1. (a) If we assume that our model of the physical system agrees with its true behavior, increasing the MPC prediction horizon N results in:

    - improved performance, since the optimal value of predicted cost is non-increasing with increasing N;
    - larger operating region, since the set of recursively feasible initial conditions is non-decreasing with increasing N;
    - higher computational load, since the number of optimization variables increases

    Hence determining N involves a trade-off between system performance and computational cost.

    If the model used by the MPC controller agrees poorly with the actual physical behavior, then the performance can actually decrease with increased prediction horizon.

    (b) We refer to Δ as a *slack variable*. It helps to avoid infeasibility of the optimization problem, at the cost of an additional decision variable in the optimization problem (and hence, additional computational cost); it can sometimes also be a disadvantage that constraints may be violated in favor of improved performance (but with the appropriate tuning of the slack variables in the optimization problem, this should not happen).

    (c) Q_1 penalizes state deviations from equilibrium, while Q_2 penalizes the control effort. Increasing Q_1 results in an aggressive control behaviour in order to bring the state rapidly to rest. On the other hand, increasing Q_2 yields smaller control (but also slower convergence).

    (d) Terminal constraints ensure that it is possible to find a control such that the predictions satisfy the system input and state constraints at all future times. This allows to guarantee recursive feasibility of the MPC problem, and (by the appropriate selection of terminal penalty) also to certify asymptotic stability.

    A terminal constraint set S must be:

    - invariant under some feedback law N to ∞, i.e., $x_k \in S$ implies $x_{k+1} \in S$, for all $k \geq N$.
    - feasible, i.e., state constraints are instantaneously satisfied and input constraints are satisfied by the feedback for every $x \in S$.

2. (a) *Open loop* stability is not affected by the B matrix and hence does not depend on the parameter b. To determine if the system is stable, we compute the eigenvalues of A as the roots of its characteristic polynomial:

$$p(\lambda) = \det \lambda I - A = \det \begin{bmatrix} \lambda & 1/2 \\ -3/2 & \lambda + 2 \end{bmatrix} =$$
$$= \lambda^2 + 2\lambda + 3/4 = (\lambda + 1/2)(\lambda + 3/2)$$

Hence, A has eigenvalues $\lambda = -1/2$ and $\lambda = -3/2$. Since one of the eigenvalues have magnitude greater than one, the system is open-loop unstable.

We thus conclude that the system is not open-loop stable for any values of the parameter b.

(b) The controllability matrix is

$$W_c = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 1 & -b/2 \\ b & 3/2 - 2b \end{bmatrix}$$

which looses rank when $\det W_c = 0$, *i.e.* when

$$\det W_c = b^2/2 - 2b + 3/2 = \frac{1}{2}(b-3)(b-1) = 0$$

Hence, the system is controllable, unless $b = 1$ or $b = 3$.

(c) With $b = 0$, the closed-loop dynamics is

$$x_{t+1} = A_c x_t$$

where

$$A_c = A - BL = \begin{bmatrix} 0 & -1/2 \\ 3/2 & -2 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} l_1 & l_2 \end{bmatrix} = \begin{bmatrix} -l_1 & -1/2 - l_2 \\ 3/2 & -2 \end{bmatrix}$$

The matrix A_c has characteristic polynomial

$$p(\lambda) = (\lambda + l_1)(\lambda + 2) + 3/2(1/2 + l_2) = \lambda^2 + (2 + l_1)\lambda + 2l_1 + 3/4 + 3/2 l_2$$

To place both eigenvalues at $\lambda = 0$, we need $p(\lambda) = \lambda^2$, *i.e.*

$$\begin{aligned} l_1 &= -2 \\ l_2 &= 13/6 \end{aligned}$$

With this feedback, the closed-loop system matrix A_c will become nilpotent. In particular,

$$A_c^2 = 0$$

so the system state will converge to zero in no more that two steps.

(d) The error dynamics has the form $e_{t+1} = A_e e_t$ with

$$A_e = A - KC = \begin{bmatrix} 0 & 0 \\ 3/2 & 0 \end{bmatrix}$$

Since the matrix is lower triangular, we can immediately see that all eigenvalues are at the origin, so the system is indeed stable. To find a Lyapunov function, we need to find a positive definite matrix P which satisfies

$$A_e^T P A_e - P = -Q$$

We let $Q = I$ and parameterize the unknown matrix P as

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}$$

The Lyapunov equation becomes

$$\begin{bmatrix} 9p_{22}/4 - p_{11} & -p_{12} \\ -p_{12} & -p_{22} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

which has solution

$$P = \begin{bmatrix} 13/4 & 0 \\ 0 & 1 \end{bmatrix}$$

The matrix is clearly positive definite (diagonal matrix with positive entries), hence the system is asymptotically stable.

3. (a) We can solve this problem either using DP or via batch optimization. Let us first consider the batch approach:

At time 0, we cannot do anything about the current state x_0, so we would like to find the control sequence $\{u_0, u_1\}$ which minimizes

$$J = 10 \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}^T \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Since $x_{t+1} = 1.5x_t + u_t$, we have

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 2.25 \end{bmatrix} x_0 + \begin{bmatrix} 1 & 0 \\ 1.5 & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}$$

Hence,

$$\begin{aligned} J = &\begin{bmatrix} u_0 \\ u_1 \end{bmatrix}^T \left(10I + \begin{bmatrix} 1 & 0 \\ 1.5 & 1 \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ 1.5 & 1 \end{bmatrix} \right) \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} + \\ &+ 2x_0^T \begin{bmatrix} 1.5 \\ 2.25 \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ 1.5 & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} + x_0^T \begin{bmatrix} 1.5 \\ 2.25 \end{bmatrix}^T \begin{bmatrix} 1.5 \\ 2.25 \end{bmatrix} x_0 = \\ &= \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}^T \begin{bmatrix} 13.25 & 1.5 \\ 1.5 & 11 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} + 2x_0^T \begin{bmatrix} 4.875 \\ 2.25 \end{bmatrix}^T \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} + 7.3125 x_0^T x_0 \end{aligned}$$

By the first-order optimality conditions, the optimal control is

$$\begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = - \begin{bmatrix} 13.25 & 1.5 \\ 1.5 & 11 \end{bmatrix}^{-1} \begin{bmatrix} 4.875 \\ 2.25 \end{bmatrix} x_0 \approx - \begin{bmatrix} 0.35 \\ 0.16 \end{bmatrix} x_0$$

Hence, the MPC controller will apply $u_t = -0.35x_t$.

To solve the problem using DP instead, we apply the recursions derived in class starting from $P_2 = q = 1$. We have $A = 1.5$, $B = 1$, $Q_1 = 1$ and $Q_2 = 10$:

$$\begin{aligned} &P_2 = 1 \\ &P_1 = Q_1 + A^2 P - A^2 B^2 P_2^2 / (Q_2 + B^2 P_2) = 3.05 \end{aligned}$$

We do not need to compute P_0, since we can compute L_0 from P_1:

$$L_0 = ABP_1 / (Q_2 + B^2 P_1) = 0.35$$

Hence, the optimal control will be $u_t = -L_0 x_t = -0.35x_t$.

(b) We should let q be equal to the stationary optimal solution to the associated LQR problem. In terms of the standard notation in the course, we have

$$x_{t+1} = Ax_t + Bu_t$$

with $A = 1.5$ and $B = 1$; while

$$J = \sum_{k=0}^{\infty} x_k^T Q_1 x_k + u_k^T Q_2 u_k$$

with $Q_1 = 1$ and $Q_2 = 10$. The Riccati equation for the (infinite-horizon) LQ-optimal is

$$P = Q_1 + A^T PA - A^T PB(Q_2 + B^T PB)^{-1} B^T PA$$

Hence, the optimal P satisfies

$$P = 1 + 2.25P - \frac{2.25P^2}{10 + P}$$

By re-ordering and simplifying, we find that P must satisfy

$$2.25P^2 - (10 + P)(1.25P + 1) = P^2 - 13.5P - 10 = 0$$

which has positive solution $P \approx 14.2$. Thus, we should set $q = 14.2$.

(c) With $u_t = -0.88x_t$, the input constraint

$$-0.5 \leq u_t \leq 1$$

is satisfied at time $t = 0$ if

$$-1.14 \leq x_0 \leq 0.57 \tag{1}$$

The closed-loop dynamics is $x_{t+1} = 1.5x_t + u_t = 0.626x_t$. Hence, if x_0 satisfies (1), then x_1 will satisfy

$$-0.71 \leq x_1 \leq 0.36$$

We note that x_1 lies in a set contained within the set of admissible x_0's. Therefore, $u_1 = -0.88x_1$ will then also satisfy the constraints. We can repeat this argument for u_2, etc.

4. (a) The dynamic programming recursion is given by

$$V_t(x) = \min_u \{u_t^2 + V_{t+1}(x_t + u_t)\}$$

(b)

$$\begin{aligned}
V_3(x) &= x_3^2 \\
V_2(x) &= \min_u \{u_2^2 + (x_2+u_2)^2\} = \min_u \{2u_2^2 + 2x_2u_2 + x_2^2\} \\
&= \frac{x_2^2}{2} \text{ with } u_2^\star = -\frac{x_2}{2} \\
V_1(x) &= \min_u \{u_1^2 + \frac{1}{2}(x_1+u_1)^2\} = \min_u \{\frac{3}{2}u_1^2 + x_1u_1 + \frac{1}{2}x_1^2\} \\
&= \frac{x_1^2}{3} \text{ with } u_1^\star = -\frac{x_1}{3} \\
V_0(x) &= \min_u \{u_0^2 + \frac{1}{3}(x_0+u_0)^2\} = \min_u \{\frac{4}{3}u_0^2 + \frac{3}{2}x_1 0 u_0 + \frac{1}{3}x_0^2\} \\
&= \frac{x_0^2}{4} \text{ with } u_0^\star = -\frac{x_0}{4}.
\end{aligned} \tag{2}$$

Hence, $V_0(x_0) = \frac{1}{4}$. Computing the optimal sequence of inputs gives $u_0^\star = -\frac{1}{4} \Rightarrow x_1 = \frac{3}{2} \Rightarrow u_1^\star = -\frac{1}{4} \Rightarrow x_2 = \frac{1}{2} \Rightarrow u_2^\star = -\frac{1}{4} \Rightarrow x_3 = 0.$

(c) From (b), we conjecture that

$$V_t(x) = \frac{x^2}{n-t+1}$$

We then proceed to verify that V_t satisfies the dynamic programming recursion:

$$\begin{aligned}
V_t(x_t) &= \min_u \left\{ u_t^2 + \frac{(x_t+u_t)^2}{n-t} \right\} = \\
&= \min_u \left(\frac{1}{n-t}\left((n-t+1)u_t^2 + 2u_tx_t + x_t^2\right) \right)
\end{aligned}$$

For which the optimal control action is

$$u_t^\star = -\frac{n-t}{n-t+1}x_t$$

and

$$V_t(x_t) = \left(1 - \frac{n-t}{n-t+1}\right)x_t^2 = \frac{x_t^2}{n-t+1}$$

Thus, the proposed value function satisfies the dynamic programming recursion. Hence, the optimal control is

$$u_t^\star = -\frac{n-t}{n-t+1}x_t$$

5. (a) We first notice that $V_0(x_0)$ has the desired form with $S_0 = C^TC$. In fact, it is also straight forward to verify that S_t has the desired form for any $t \geq 0$ since

$$V_t(x_0) = \sum_{k=0}^{t} y_k^T y_k = \sum_{k=0}^{t} x_k^T C^T C x_k = \\ = x_0^T \left(\sum_{k=0}^{t} (A^k)^T C^T C A^k \right) x_0$$

Thus,

$$S_t = \sum_{k=0}^{t} (A^k)^T C^T C A^k$$

Therefore,

$$S_{t+1} = \sum_{k=0}^{t+1} (A^k)^T C^T C A^k = C^T C + \sum_{k=1}^{t+1} (A^k)^T C^T C A^k = \\ = C^T C + A^T \left(\sum_{k=0}^{t} (A^k)^T C^T C A^k \right) A = \\ = C^T C + A^T S_t A$$

(b) If the iteration converges, we have $S_t = S$ for all t. Hence,

$$S = A^T S A + C^T C$$

We recognize that this expression has the form of a Lyapunov equation $A^T P A - P = -Q$, for $P = S$ and $Q = C^T C$.

(c) When $a = 0$, the two system states are decoupled and do not affect each other. At the same time, we can only observe the first state in the output, so we should expect S to have the form

$$S = \begin{bmatrix} s_{11} & 0 \\ 0 & 0 \end{bmatrix}$$

(d) When $a = 0.115$, we parameterize the unknown S as

$$S = \begin{bmatrix} s_{11} & s_{12} \\ s_{12} & s_{22} \end{bmatrix}$$

and attempt to solve the matrix equation $S - A^T S A = C^T C$:

$$\begin{bmatrix} 0.36 s_{11} & 0.92(s_{12} - 0.1 s_{11}) \\ 0.92(s_{12} - 0.1 s_{11}) & 0.99 s_{22} - 0.115(0.115 s_{11} + 0.2 s_{12}) \end{bmatrix} = \begin{bmatrix} 0.36 & 0 \\ 0 & 0 \end{bmatrix}$$

From which we find

$$
\begin{aligned}
s_{11} &= 1 \\
s_{12} &= 0.1 \\
s_{22} &= 0.015525/0.99 \approx 0.0157
\end{aligned}
$$

Hence,

$$
V_t(x_0) = x_0^T \begin{bmatrix} 1 & 0.1 \\ 0.1 & 0.157 \end{bmatrix} x_0
$$

from which we can see that the second state still generates quite limited output energy, and hence should intuitively be harder to estimate than the first state (at least in the presence of disturbances and measurement noise).