Introduction

Tasks

- Write a function that forms prediction matrices from given discrete-time state space matrices and a horizon length.
- Write a function that computes the matrices for a quadratic cost function.
- Compute a linear, unconstrained RHC law for the Gantry Crane model.

Framework provided:

myPrediction.m	Template for submission
myCostMatrices.m	Template for submission
myRHC.m	Template for submission
SSmodelParams.mat	Physical parameters for the linear ODE
genCraneODE.p	Answer function for previous assignment
Params_Simscape.mat	Physical parameters for nonlinear simulation
SimscapeCrane_RHC.slx	Simscape model for testing RHC performance
testMyRHC.m	Function to test your RHC performance
GantryResponsePlot.m	Utility function

Approach

- Edit **myPrediction.m** to generate prediction matrices Φ (`Phi`) and Γ (`Gamma`) over a horizon of length N such that

$$\mathbf{x} := \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} = \Phi x_0 + \Gamma \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix}, \tag{1}$$

  where x_i and u_i are state and input vectors at time instant i related by the discrete-time update equation $x_{i+1} = Ax_i + Bu_i$, $i = 0, \ldots, N-1$.

  Your code should be written for the general case and not just for the gantry crane.

- Edit **myCostMatrices.m** to compute the matrices G and H in

$$\mathbf{u}^*(x_0) := \arg\min_{\mathbf{u}} \frac{1}{2}\mathbf{u}^T H \mathbf{u} + x_0^T G^T \mathbf{u} = \arg\min_{\mathbf{u}} \; x_N^T P x_N + \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right) \text{ s.t. (1)},$$

  where P, Q and R are given positive semi-definite matrices and $\mathbf{u} := [u_0^T \; u_1^T \; \cdots \; u_{N-1}^T]^T$.

  Your code should be written for the general case and not just for the gantry crane.

- Edit **myRHC.m** so that it computes the RHC control law gain matrix K such that

$$u_0^*(x_0) = Kx_0,$$

where

$$\mathbf{u}^*(x_0) =: [u_0^*(x_0)^T \ u_1^*(x_0)^T \ \cdots \ u_{N-1}^*(x_0)^T]^T.$$

Your code should be written for the general case and not just for the gantry crane.

- Open and run **testMyRHC.m**. It will take a short while to finish since it runs the **SimscapeCrane_RHC.slx** Simulink model twice. This model implements two controllers: a PI and a linear control gain K. The test script runs the model for each controller, saves the state data and computes the settling time of the controller.

  Use this script for observing and modifying the behaviour of your controller. In particular, explore what happens when you vary the sample period, horizon length and stage cost weight matrices. This will be helpful for the final assignment.

Submission

filename	Description.
myPrediction.m	Generates prediction matrices Φ and Γ for given discrete-time A, B and N
myCostMatrices.m	Computes the cost function matrices H and G for given Γ, Φ, P, Q and R
myRHC.m	Computes control law gain K for given cost matrices and number of inputs

Edit or upload the Matlab template for the three files above into Cody Coursework.