

1.1.1)

Consider fully observable linear state-space model $x_{t+1} = Ax_t + Bu_t$ where (A, B) is a stabilizable pair. Provide an algorithm to compute K s.t. the static state feedback controller $u_t = Kx_t$ stabilizes the closed-loop system.

Solution: This problem can be solved using a discrete time infinite horizon LQR controller. The control is defined as $u^* = Kx = -(R + B^T S B)^{-1} B^T S A x$ so $K = -(R + B^T S B)^{-1} B^T S A$. Where S is computed by solving the Discrete Algebraic Riccati Equation (DARE) for its one positive semidefinite matrix solution.

$$S = Q + A^T S A - (A^T S B)(R + B^T S B)^{-1}(B^T S A)$$

Solving the DARE is a very common procedure for control software and it has many efficient implementations.

1.1.2)

Consider the partially observable linear state-space model

$$x_{t+1} = Ax_t + Bu_t, \quad y_t = Cx_t$$

Solution Extending the solution for part 1.1.1), adding in a discrete "infinite horizon" kalman filter allows for a discrete infinite horizon LQG controller to be developed to stabilize the system. Following the procedure for part 1.1.1), the LQR feedback gain K_r can be computed through solving the DARE. The Kalman gain matrix can be solved in an analogous way because the Kalman filter and LQR are dual problems of each other. By transposing the A matrix, swapping the B matrix with C^T , changing R to be the measurement noise matrix V_n , changing Q to be the dynamics noise matrix V_d , and changing S to P for clarity, solving DARE gives the matrix necessary for computing the Kalman gain $K_f = -(V_n + CPC^T)^{-1}CPA^T$.

$$P = V_d + APA^T - (APC^T)(V_n + CPC^T)^{-1}(CPA^T)$$

The state estimate gets updated as

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k - K_f(y_k - \hat{y}_k)$$

1.1.3)

Consider state space model from part 1.1.2) where a static output feedback policy $u = Ky_t$ must make the closed-loop system is stable. What do we know about such problems. Solution Reference 6 says when the number of inputs and outputs are small, there exists constructive static output feedback pole placement (SOFPP) methods. Also, determining if SOFPP problem can be solved for a given system and set of poles is NP-hard. Reference 7 talks about the following approaches to solving this problem

- Iterative LMI heuristic approaches
- LMI with rank constraints
- Decoupled Lyapunov matrices