



Write Python code to implement model-predictive control for a turbine in a thermal power plant.



Implementing a Model Predictive Control (MPC) for a turbine in a thermal power plant is a complex task that involves creating a mathematical model of the system, defining the control objectives, and tuning the controller parameters. Here's an outline of the steps you need to follow:

1. Create a mathematical model of the turbine system.
2. Define the control objectives and constraints.
3. Implement the MPC algorithm.
4. Tune the controller parameters.

Here's a Python code example using the `cvxpy` library to solve the optimization problem:

python

Copy code

```
import numpy as np
import cvxpy as cp

# Define your system model here (A, B, C, D matrices)
A = np.array([[...]])
B = np.array([[...]])
C = np.array([[...]])
D = np.array([[...]])

# Define the system dimensions
n_x = A.shape[0]
n_u = B.shape[1]

# Define the MPC parameters
horizon = 20
Q = np.diag([...]) # State cost matrix
R = np.diag([...]) # Control cost matrix
x_ref = np.array([...]) # Reference state
u_min = np.array([...]) # Control input lower bounds
u_max = np.array([...]) # Control input upper bounds

# Define the decision variables
x = cp.Variable((n_x, horizon + 1))
u = cp.Variable((n_u, horizon))

# Define the constraints and cost function
constraints = []
cost = 0

for t in range(horizon):
    cost += cp.quad_form(x[:, t] - x_ref, Q) + cp.quad_form(u[:, t], R)
    constraints += [x[:, t + 1] == A @ x[:, t] + B @ u[:, t]]
    constraints += [u_min <= u[:, t], u[:, t] <= u_max]

# Initial state constraint
x_init = np.array([...])
constraints += [x[:, 0] == x_init]

# Final state constraint (if required)
x_final = np.array([...])
constraints += [x[:, horizon] == x_final]

# Set up the optimization problem
problem = cp.Problem(cp.Minimize(cost), constraints)

# Solve the optimization problem
problem.solve()

# Get the optimal control input
optimal_control_input = u[:, 0].value
print("Optimal control input:", optimal_control_input)
```

This code example sets up and solves the optimization problem for a single MPC iteration. You would need to run this code in a loop and update the initial state `x_init` with the current state of the turbine at each time step.

Please note that this code is only a starting point, and you'll need to provide the specific system matrices, constraints, cost weights, and other parameters related to the turbine system for a real-world application.