

Linear Dynamical Systems - Summary

Lecturer: Dr. Oron Sabag

Name: Hadar Tal

Contents

1	The Shortest Path Problem	2
2	Markov Decision Processes (MDPs)	3
3	Linear Systems	4
4	Linear Systems with Control	5
5	The Linear Quadratic Regulator (LQR)	6
6	Kalman Filter	7

1 The Shortest Path Problem

Definition 1.1 (The Shortest Path Problem)

Given a directed graph $G = (V, E)$, each edge in the graph has a cost a_{ij}^t , where i is the outgoing node, j is the node to which the edge is connected, and $t \in \{0, \dots, N+1\}$ refers to the time. We adopt the convention that no edge implies an infinite cost $a_{ij}^t = \infty$.

The objective is to minimize the cumulative cost on a path from the source node $S_0 = S$ to the terminal node $S_{N+1} = T$. Formally, we aim to solve the optimization

$$J^* = \min_{\{n_i \in \mathcal{S}_i\}_{i=0}^{N+1}} \sum_{t=0}^N a_{n_t, n_{t+1}}^t. \quad (1.1)$$

Definition 1.2 (Cost-to-Go Function)

We define $J_k(i)$ as the cost-to-go function corresponding to the minimal cost from time k until the end when starting at node i . Formally, for $k = 0, \dots, N$, define

$$J_k(i) = \min_{\{n_j \in \mathcal{S}_j | j=k+1, \dots, N, n_k=i\}} \sum_{j=k}^N a_{n_j, n_{j+1}}^j, \quad \forall i \in \mathcal{S}_k. \quad (1.2)$$

Algorithm 1: Dynamic Programming Solution for the Shortest Path Problem (Cost-to-Go)

Input: Cost matrix a_{ij}^t and nodes $\mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_{N+1}$

Output: Cost-to-go functions $J_k(i)$

Initialize $J_N(i) = a_{iT}^N$;

for $k = N-1, \dots, 0$ **do**

for $i \in \mathcal{S}_k$ **do**

$J_k(i) = \min_{j \in \mathcal{S}_{k+1}} [a_{ij}^k + J_{k+1}(j)]$

end

end

Definition 1.3 (Cost-to-Arrive Function)

We define $J_{N-k}(j)$ as the cost-to-arrive function corresponding to the minimal cost from time 1 until time k when arriving at node j . Formally, for $k = 0, \dots, N$, define

$$J_{N-k}(j) = \min_{\{n_i \in \mathcal{S}_i | i=1, \dots, k-1, n_k=j\}} \sum_{i=1}^k a_{n_{i-1}, n_i}^i, \quad \forall j \in \mathcal{S}_k. \quad (1.3)$$

Algorithm 2: Forward Algorithm for the Shortest Path Problem (Cost-to-Arrive)

Input: Cost matrix a_{ij}^t and nodes $\mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_{N+1}$

Output: Cost-to-arrive functions $J_{N-k}(j)$

Initialize $J_N(j) = a_{s_j}^0, \forall j \in \mathcal{S}_1$;

for $k = 1, \dots, N$ **do**

for $j \in \mathcal{S}_{N-k+1}$ **do**

$J_k(j) = \min_{i \in \mathcal{S}_{N-k}} [a_{ij}^{N-k} + J_{k+1}(i)]$

end

end

2 Markov Decision Processes (MDPs)

Definition 2.1 (MDP)

An MDP is defined by the following elements:

1. The state at time k is x_k and takes values in the set \mathcal{S}_k .
2. The action at time k is u_k and takes values from \mathcal{U}_k .
3. The disturbance at time k is w_k and takes values from \mathcal{W}_k .
4. A dynamical system is given by the function

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, \dots, N-1. \quad (2.1)$$

5. The probabilistic law of the disturbance random variable w_k is characterized by $P_{W_k}(\cdot|x_k, u_k)$ conditioned on the state x_k and the action u_k .
6. A cost function $g_k : \mathcal{S}_k \times \mathcal{U}_k \rightarrow \mathbb{R}$.

The cost over a horizon N is

$$S_\pi(x_0) = \mathbb{E}[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k)], \quad (2.2)$$

where $g_N(\cdot)$ is the terminal cost.

Definition 2.2 (History-dependent policy)

A history-dependent policy is defined by a sequence of functions:

$$\mu_k : \mathcal{S}_1 \times \dots \times \mathcal{S}_k \times \mathcal{U}_1 \times \dots \times \mathcal{U}_{k-1} \rightarrow \mathcal{U}_k, \quad (2.3)$$

such that $u_k = \mu_k(x_1, x_2, \dots, x_k, u_1, \dots, u_{k-1})$.

Definition 2.3 (Markovian policy)

A Markovian policy is defined by a sequence of functions:

$$\mu_k : \mathcal{S}_k \rightarrow \mathcal{U}_k, \quad (2.4)$$

such that $u_k = \mu_k(x_k)$.

Remark 2.1 (The Markov property)

We defined the dynamical system using a deterministic function $f_k(\cdot)$.

Equivalently, we could describe the evolution with the conditional probability

$$P_k(x_{k+1}|x_k, u_k) = P_s^u(s') \quad (2.5)$$

In particular, we assume that the new state conditioned on the current state and action is not affected by the past. Formally, we assume the Markov chain induced from

$$P(x_{k+1}|x_1, \dots, x_k, u_1, \dots, u_k) = P_k(x_{k+1}|x_k, u_k). \quad (2.6)$$

The MDP described above is called fully observable since the actions depend directly on the state. We will later encounter partially observable MDP where only a noisy version of the state is available to the controller.

3 Linear Systems

Definition 3.1 (Linear System)

A linear system is given by

$$x_{t+1} = Ax_t, \quad t = 0, 1, \dots \quad (3.1)$$

with some initial state x_0 .

- $x_t \in \mathbb{R}^n$ is the state vector.
- $A \in \mathbb{R}^{n \times n}$ is the state-transition matrix.

Lemma 3.1 (State and Decoupling in Linear Systems)

Given a diagonalizable matrix $A = TDT^{-1}$, the state at time t in a linear system is

$$x_t = TD^tT^{-1}x_0. \quad (3.2)$$

By defining a new state $z_t = T^{-1}x_t$, we have

$$z_t = D^tz_0, \quad (3.3)$$

indicating that the states are decoupled, with each entry of z_t depending only on the corresponding entry of z_0 .

Remark 3.1

Since the eigenvalues of real matrices may be complex, we have

$$\lambda = a + ib = re^{i\theta} \rightarrow \lambda^t = r^te^{it\theta} (e^{i\theta} = \cos \theta + i \sin \theta).$$

As we increase t , the magnitude of $e^{it\theta}$ is clearly unchanged. However, the length of r determines whether it converges to zero, oscillates, or blows up.

Definition 3.2 (Stable System)

A system A is stable if all of its eigenvalues have magnitude smaller than 1, i.e., $r < 1$.

4 Linear Systems with Control

Definition 4.1 (Linear System with Control)

A linear system with control is given by

$$x_{t+1} = Ax_t + Bu_t, \quad t \geq 0, \quad x_0 \in \mathbb{R}^n, \quad (4.1)$$

where we added:

- $u_t \in \mathbb{R}^m$ is the control signal (action).
- $B \in \mathbb{R}^{n \times m}$ is the control matrix.

Definition 4.2 (State-feedback controller)

A controller (policy) is defined by a sequence of mappings $\mu_t : \mathbb{R}^n \rightarrow \mathbb{R}^m$ for $t = 0, 1, \dots, N$ such that $u_t = \mu_t(x_t)$.

Definition 4.3 (State-Feedback, Time-Invariant, Linear Controller)

A state-feedback, time-invariant, linear controller is any mapping of the form

$$u_t = -Kx_t.$$

Definition 4.4 (Closed-Loop Matrix)

The matrix $A_K = A - BK$ is called the closed-loop matrix of the system $A, I.H.T$ -

$$x_{t+1} = A_K x_t = (A - BK)x_t = Ax_t + B(-Kx_t) = Ax_t + Bu_t$$

Definition 4.5 (Controllability)

The pair (A, B) is controllable if the system can reach any $\xi \in \mathbb{R}^n$ from any initial state $x_0 \in \mathbb{R}^n$ at some finite time.

Lemma 4.1 (Controllability Matrix)

A pair (A, B) is controllable if and only if the controllability matrix

$$C \triangleq [B \quad AB \quad \dots \quad A^{n-1}B]$$

has $\text{rank}(C) = n$.

Lemma 4.2 (Poles Placement in Controllable System)

Controllability implies that we can choose the eigenvalues (poles) of $A - BK$ arbitrarily.

5 The Linear Quadratic Regulator (LQR)

Definition 5.1 (The LQR problem)

For the linear model in (20), find a controller that minimizes

$$J_N(u^N) = \sum_{i=0}^N [x_i^T Q x_i + u_i^T R u_i] + x_{N+1}^T Q_f x_{N+1},$$

where $u^N \triangleq u_0, u_1, \dots, u_{N-1}$, and

1. $Q, Q_f \succeq 0$ are state weights
2. $R \succ 0$ is the input/action/control weight. (Reminder: $R \succ 0 \iff x^T R x > 0 \forall x$).

6 Kalman Filter

Definition 6.1 (Kalman Filter State Space)

$$\begin{aligned} x_{i+1} &= Fx_i + Gw_i \\ y_i &= Hx_i + v_i \end{aligned} \quad (6.1)$$

where:

- x_{i+1} is the state vector at time $i + 1$,
- x_i is the state vector at time i ,
- y_i is the measurement vector at time i ,
- w_i is the process noise (zero mean, uncorrelated),
- v_i is the measurement noise (zero mean, uncorrelated).

Definition 6.2 (Kalman Filter Covariance Matrix)

Formally, the following covariance matrix describes the model:

$$\mathbb{E} \left[\begin{pmatrix} w_i \\ v_i \\ x_0 \end{pmatrix} \begin{pmatrix} w_j^* & v_j^* & x_0^* & 1 \end{pmatrix} \right] = \begin{pmatrix} \begin{pmatrix} Q & S \\ S^* & R \end{pmatrix} \delta_{ij} & 0 & 0 \\ 0 & \Pi_0 & 0 \end{pmatrix}, \quad (6.2)$$

where $\begin{pmatrix} Q & S \\ S^* & R \end{pmatrix}$ and Π_0 are positive semidefinite matrices and δ_{ij} equals 1 if $i = j$ and is zero otherwise. Note that w_i is uncorrelated as a process over time but its coordinates at a fixed time can be correlated via Q .

Markings:

- P_i - The error covariance matrix at time i

$$P_i \triangleq (x_i - \hat{x}_i)(x_i - \hat{x}_i)^T \quad (6.3)$$

- $R_{e,i}$ - The covariance of the innovation (or residual) at time i

$$R_{e,i} \triangleq HP_iH^* + R \quad (6.4)$$

- $K_{p,i}$ - The optimal Kalman gain at time i

$$K_{p,i} \triangleq (FP_iH^* + GS)R_{e,i}^{-1} \quad (6.5)$$

Kalman Filter Optimality

We suggest the following predictor:

$$\hat{x}_{i+1|i} = F\hat{x}_{i|i-1} + K_{p,i}(y_i - H\hat{x}_{i|i-1}) \quad (6.6)$$

Lemma 6.1

$$\tilde{x}_{i+1} = (F - K_{p,i}H)\tilde{x}_i + (G - K_{p,i}) \begin{pmatrix} w_i \\ v_i \end{pmatrix}. \quad (6.7)$$

Proof.

$$\begin{aligned} \tilde{x}_{i+1} &= x_{i+1} - \hat{x}_{i+1|i} \\ &= (Fx_i + Gw_i) - (F\hat{x}_i + K_{p,i}(y_i - H\hat{x}_i)) \\ &= Fx_i + Gw_i - F\hat{x}_i - K_{p,i}(Hx_i + v_i - H\hat{x}_i) \\ &= Fx_i + Gw_i - F\hat{x}_i - K_{p,i}Hx_i - K_{p,i}v_i + K_{p,i}H\hat{x}_i \\ &= Fx_i - F\hat{x}_i - K_{p,i}Hx_i + K_{p,i}H\hat{x}_i + Gw_i - K_{p,i}v_i \\ &= (F - K_{p,i}H)(x_i - \hat{x}_i) + Gw_i - K_{p,i}v_i \\ &= (F - K_{p,i}H)\tilde{x}_i + Gw_i - K_{p,i}v_i. \end{aligned}$$

□

Lemma 6.2

For $j < i$, the recursion can be evolved as

$$\begin{aligned}\tilde{x}_i &= (F - K_{p,i-1}H)\tilde{x}_{i-1} + (G - K_{p,i-1}) \begin{pmatrix} w_{i-1} \\ v_{i-1} \end{pmatrix} \\ &= \dots \\ &= \phi_p(i, j)\tilde{x}_j + \xi_i(j),\end{aligned}$$

where

$$\begin{aligned}\phi_p(i, j) &= \prod_{k=j}^{i-1} (F - K_{p,k}H), \\ \xi_i(j) &= \sum_{k=j}^{i-1} \phi_p(i, k+1)(Gw_k - K_{p,k}v_k).\end{aligned}$$