Problem 1. Note that from the read-out map we have, $y^T \tau Q y = x^T \tau C^T QC x \tau$. Thus the original LQR problem can be re-written as:

$$\min_{x,u} N-1 \sum_{\tau=0} \left( x^T \tau \bar{Q} x \tau + u^T \tau Ru \tau \right),$$

which is the standard LQR problem with $\bar{Q}$ as the state penalty matrix and $Q_f = 0$. The optimal cost-to-go and the optimal control at time $t$ are thus given by:

$$J^*_t(z) = z^T P_t z$$
$$u^*_t = -K_t z,$$

where $t \in \{0, 1, \ldots, N-1\}$ and

$$P_t = \bar{Q} + K_t^T R K_t + (A - BK_t)^T P_{t+1} (A - BK_t), \quad P_N = 0$$
$$K_t = (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A.$$

Problem 2. We can code up the optimal control policy derived in Problem-1 to analyze the system. Here are the plots for the optimal control, output and cost-to-go for the three cases:

![Figure 1: Used control authority](image-url)
Since the output penalty is higher in case (ii), output is quickly driven to zero compared to the other two cases (see Figure 2). To do so, a higher control authority is used, as evident from Figure 1. On the other hand, when input penalty is higher, the control becomes very expensive. Thus, used control magnitude is very small as evident from Figure 1. As a result, output is not driven to zero even by the end of the horizon.

Problem 3. (a) The optimal control of the infinite horizon problem is given by

\[ u(t) = -\frac{1}{\rho} B^T P x(t), \]  

(4)
where $P$ is the unique positive definite matrix that solves the continuous-time algebraic Riccati equation

$$A^T P + PA - \frac{1}{\rho} PBB^T P + C^T C = 0. \quad (5)$$

Let $P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}$. We can rewrite the Riccati matrix equation as the following set of scalar equations:

$$1 - \frac{1}{\rho}p_2^2 = 0$$
$$2p_2 - \frac{1}{\rho}p_3 = 0$$
$$p_1 - \frac{1}{\rho}p_2p_3 = 0$$

Solving this set of equations, we have $p_1 = \sqrt{2}\rho^{1/4}$, $p_2 = \sqrt{\rho}$, and $p_3 = \sqrt{2}\rho^{3/4}$. Therefore, the optimal control is

$$u(t) = -\rho^{-1/2}x_1(t) - \sqrt{2}\rho^{-1/4}x_2(t)$$

(b) The closed-loop system is given by

$$\dot{x} = Ax + Bu = \begin{bmatrix} 0 & 1 \\ -\rho^{-1/2} & \sqrt{2}\rho^{-1/4} \end{bmatrix} x =: \tilde{A}x. \quad (6)$$

The eigenvalues of $\tilde{A}$ are $\lambda = 2^{-1/2}\rho^{-1/4}(-1 \pm j)$. The first thing to note is that the system is stable since the eigenvalues are in the left half plane. Moreover, we see that for increasing $\rho$ the eigenvalues approach the imaginary axis (in fact, the origin) and the system response becomes slower. Conversely, as $\rho \to 0$, the eigenvalues move far left in the complex plane and the system response becomes faster. This is intuitive, as $\rho$ is the control weighting matrix, so a low $\rho$ corresponds to a “cheap” control. In this case the state cost dominates, and the optimal controller tries to drive the state to zero fast using a high gain.

**Problem 4.** Before we can design a controller, we must first derive the dynamics of this system. Let us consider the states of the system as $x$ and $\dot{x}$. We will denote the state vector as $z = [x \ \dot{x}]^T$. The system dynamics can be written as:

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

Having found our $A$ and $B$ matrices, we need only to define our cost matrices $Q$ and $R$ in order to use MATLAB, python, etc. to solve for our optimal feedback control law (we will approach this problem as infinite-horizon, continuous-time LQR). When we want to drive the system to the origin, the optimal feedback policy is given by $u(t) = -Fz(t)$ (where $F$ is computed as $R^{-1}B^T P$, with $P$ solving the continuous time algebraic Riccati equation). If we want to guide to some nonzero final position $xf$ such that $\dot{x} = 0$, we can apply the feedback law $u(t) = -F \left( z(t) - [xf \ 0]^T \right)$. We can make this shift for this particular system because the dynamics do not have a dependency on position ($x$ appears in neither the equation for $\dot{x}(t)$ nor the equation for $\ddot{x}(t)$).
Here is an example with cheap control ($Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $R = \begin{bmatrix} 0.1 \end{bmatrix}$, $z_0 = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$)

Here is an example with expensive control ($Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $R = \begin{bmatrix} 20 \end{bmatrix}$, $z_0 = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$)