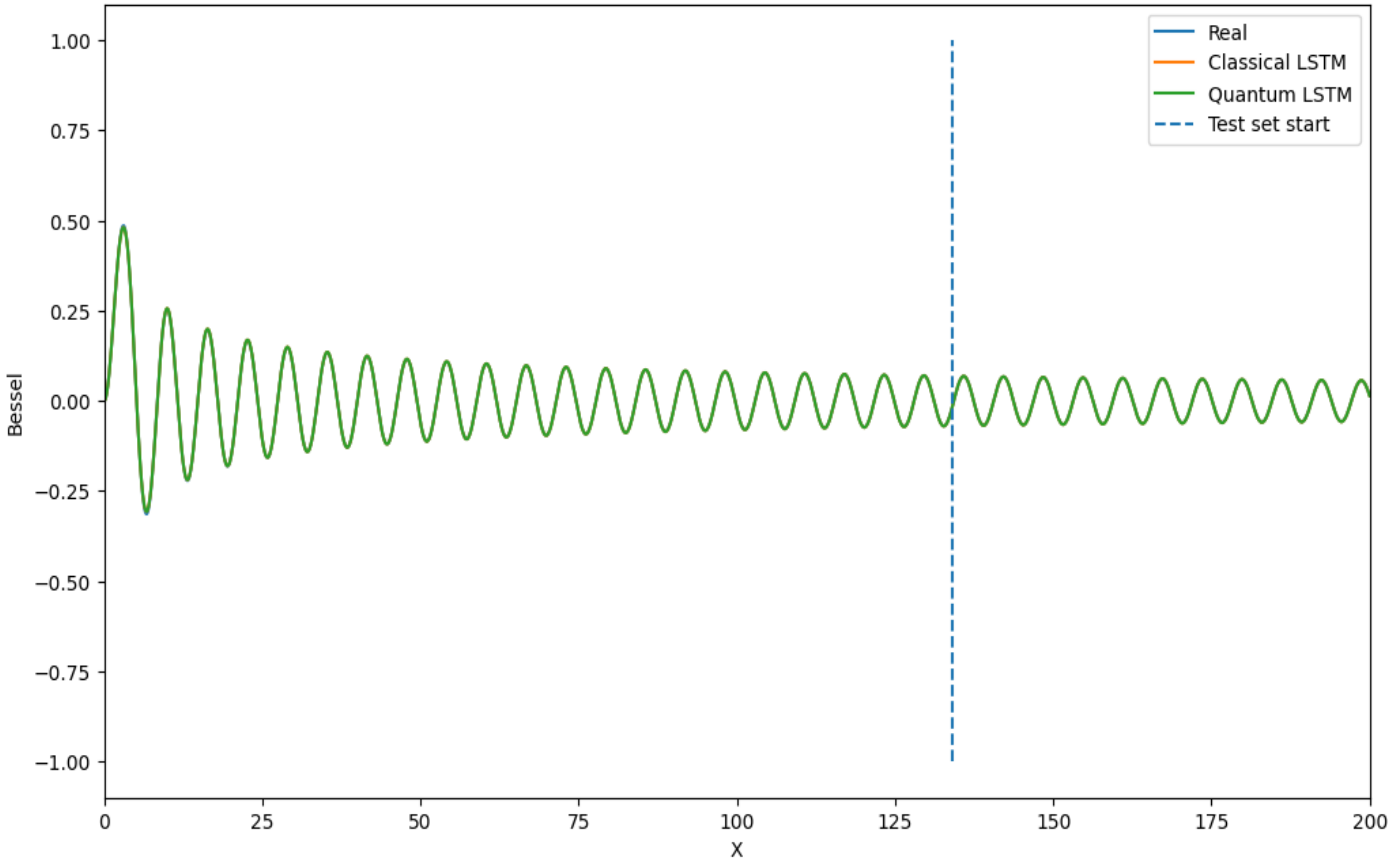


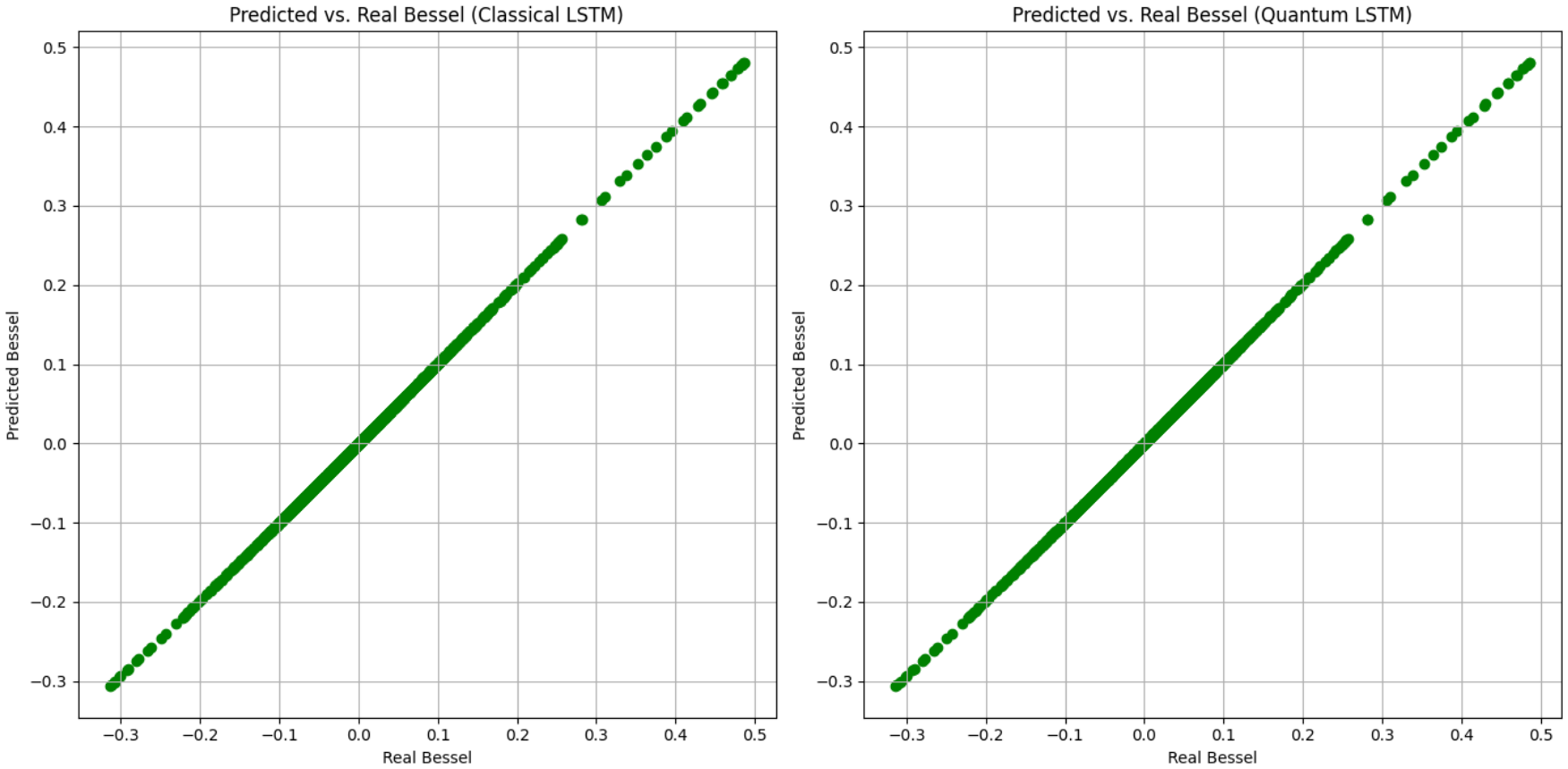
Figure 10: Bessel function approximation

- Values of $J_2(x)$ were computed over $x \in [0,200]$, resulting in a dataset with 2001 points.
- Previous five values of $J_2(x)$ were used as inputs to predict the next value.
- Both Quantum LSTM and Classical LSTM accurately modelled $J_2(x)$.

	Training Loss	Testing Loss
Quantum LSTM	5.51×10^{-5}	1.00×10^{-5}
Classical LSTM	1.10×10^{-6}	1.27×10^{-6}

Table 4: Average training and testing loss comparison for five training repetitions

Prediction of Bessel functions



- Scatter plots show predictions closely align with actual values, with points concentrated on the bisector, indicating good model performance.

Figure 11: Scatter plot with the data on the x-axis and the model predictions on the y-axis for sine function

Prediction of Fully Observable Cart-Pole

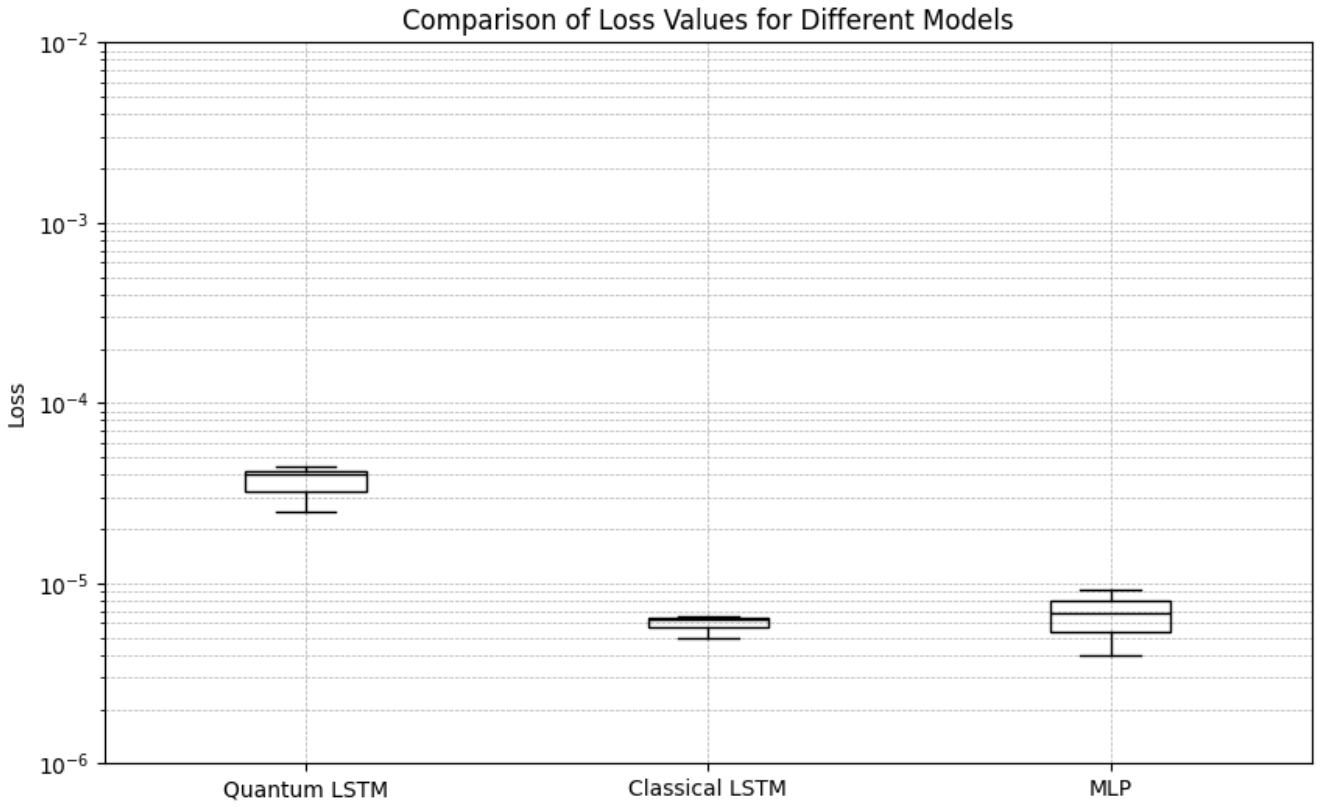


Figure 12: Loss comparison for quantum LSTM, classical LSTM, and MLP for FOCP benchmark

- We observe that classical LSTM and MLP give us the best results followed by quantum LSTM.
- The range of loss values obtained for classical and quantum LSTM is small enough to be considered for rollouts.

Scatter plots – Quantum LSTM

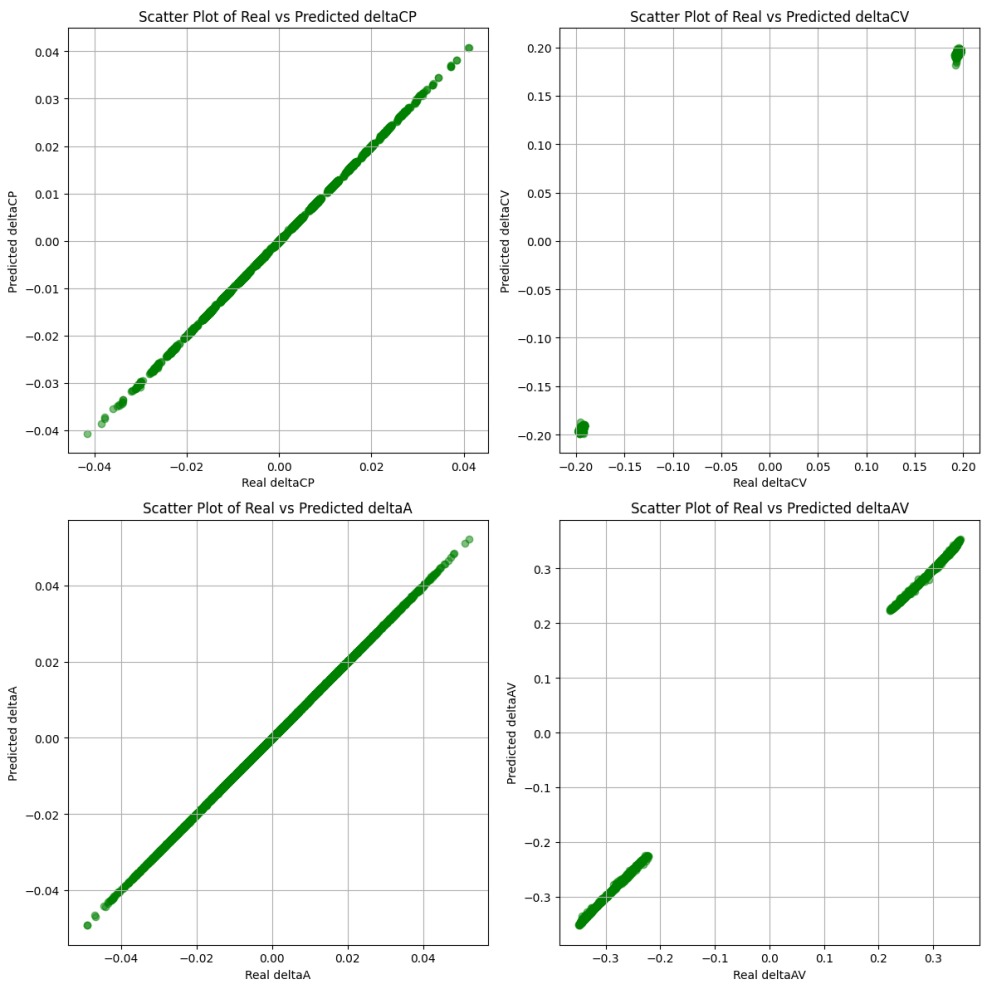


Figure 13: Data on the x-axis and the QLSTM's prediction on the y-axis

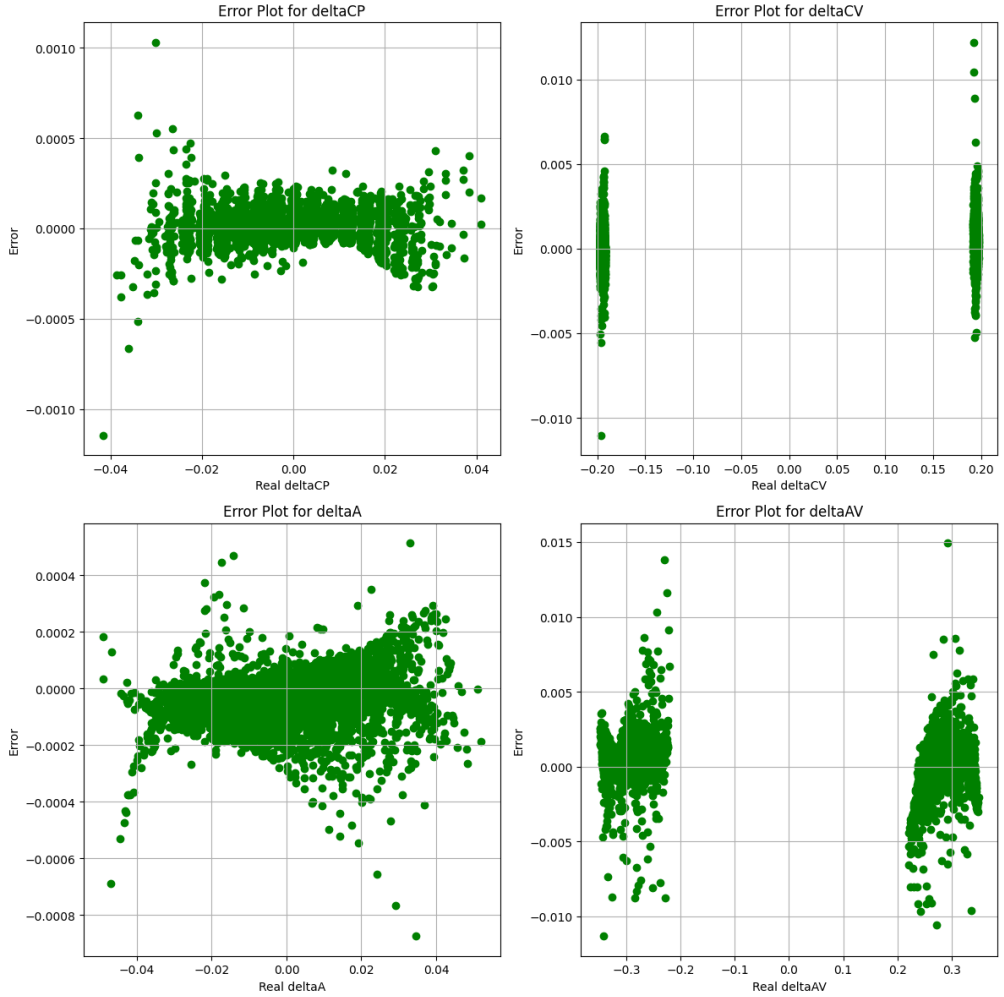


Figure 14: Data on the x-axis and the error in QLSTM's prediction on the y-axis

Scatter plots – Classical LSTM

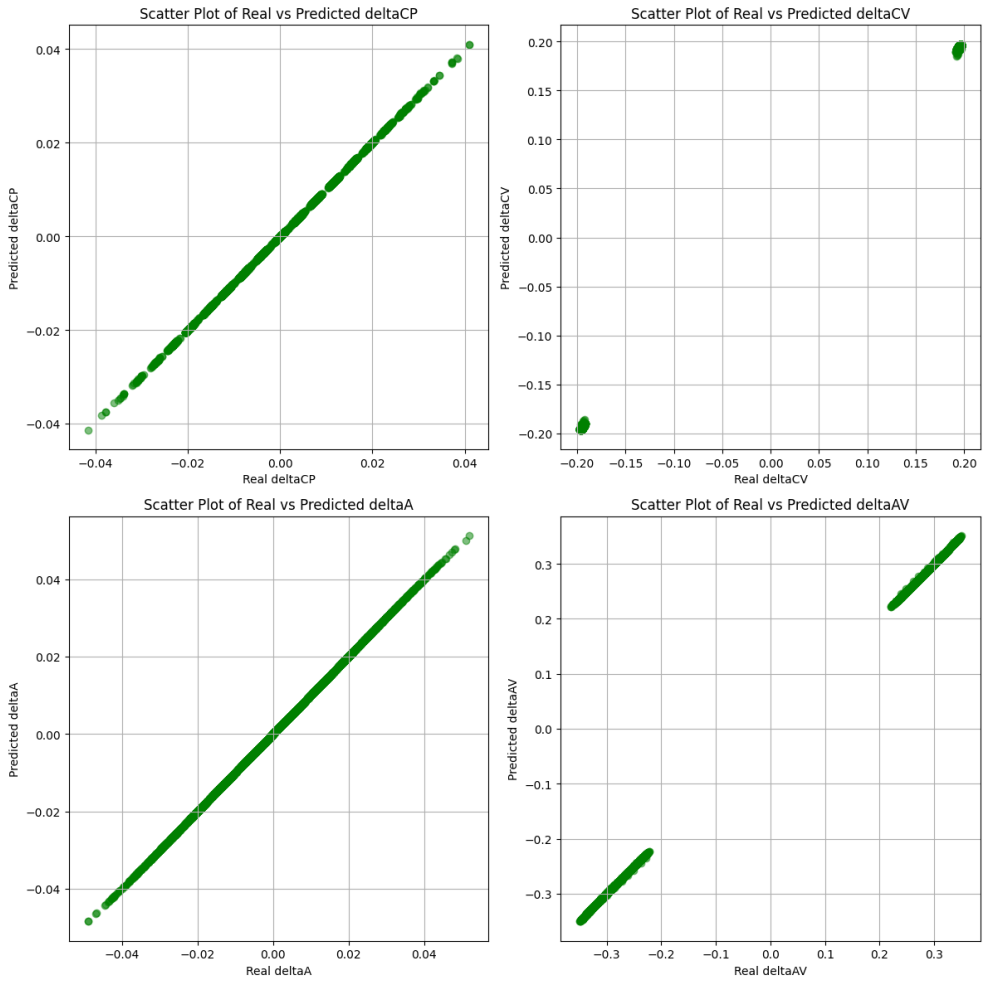


Figure 15: Data on the x-axis and the LSTM's prediction on the y-axis

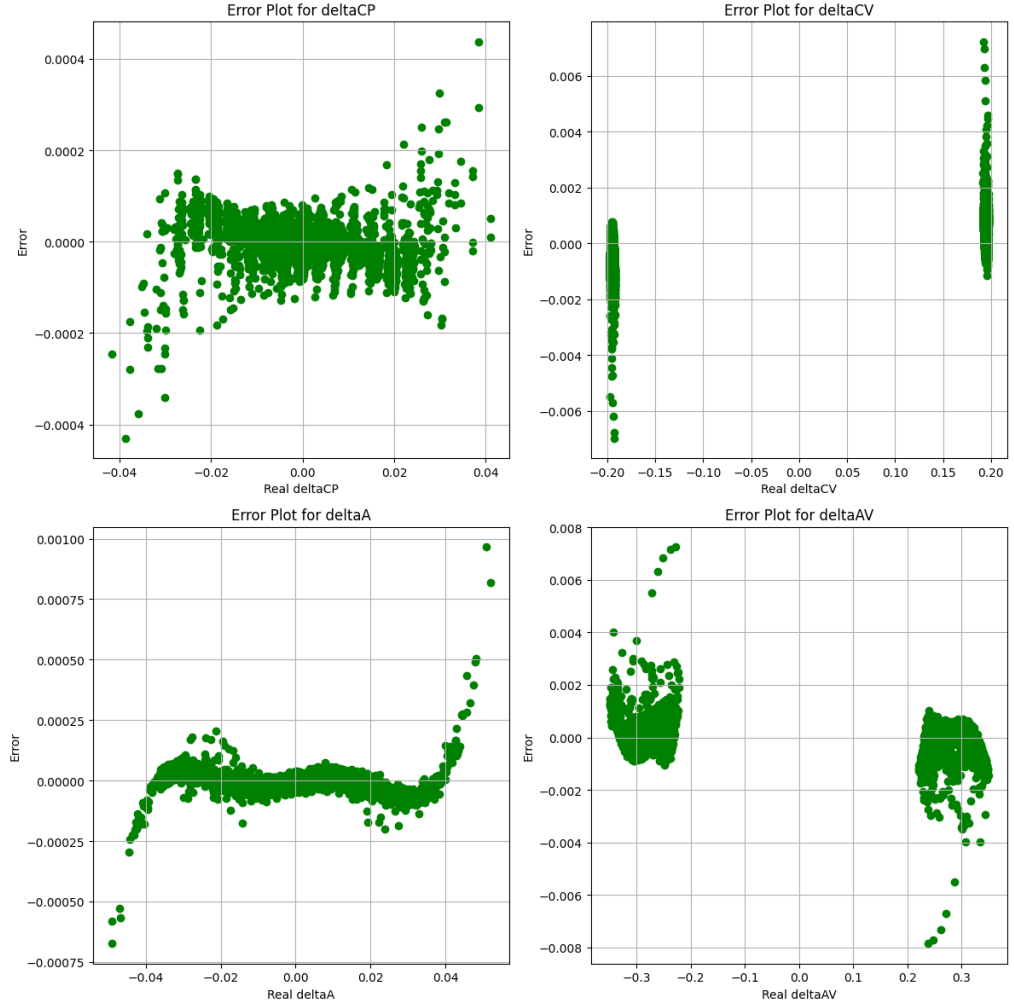


Figure 16: Data on the x-axis and the error in LSTM's prediction on the y-axis

Scatter plots - MLP

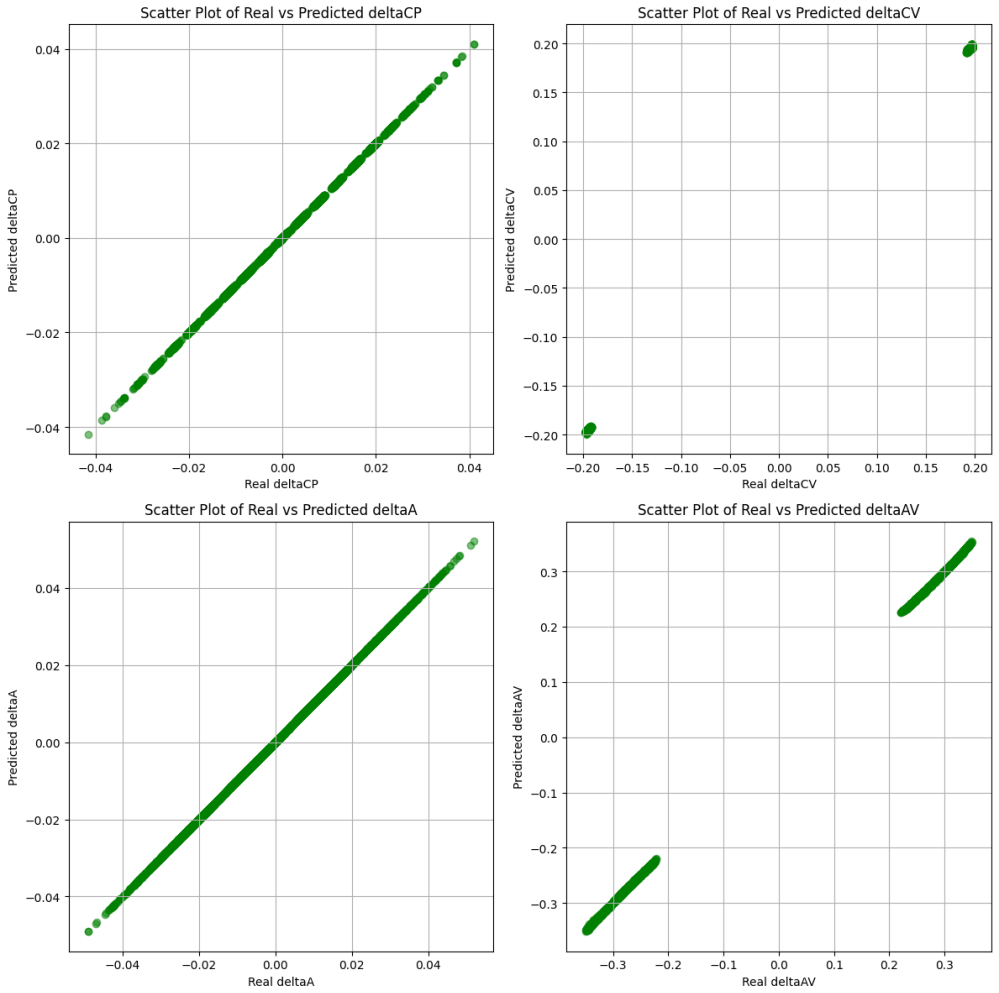


Figure 17: Data on the x-axis and the MLP's prediction on the y-axis

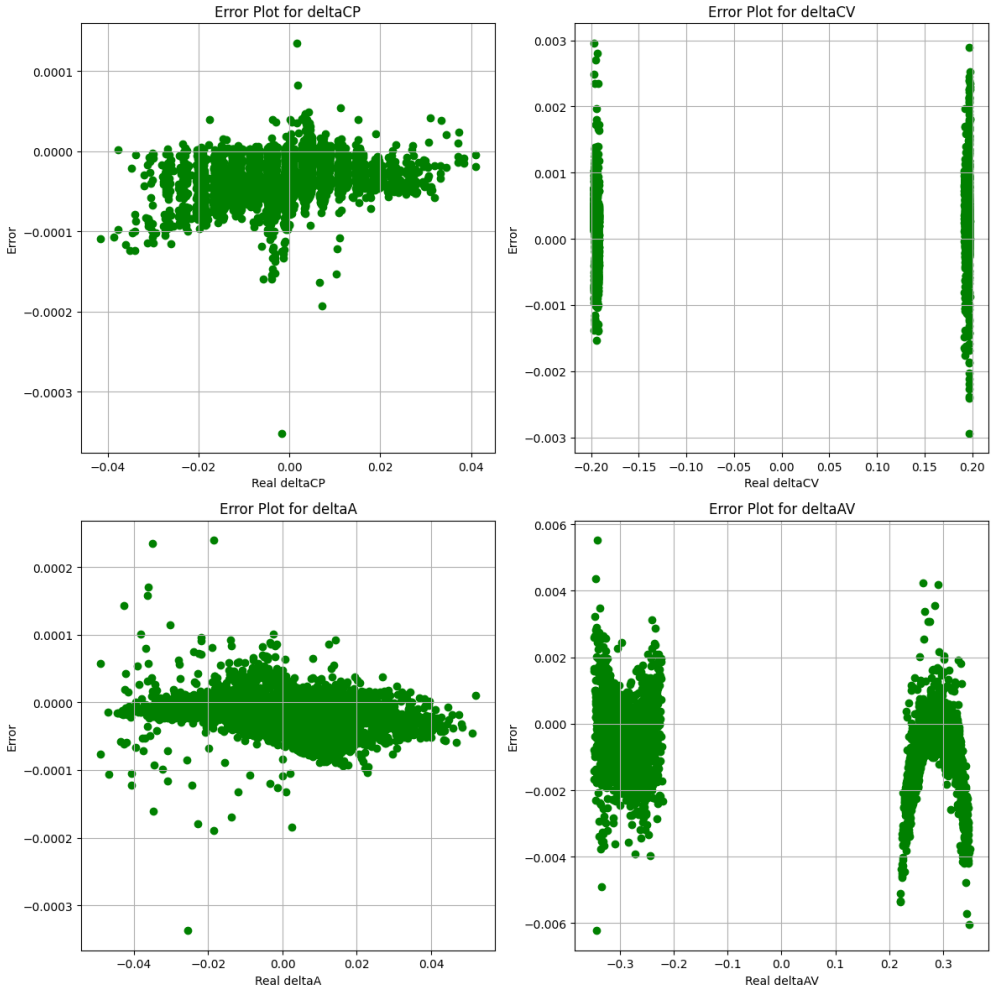


Figure 18: Data on the x-axis and the error in MLP's prediction on the y-axis

Prediction of Fully Observable Cart-Pole

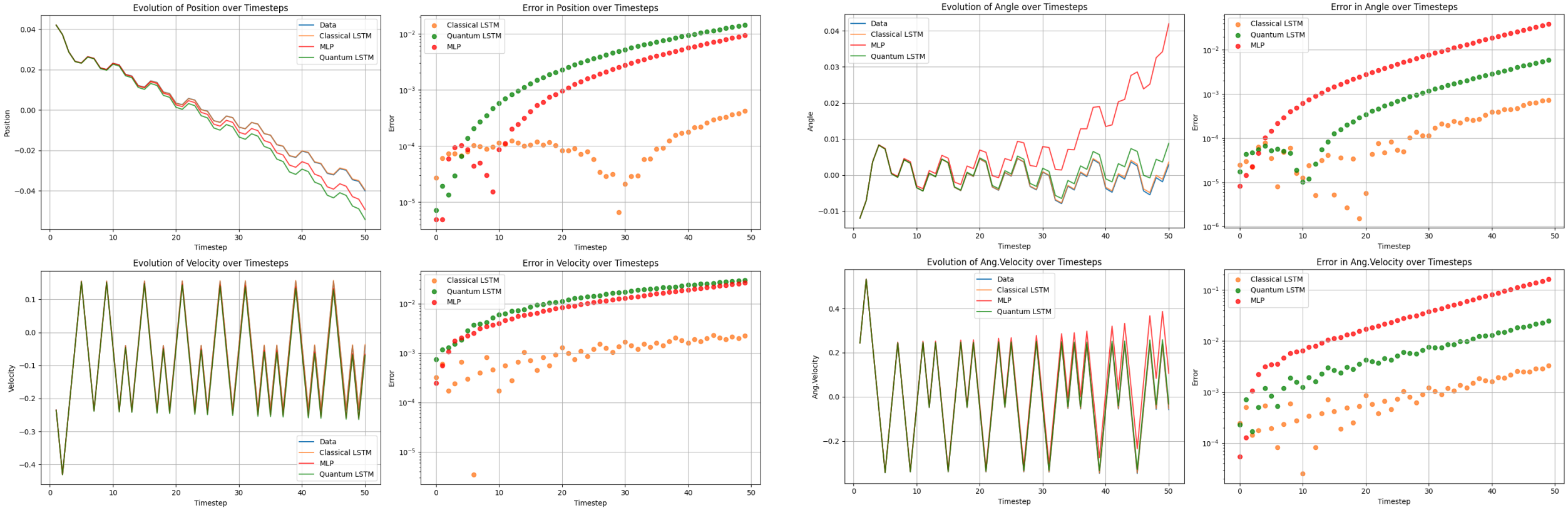


Figure 19: Cart-pole rollouts and their absolute error comparison between classical LSTM, quantum LSTM, and MLP

Prediction of Partially Observable Cart-Pole

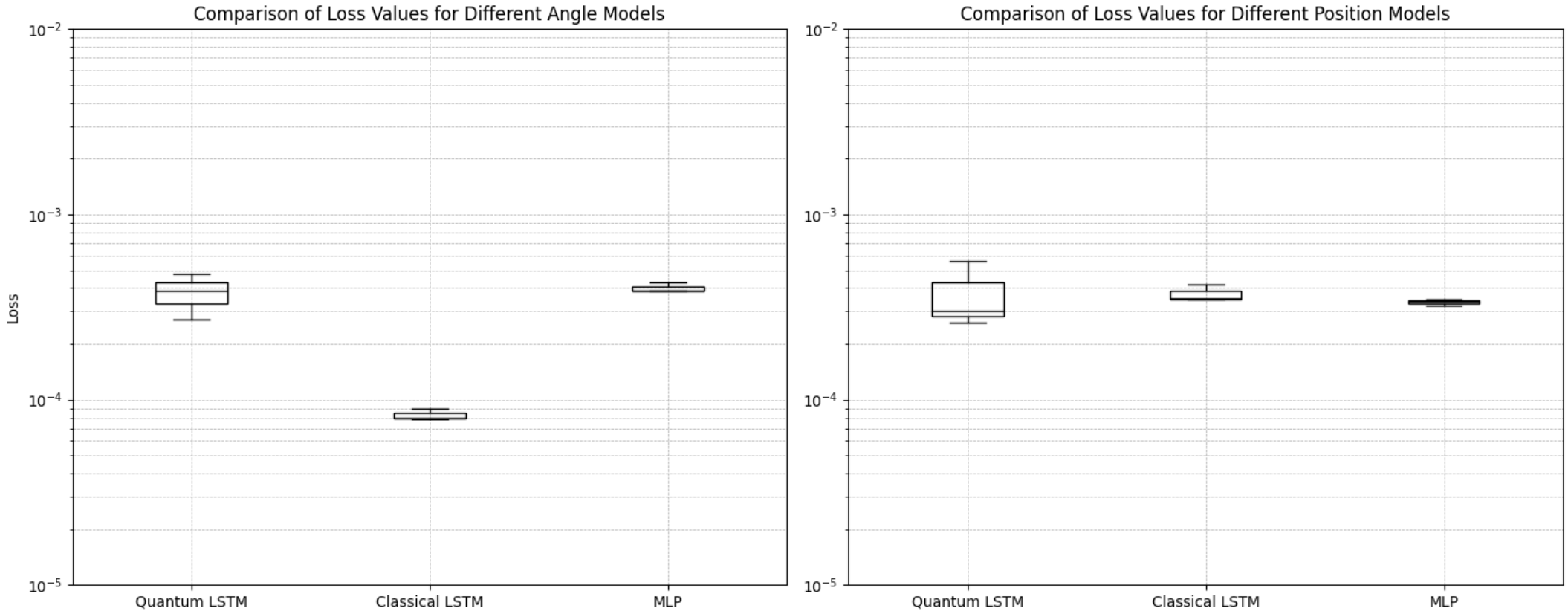


Figure 20: Loss comparison for quantum LSTM, classical LSTM, and MLP for POCP benchmark

Prediction of Partially Observable Cart-Pole

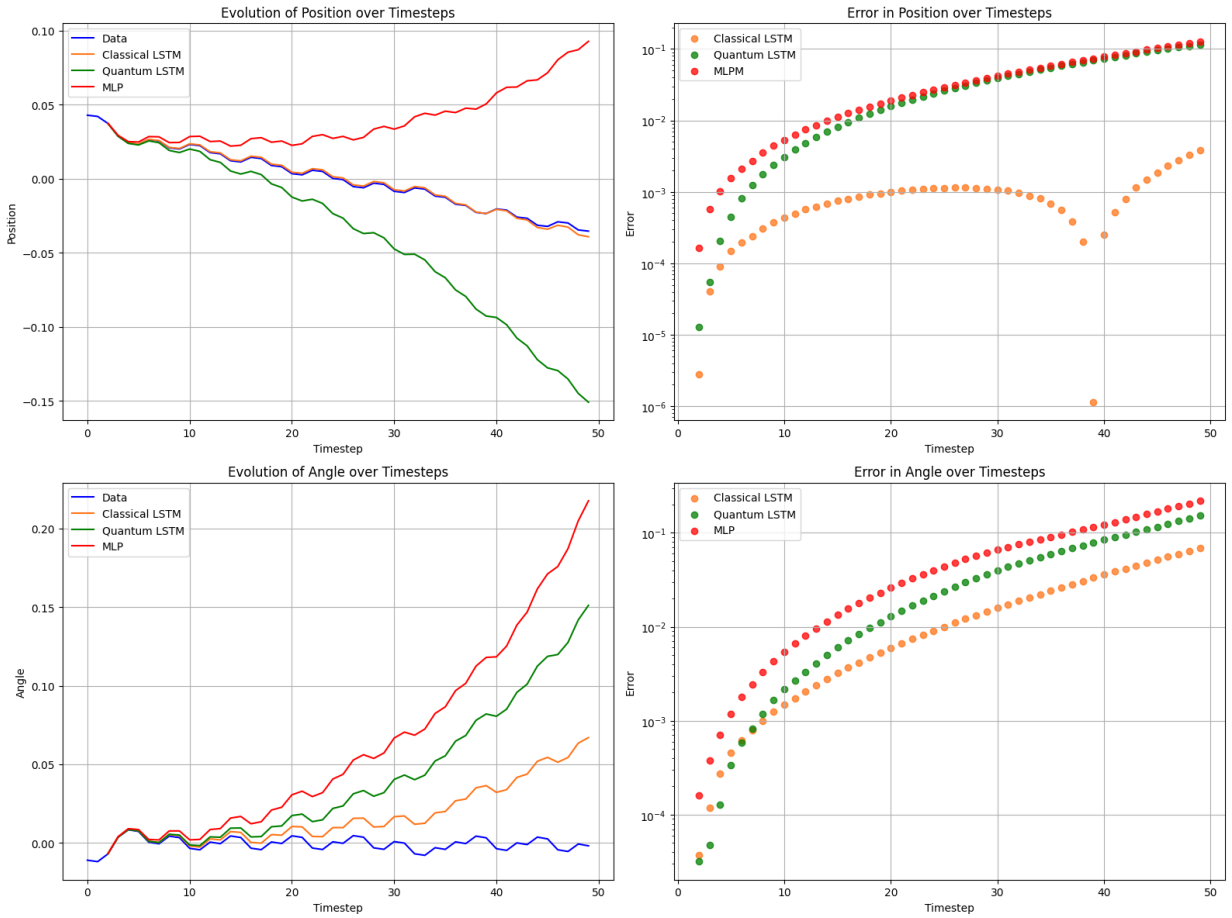
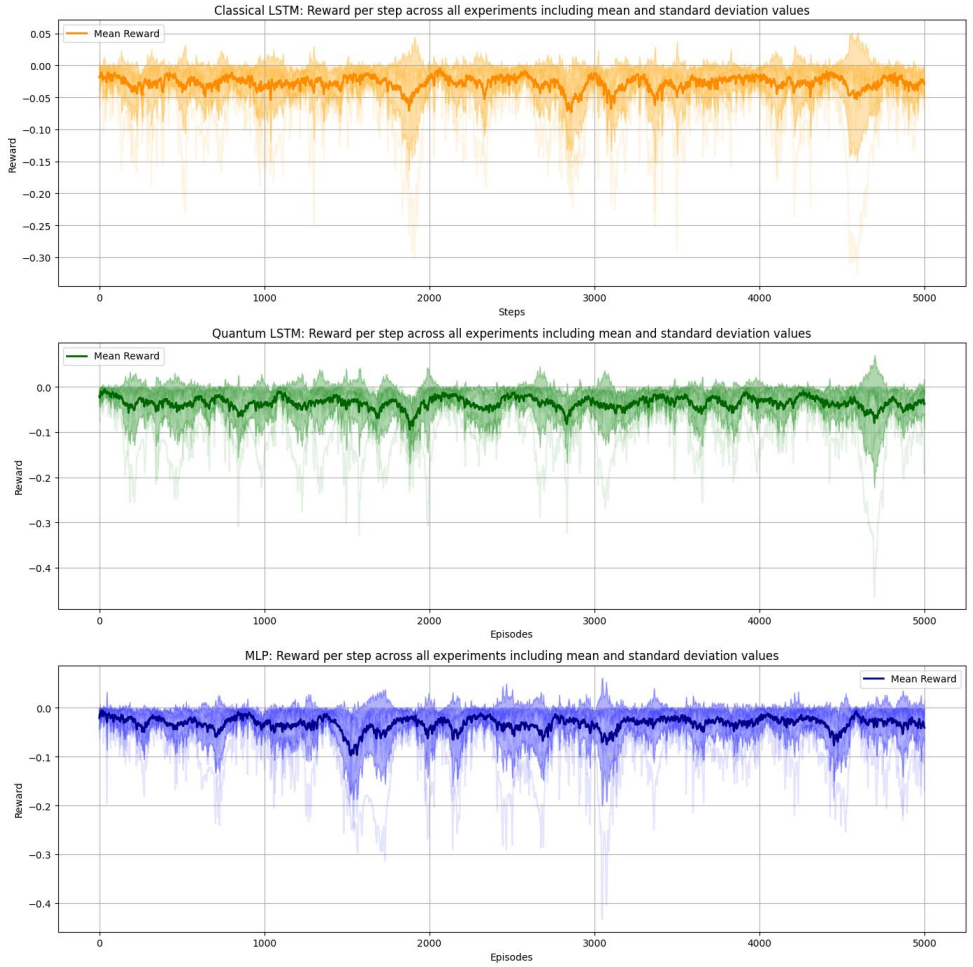


Figure 21: POCP rollouts and their absolute error comparison between classical LSTM, quantum LSTM, and MLP

- Unlike the FOCP case, we now see significant deviations in position and angle for classical LSTM and quantum.
- Up to the first 20 timesteps, the rollouts of the models stay close to the original data. As prediction errors start accumulating, we observe the deviation in the state.
- Classical LSTM does not deviate much from the original position data, unlike the quantum LSTM. However, both the models deviate heavily from the original angle data, with classical LSTM deviating less than quantum LSTM.
- MLP deviates the most in both position and angle.

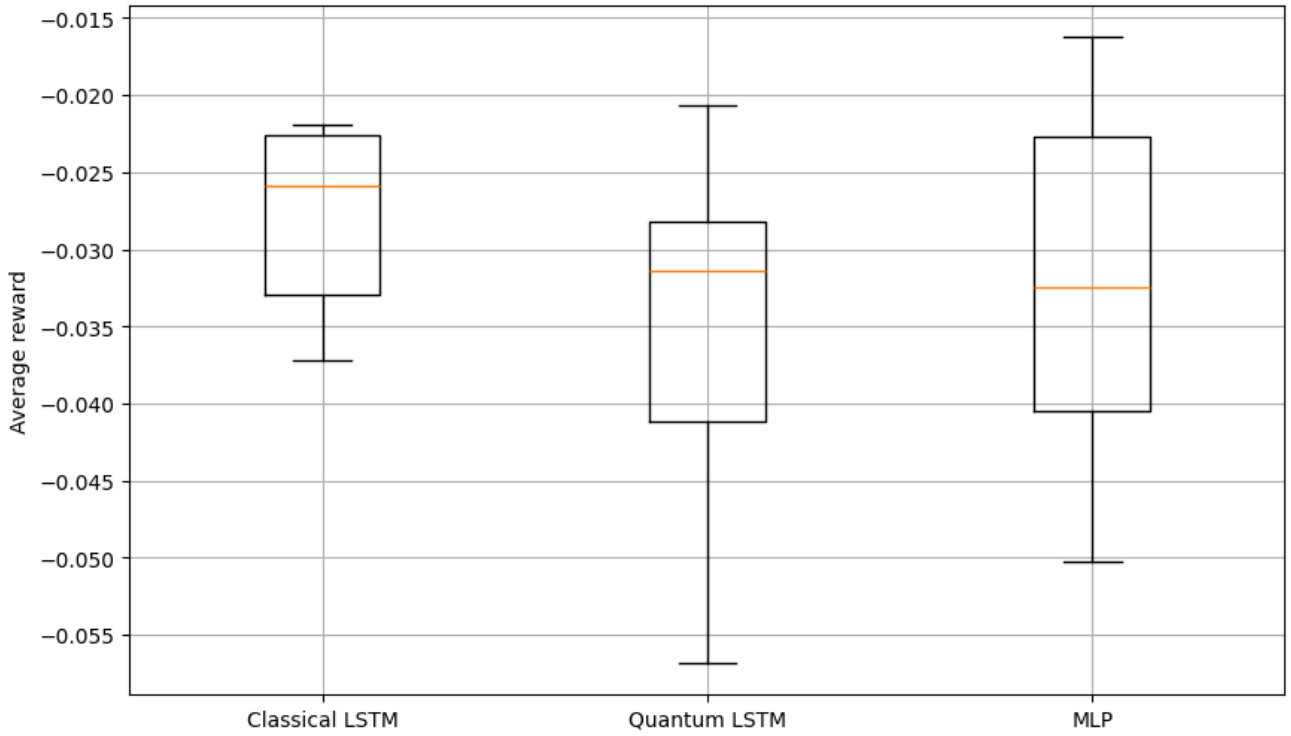
Model Predictive Control for Fully Observable Cart-Pole



- Reward trajectories over 5000 steps show mean values (thicker line) and variability (shaded region) for 10 experiments.
- The reward function penalizes deviations from zero position and angle, keeping the cart-pole balanced near the desired state.
- Regular drifts in reward indicate occasional deviations, but the MPC strategy rebalances the system by penalizing corrective actions.
- 100 particles have been used and 5 PSO iterations have been conducted. Same number of fitness function evaluations were applied for each model.

Figure 22: Visualization of reward per step across all 10 experiments with mean and standard deviation for classical LSTM, quantum LSTM and MLP

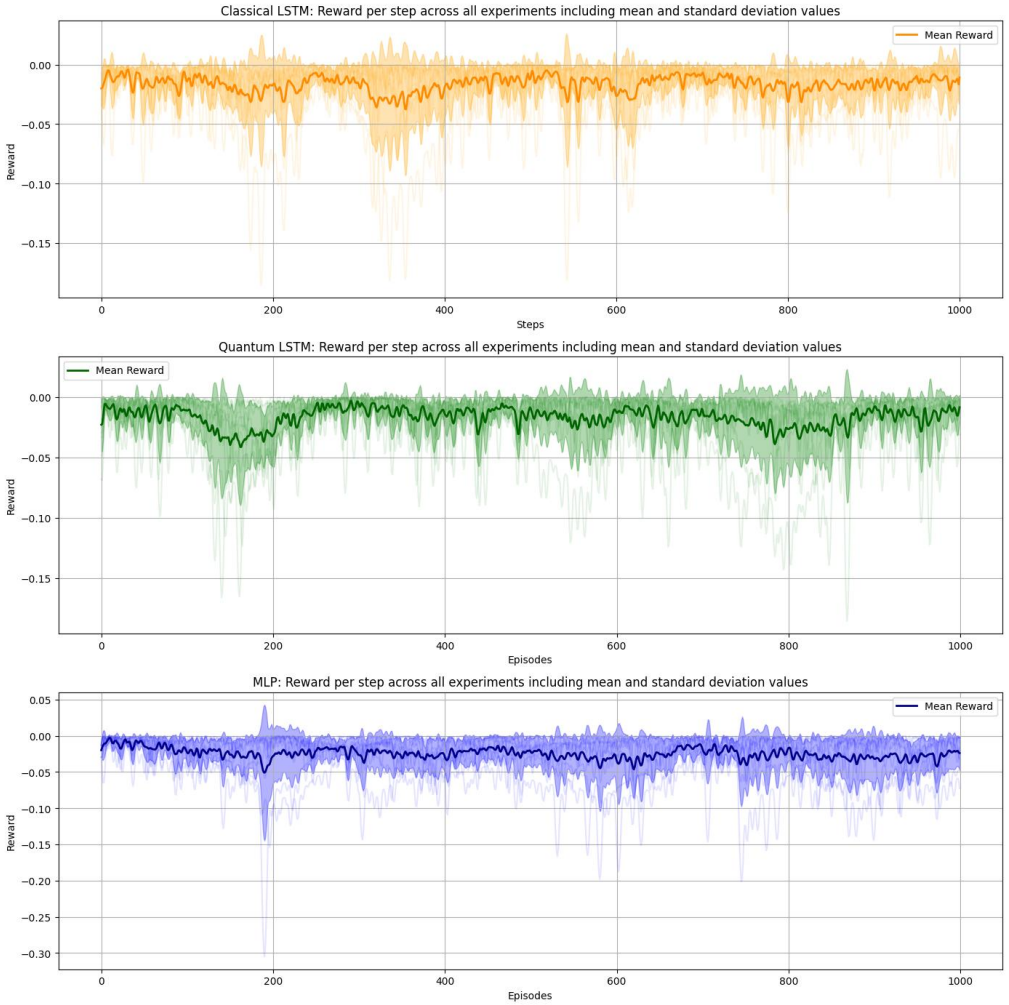
Model Predictive Control for Fully Observable Cart-Pole



- Classical LSTM: Highest average reward with stable performance (narrow IQR).
- Quantum LSTM: Slightly lower rewards with higher variability, likely due to sensitivity to initial conditions.
- MLP: Lowest average reward, indicating suboptimal control.

Figure 23: Boxplots representing the average reward over 5000 timesteps collected from 10 experiments each for classical LSTM, quantum LSTM and MLP

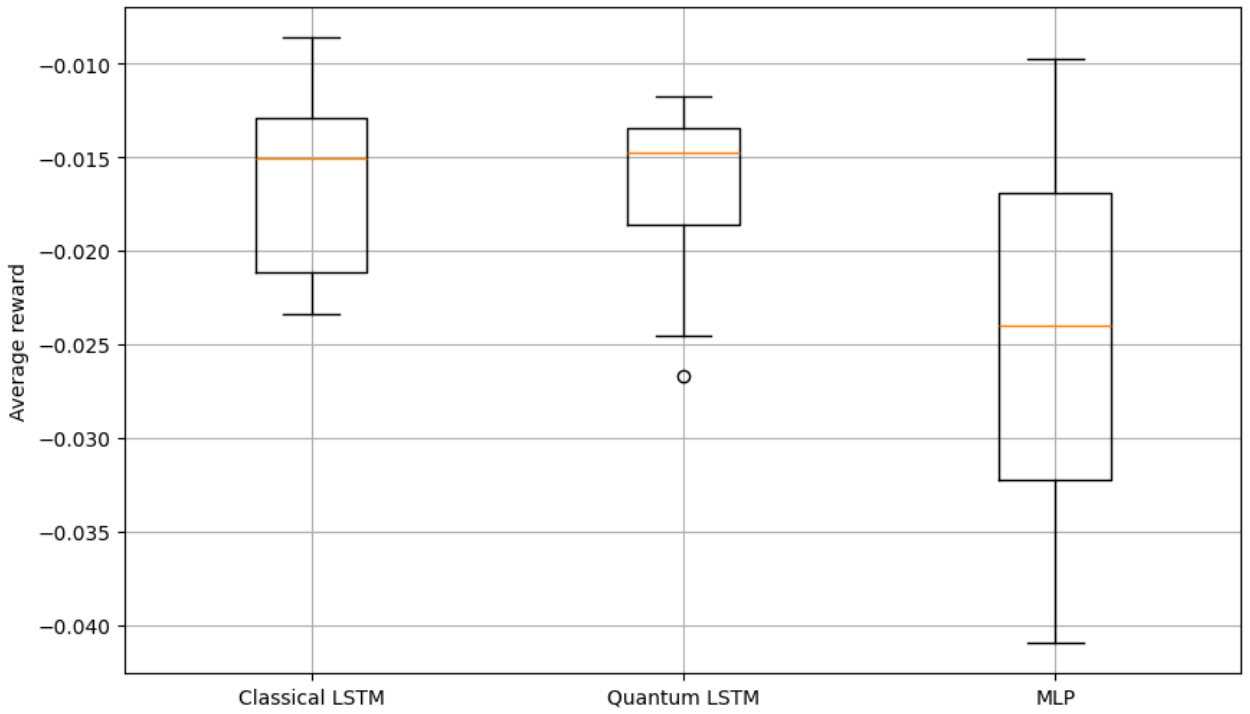
Model Predictive Control for Partially Observable Cart-Pole



- Reward trajectories over 1000 steps show mean values (thicker line) and variability (shaded region) for 10 experiments.
- The reward function penalizes deviations from zero position and angle, keeping the cart-pole balanced near the desired state.
- Regular drifts in reward indicate occasional deviations, but the MPC strategy rebalances the system by penalizing corrective actions.
- 100 particles have been used and 10 PSO iterations have been conducted. Same number of fitness function evaluations were applied for each model.

Figure 24: Visualization of reward per step across all 10 experiments with mean and standard deviation for classical LSTM, quantum LSTM and MLP

Model Predictive Control for Partially Observable Cart-Pole



- Quantum LSTM achieves slightly higher average rewards with a more compact IQR, indicating greater consistency.
- Classical LSTM and MLP show wider IQRs, suggesting higher variability in average rewards across experiments.
- A single outlier in Quantum LSTM's results indicates one instance of unusually low performance.
- Overall, Quantum LSTM demonstrates better reliability in the POCP task compared to Classical LSTM and MLP.

Figure 25: Boxplots representing the average reward over 1000 timesteps collected from 10 experiments each for classical LSTM, quantum LSTM and MLP

Conclusion

- Developed a quantum LSTM-based system identification approach and integrated it into the MPC framework for FOCP and POCP benchmarks.
- Conducted a comparative analysis of quantum LSTM, classical LSTM, and MLP models for system identification and control.
- Quantum LSTM:
  - FOCP Benchmark: Demonstrated potential but with higher variability compared to Classical LSTM.
  - POCP Benchmark: Achieved slightly higher rewards and more stability than Classical LSTM and MLP.
- Demonstrated the effectiveness of PSO for optimizing control actions in both classical and quantum MPC strategies.

Future Outlook

- Optimize Quantum LSTM:
  - Enhance training and inference processes to reduce computational overhead.
  - Explore advanced quantum simulators and backend technologies for improved efficiency.
- Test on Complex Environments:
  - Apply quantum LSTMs to realistic benchmarks like the Industrial Benchmark to demonstrate their potential in real-world scenarios.
- Leverage Advancements in Quantum Hardware:
  - Transition from simulators to real quantum hardware as technology matures.
  - Develop hybrid quantum-classical algorithms for dynamic system identification and control.

References

1. Schwenzer, Max & Ay, Muzaffer & Bergs, Thomas & Abel, Dirk. (2021). Review on model predictive control: an engineering perspective. The International Journal of Advanced Manufacturing Technology. 117. 1-23. 10.1007/s00170-021-07682-3.
2. S. Y. - C. Chen, S. Yoo, and Y.-L. L. Fang. “Quantum Long Short-Term Memory”. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2020), pp. 8622–8626. url:https://api.semanticscholar.org/CorpusID:221470351.
3. S. Y.- C. Chen. Quantum deep recurrent reinforcement learning. 2022. arXiv: 2210.14876 [quant-ph]. url: https://arxiv.org/abs/2210.14876.
4. Chehimi, M., Chen, S. Y. C., Saad, W., & Yoo, S. (2024). Federated quantum long short-term memory (fedqlstm). Quantum Machine Intelligence, 6(2), 43.
5. S. Y. -C. Chen, "Efficient Quantum Recurrent Reinforcement Learning Via Quantum Reservoir Computing," ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Korea, Republic of, 2024, pp. 13186-13190, doi: 10.1109/ICASSP48485.2024.10446089.
6. D. Hein, A. Hentschel, T. Runkler, and S. Udluft. “Particle Swarm Optimization for Model Predictive Control in Reinforcement Learning Environments”. In: Feb. 2018, pp. 401–427. isbn: 9781522551348. doi: 10.4018/978-1-5225-5134-8.ch016.

Additional Slides

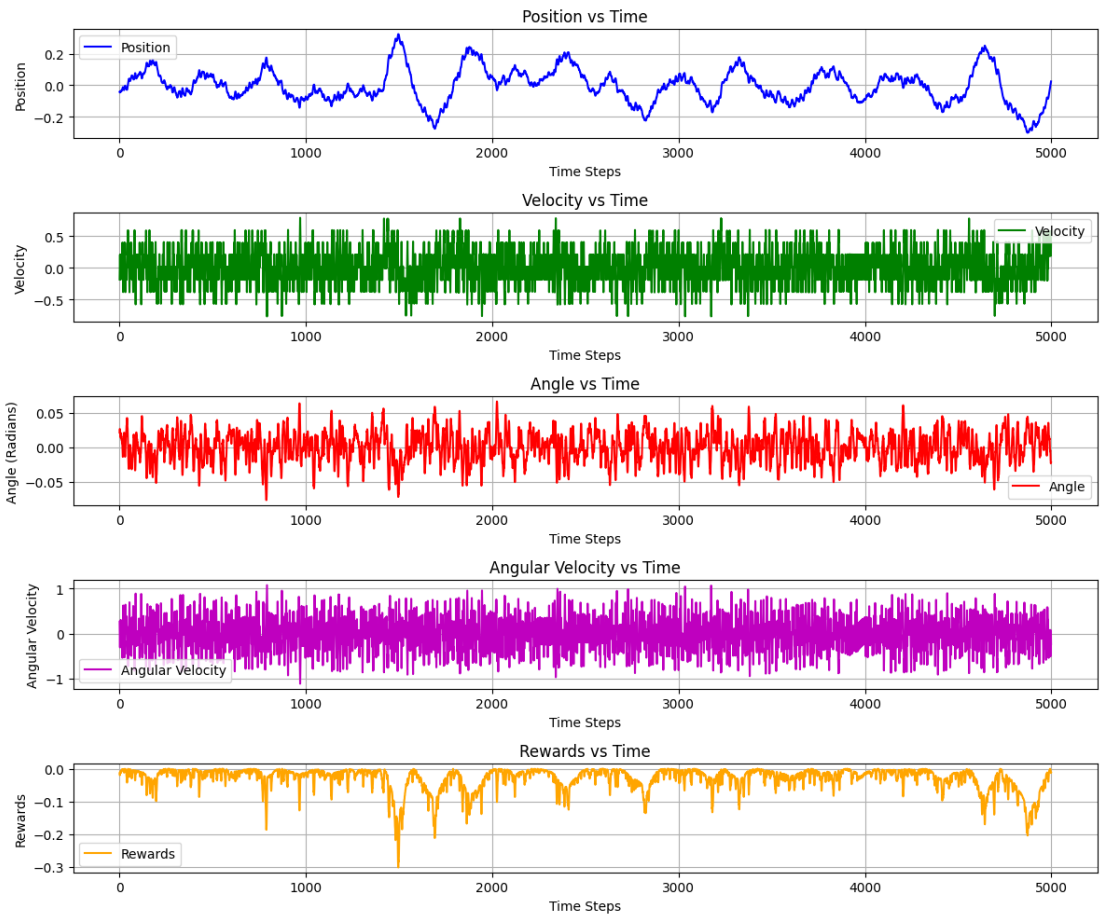


Figure 26: QLSTM balancing

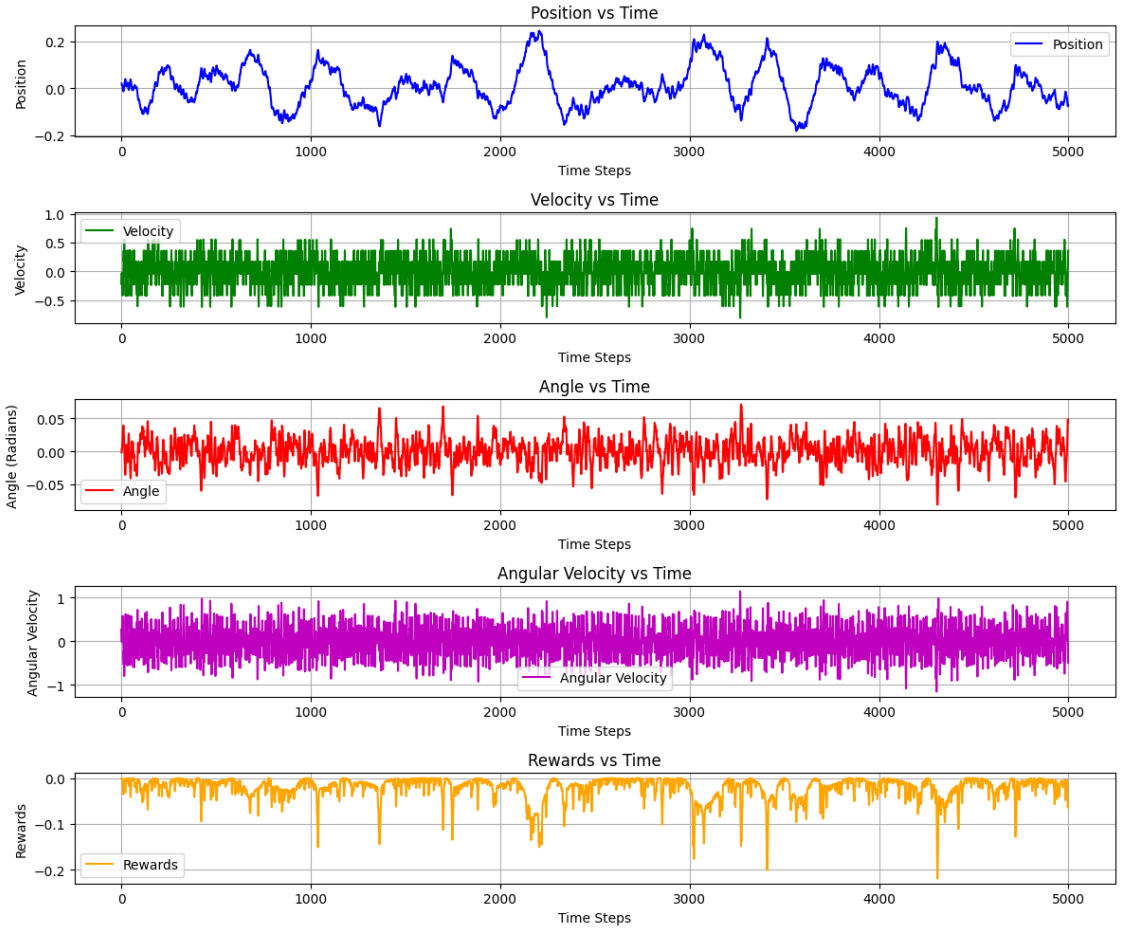


Figure 27: CLSTM balancing

Additional Slides

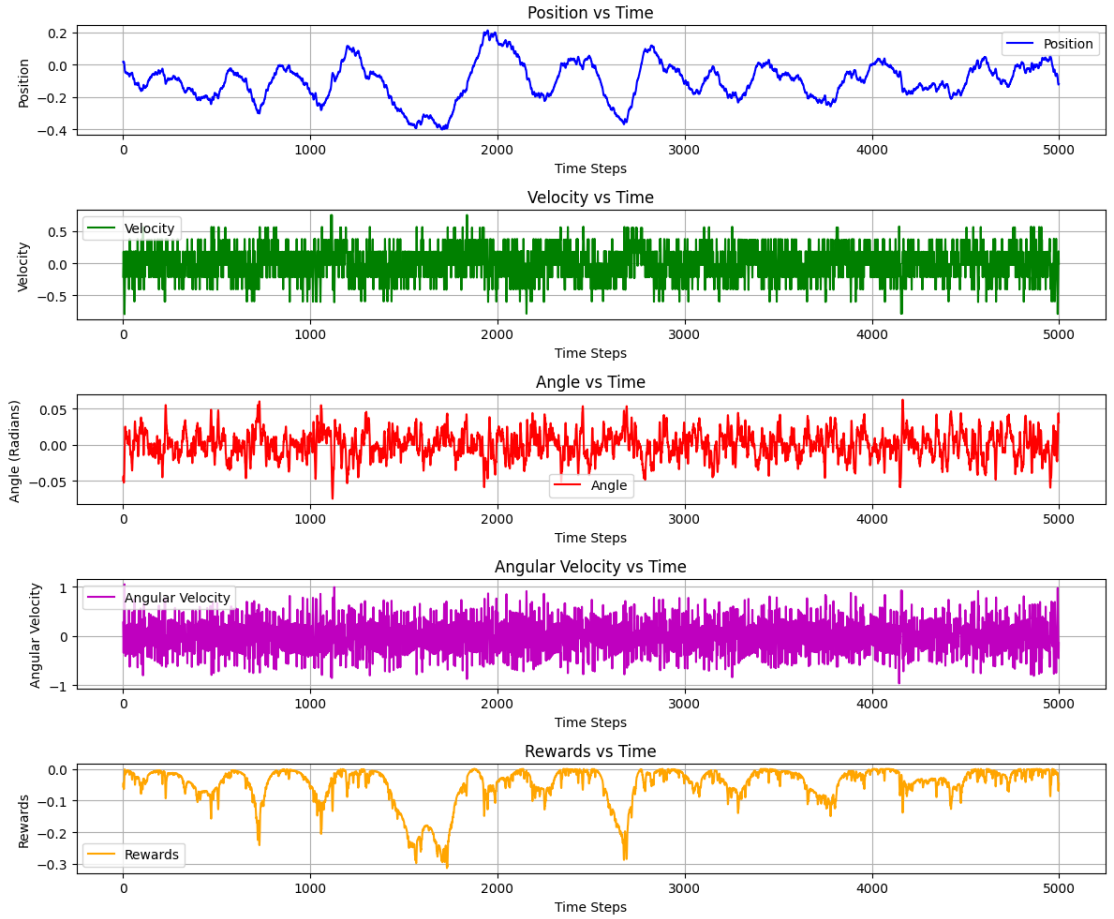


Figure 29: MLP balancing

Additional Slides

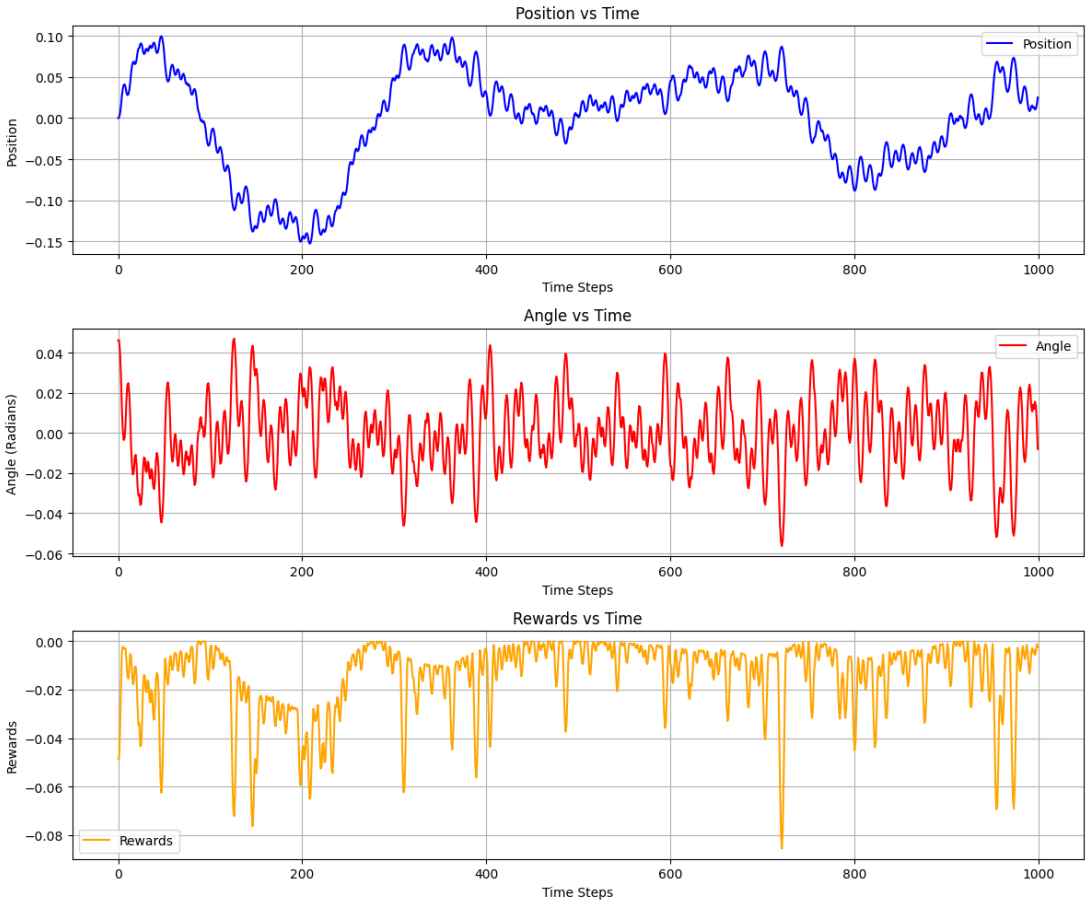


Figure 30: QLSTM balancing

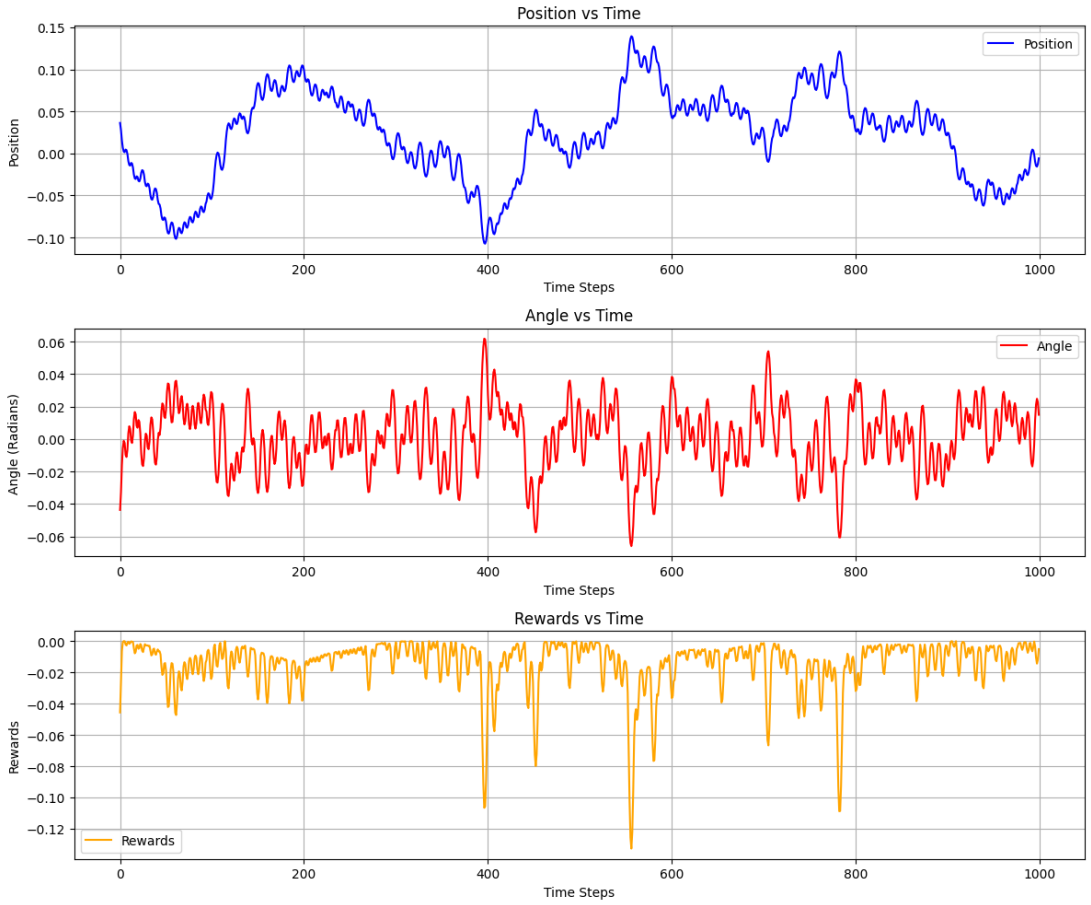


Figure 31: CLSTM balancing

Additional Slides

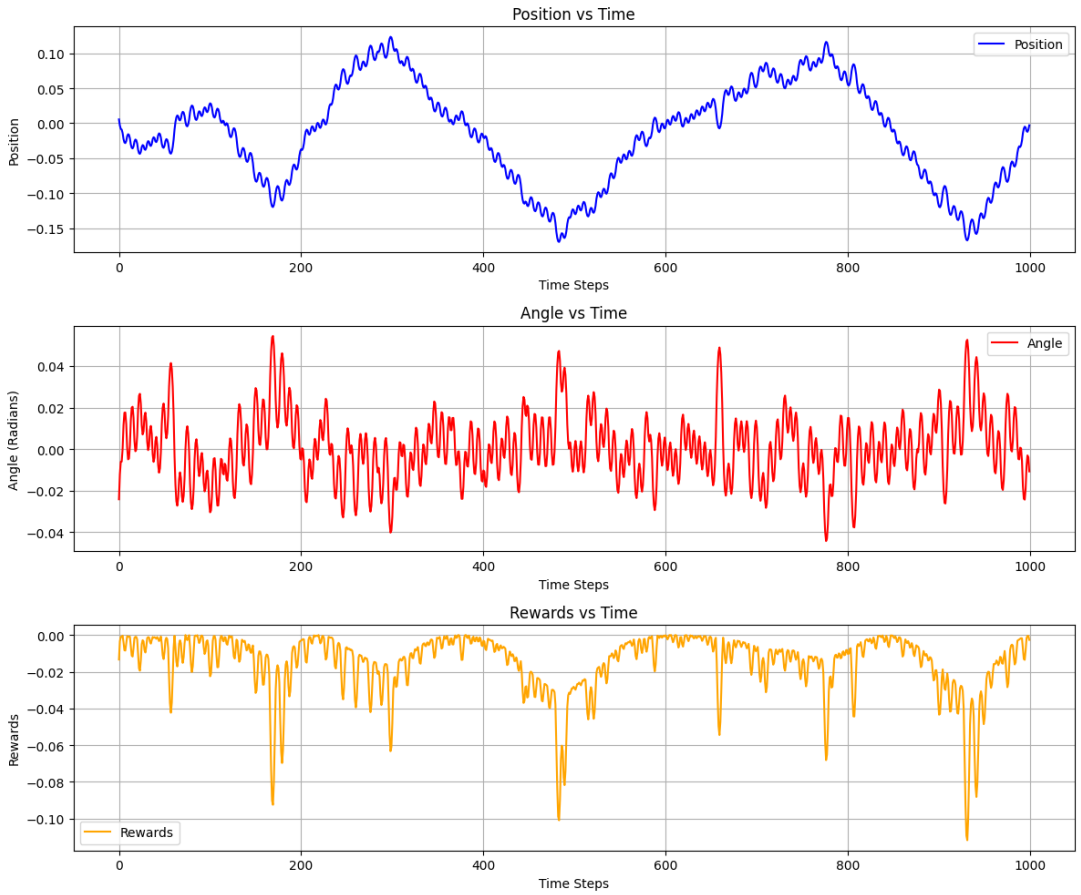


Figure 32: MLP balancing

Additional Slides

	Time taken (in seconds)
Quantum LSTM	240
Classical LSTM	0.5
MLP	0.5

Figure 33: Time taken to optimize single action (FOCP)

	Time taken (in seconds)
Quantum LSTM	720
Classical LSTM	1.5
MLP	1.25

Figure 34: Time taken to optimize single action (POCP)