SSY281 - Model predictive control
Assignment 1

Nicholas Granlund

February 2, 2023

Question 1: Discretization of a state-space model

(a) Find A, B, C
While assuming $\sin(x) \approx x$ we can achieve the approximate linear model with the output $y(t) = \theta_1$ as seen in equation (1).

$$\begin{aligned} \dot{x}(t) &= \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ b & 0 & 0 & a \\ 0 & 0 & 0 & 1 \\ q & 0 & 0 & p \end{bmatrix}}_{A_c} x(t) + \underbrace{\begin{bmatrix} 0 \\ c \\ 0 \\ r \end{bmatrix}}_{B_c} u(t) \\ y(t) &= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}}_{C_c} x(t) \end{aligned} \tag{1}$$

If we apply Lemma 2.1 from the lecture notes, we can convert the state space to a discretized system. Using the given numerical values for a, b, c, p, q, r and h we get the discretized system below in equation (2).

$$\begin{aligned} x(k+1) &= \underbrace{\begin{bmatrix} 1.4421 & 0.1143 & 0.0000 & -0.0045 \\ 9.4370 & 1.4421 & 0.0000 & -0.0908 \\ 0.0237 & 0.0008 & 1.0000 & 0.0833 \\ 0.4814 & 0.0237 & 0.0000 & 0.6845 \end{bmatrix}}_{A_d} x(k) + \underbrace{\begin{bmatrix} 0.0677 \\ 1.3715 \\ 0.2530 \\ 4.7660 \end{bmatrix}}_{B_d} u(k) \\ y(k) &= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}}_{C_d} x(k) \end{aligned} \tag{2}$$

(b) Find A_a, B_a, C_a
If we introduce the computational delay $\tau = 0.8h$ we will then get a new, augmented, system where we have to insert the previous input $u(k-1)$ into the state vector. The augmented state-space is thus given by equation (3).

$$\begin{aligned} \xi(k+1) &= \begin{bmatrix} A_d & B_1 \\ 0 & 0 \end{bmatrix} \xi(k) + \begin{bmatrix} B_2 \\ I \end{bmatrix} \\ y(k) &= \begin{bmatrix} C_d & 0 \end{bmatrix} \xi(k) \end{aligned} \tag{3}$$

We calculate the vectors B_1 and B_2 with Lemma 2.2 in the lecture notes. We thereafter obtain the numerical values for the matrices as seen in equation (4)

$$\begin{aligned} A_a &= \begin{bmatrix} A_d & B_1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1.4421 & 0.1143 & 0.0000 & -0.0045 & 0.0649 \\ 9.4370 & 1.4421 & 0.0000 & -0.0908 & 1.0958 \\ 0.0237 & 0.0008 & 1.0000 & 0.0833 & 0.2419 \\ 0.4814 & 0.0237 & 0.0000 & 0.6845 & 3.6657 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \\ B_a &= \begin{bmatrix} B_2 \\ I \end{bmatrix} = \begin{bmatrix} 0.0028 & 0.2757 & 0.0111 & 1.1002 & 1.0000 \end{bmatrix}^\top \\ C_a &= \begin{bmatrix} C_d & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{4}$$

After comparing the matrices A_d and A_a we can tell that the matrices has the same eigenvalues, but the augmented matrix has an additional pole in the origin. Nevertheless, both systems are unstable since they have a pole outside the unit disc. The poles for the different systems are given by expression (5).

$$\begin{aligned} eig(A_d) &= \{1.0000, 2.4781, 0.4008, 0.6899\} \\ eig(A_a) &= \{1.0000, 2.4781, 0.4008, 0.6899, 0.0000\} \end{aligned} \tag{5}$$

Through curiosuty, we notice that when the computational delay $\tau = 0$, the augmented system is identical to the discrete system. i.e $\tau \to 0$ then $B_1 \to 0$ and $B_2 \to B_d$.

Question 2: DP solution of the LQ problem

(a) Find the shortest N

The problem was solved using Matlab. A function `find_minimum_horizon_DP` was created which iteratively increased the length of the horizon N as long as the closed loop $(A+BK)$ was unstable. Given the parameters $Q = I_4$, $P_f = 10 \times I_4$ and $R = 1$, A stable feedback was obtained when the horizon reached $N = 33$. The output of the function was as as seen in expression (6) with the closed loop poles given by expression (7).

$$N_{DP} = 33, \qquad K_{DP} = \begin{bmatrix} -46.1565 & -5.2278 & 0.01554 & 0.4031 \end{bmatrix} \tag{6}$$

$$eig(A + BK_{DP}) = \{0.5584 + 0i, 0.9235 + 0i, 0.9523 + 0i, 0.9995 + 0i\} \tag{7}$$

(b) Find stationary solution P_∞

By using the `idare` function the stationary solution$P_{N=\infty}$ for the Riccati equation could be found. It was calculated to the numerical values given in expression (8);

$$P_{N=\infty} = \begin{bmatrix} 4.8730e+04 & 5.3230e+03 & -1.0378e+03 & -1.1542e+03 \\ 5.3230e+03 & 5.8747e+02 & -1.1437e+02 & -1.2712e+02 \\ -1.0378e+03 & -1.1437e+02 & 1.2499e+02 & 2.6626e+01 \\ -1.1542e+03 & -1.2712e+02 & 2.6626e+01 & 2.9814e+01 \end{bmatrix} \tag{8}$$

By trying to find the same stationary solution with DP and having $(P(k+1) - P(k)) \leq 0.1$ as the terminal criterion, another function `find_stationary_sol` was created which increments the control horizon N untill that the terminal criterion is fullfilled. It was calculated that the stationary solution was obtained when $N = 427$. The solution $P_{N=427}$, which is seen in expression (9), is evidently rather similar to the solution found by the `idare` function in expression (8).

$$P_{N=427} = \begin{bmatrix} 4.8726e+04 & 5.3225e+03 & -1.0373e+03 & -1.1541e+03 \\ 5.3225e+03 & 5.8741e+02 & -1.1431e+02 & -1.2711e+02 \\ -1.0373e+03 & -1.1431e+02 & 1.2493e+02 & 2.6613e+01 \\ -1.1541e+03 & -1.2711e+02 & 2.6613e+01 & 2.9811e+01 \end{bmatrix} \tag{9}$$

(c) Consider P_∞ as the new terminal cost gain P_f, Find the shortest N

By asserting that $P_f := P_\infty$ we can rerun the function `find_minimum_horizon_DP` to determine what control horizon N is needed to achieve a stable closed loop. By doing this we get a stable feedback when $N = 1$.

This can be explained by the fact that our final estimation error covariance P_f is set to be the same as the solution which satisfies the Riccati equation, P_∞. i.e we have found P such that $P(N-1) = P(N)$ is satisfied and thus no horizon beyond $N = 1$ is needed.

Question 3: Batch solution of the LQ problem

(a) Find the shortest N

This task was approached in a similar manor as the previous task. The function `find_minimum_horizon_batch` was created which iteratively increased the length of the horizon N as long as the closed loop $(A + BK)$ was unstable. In this function, the vector Ω and matrix Γ is calculated for a given N and are used to solve for K_b. Given the same parameters as the previous task, A stable feedback was obtained when the horizon $N = 33$. The output of the function was as as seen in expression (10) with the closed loop poles given by expression (11)

$$N_b = 33, \qquad K_b = \begin{bmatrix} -46.1565 & -5.2278 & 0.01554 & 0.4031 \end{bmatrix} \tag{10}$$

$$eig(A + BK_b) = \{0.5584 + 0i, 0.9235 + 0i, 0.9523 + 0i, 0.9995 + 0i\} \tag{11}$$

Interestingly the results (10-11) are the same as (6-7). Both the DP and batch approach result in same control horizon N, the same feedback gain K and thus the same eigenvalues of the closed loop system $(A + BK)$.

Question 4: Receding horizon control

(a) Design four RHC with different parameters

This problem was solved with DP. Batch is more computational expensive, but since the horizon is rather small, the difference may be negligible. The system was simulated with the help of a created function `simulate_system_DP` for the initial condition $x(0) = [\frac{\pi}{36}\ 0\ 0\ 0]^\top$. The system was plotted according to instructions and are seen in Figure 1.

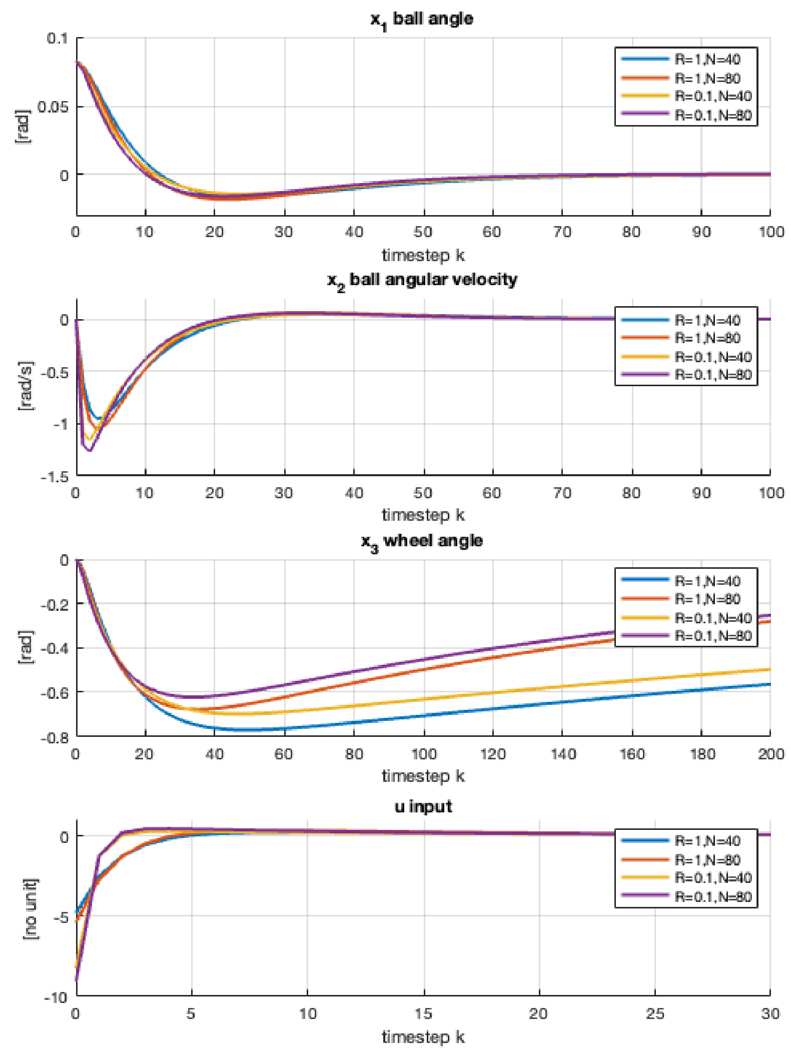

Figure 1: Unconstrained system states x_1, x_2, x_3 and system input u for the four different RH controllers.

The behaviour varies from the different controllers. Firstly, the controllers with smaller R has more actuator activity. This is expected since small R enforces a rather "cheap" control. This can be seen in the last plot for the inputs u, where it is visible that controller 3 and 4 (yellow and purple) reaches both the lowest and highest values. The first and second controller (blue and red) have more expensive control and thus lower actuator activity.

The decision of control horizon N result in different state trajectories. This is especially clear in the third plot for state x_3. For the controllers with the longer horizon $N = 80$, the wheel angle moves towards 0 faster than the controllers with a shorter horizon $N = 40$. This can also be seen if the state trajectory is plotted for $k > 200$ where x_3 converges to 0. Interestingly, if we plot the system for a horizon of $N < 33$, x_3 will not converge to 0 but instead diverge since such horizons are too short to yield a stable feedback.

Question 5: Constrained receding horizon control

When the system is constrained, the optimal control policy can no longer be calculated explicitly. Instead we have to utilize the function `quadprog` to solve for a constrained optimization to obtain the optimal control policy. With the constraints $| x_2 | \leq 1$ and $| u | \leq 8$, the optimal control yields the state trajectories in Figure 2.

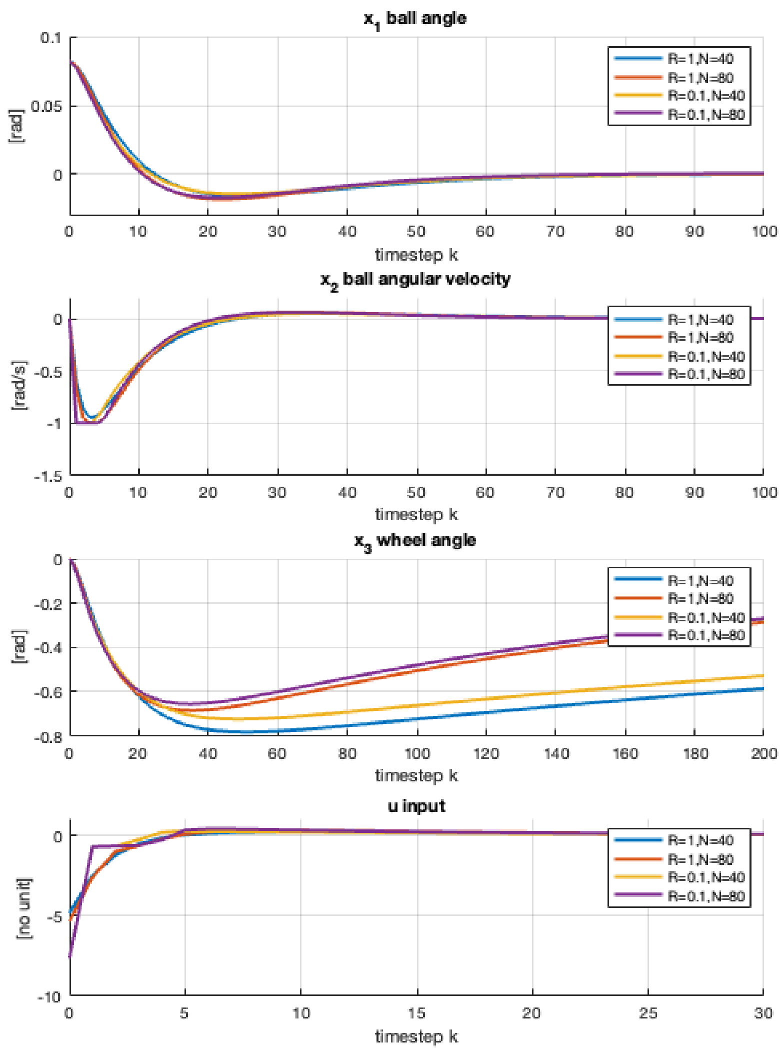

Figure 2: Constrained system states x_1, x_2, x_3 and system input u for the four different RH controllers. $| x_2 | \leq 1$ and $| u | \leq 8$

The impact of the constraint is clearly visible in the second plot for x_2. For the two controllers with the cheapest control (yellow and purple) the state quickly reaches its limit where it stays constant. In the unconstrained control, the optimal control enforces the state x_2 to go beyond -1. Since this state constrained, we can also observe that the actuator activity for those controllers are also limited as they stay somewhat constant for the time instances $k = 1, 2, 3$ where x_2 has reached its limit.

The first and third states, x_1 and x_3, are also impacted by the constraints, though it is not clearly visible. The limit in angular velocity of the ball, also limits how fast the wheel may turn, thus the slope for state x_3 is somewhat smaller in the constrained case compared to the unconstrained case. Same analogy goes for state x_1. This is even more empathised when $\mid x_2 \mid$ is more constrained.

/ Nicholas Granlund