Assignment: Benchmarking Control Algorithms in `rcognita-edu`

Objective

You are to implement and compare three control strategies for trajectory tracking of a mobile robot:

1. Kinematic (Nominal) Controller
2. Linear Quadratic Regulator (LQR)
3. Model Predictive Control (MPC)

Work within the simulation framework provided in the GitHub repository:

`https://github.com/thd-research/rcognita-edu`

Project Structure and Tasks

1. `systems.py`

Define your system model, e.g. for a differential-drive robot. You may reuse or adapt models from the lecture notebooks.

2. `controllers.py`

Complete or implement the following controllers:

Kinematic Controller: Implement from scratch using the lecture notebooks. Your implementation should compute closed-form control law using polar coordinates and gains $\kappa_\rho, \kappa_\alpha, \kappa_\beta$.

LQR Controller: Implement discrete-time LQR using DARE. Use `scipy.linalg.solve_discrete_are` to compute the gain K and apply control $u_k = -Kx_k$.

MPC Controller: You only need to define the cost function (stage and terminal). The optimization infrastructure is already implemented.

Benchmarking and Experiments

Run simulations using each controller and compare their performance. Later, these will be deployed on real TurtleBots.

Experiment A: Kinematic Controller

- Test three sets of gain parameters $\kappa_\rho, \kappa_\alpha, \kappa_\beta$.
- For each run, record:
  - Robot trajectory vs reference
  - Tracking error over time
  - Control inputs v, ω over time
- Discuss stability and convergence.

Experiment B: LQR Controller

- Vary cost matrices Q, R.
- Test performance under limited input saturation (simulate actuator limits).
- Highlight where LQR fails to stabilize or track.

Experiment C: MPC Controller

- Vary:
  - Horizon length N
  - Weight matrices Q, R, Q_f
  - Terminal cost on/off
- Evaluate:
  - Tracking performance
  - Total accumulated cost
  - Effect of long vs short horizon

Visualization and Reporting

For each controller, provide:

- 2D trajectory plot
- Time plots of tracking error
- Time plots of control inputs
- Accumulated cost comparison (kinematic, LQR, MPC)

Ensure that initial conditions and reference trajectory are consistent across all runs.

Submission and Integration

This is a two-phase assignment:

A. **Simulation Phase:** Complete implementation and benchmarking in the simulator.

B. **Hardware Phase:** Integrate and test controllers with real TurtleBot3.

You are expected to follow best practices in code structure and documentation. Submit via Git pull request. Report submit via iLearn.