

Linear Dynamical Systems - Summary

Lecturer: Dr. Oron Sabag*Name:* Hadar Tal**Contents**

| | | |
|----------|---|-----------|
| 1 | Mathematical Tools | 2 |
| 1.1 | Completing The Square | 2 |
| 1.2 | The Z-transform | 2 |
| 1.2.1 | Properties of the Z-transform | 2 |
| 1.2.2 | Differences Between Z-transform and Laplace Transform | 3 |
| 2 | The Shortest Path Problem | 4 |
| 3 | Markov Decision Processes (MDPs) | 5 |
| 4 | Linear Systems | 6 |
| 5 | Linear Systems with Control | 7 |
| 6 | The Linear Quadratic Regulator (LQR) | 8 |
| 7 | State-space models as mappings | 10 |
| 7.1 | Controllable Canonical Form | 10 |
| 8 | The Asymptotic Observer | 12 |
| 9 | Kalman Filter | 13 |

1 Mathematical Tools

1.1 Completing The Square

Lemma 1.1 (Completing the Square (scalars))

To complete the square for a quadratic equation of the form

$$ax^2 + bx + c = 0,$$

we can rewrite it as

$$a(x + d)^2 + e = 0,$$

where

$$d = \frac{b}{2a} \quad \text{and} \quad e = c - \frac{b^2}{4a}.$$

Proof.

$$\begin{aligned} ax^2 + bx + c = 0 & \rightarrow x^2 + \frac{b}{a}x + \frac{c}{a} = 0 & \rightarrow x^2 + \frac{b}{a}x = -\frac{c}{a} & \rightarrow \\ x^2 + \frac{b}{a}x + \left(\frac{b}{2a}\right)^2 = -\frac{c}{a} + \left(\frac{b}{2a}\right)^2 & \rightarrow \left(x + \frac{b}{2a}\right)^2 = \frac{b^2}{4a^2} - \frac{c}{a} & \rightarrow \left(x + \frac{b}{2a}\right)^2 = \frac{b^2 - 4ac}{4a^2} & \rightarrow \\ a\left(x + \frac{b}{2a}\right)^2 = a\left(\frac{b^2 - 4ac}{4a^2}\right) & \rightarrow a\left(x + \frac{b}{2a}\right)^2 = \frac{b^2 - 4ac}{4a} & \rightarrow a\left(x + \frac{b}{2a}\right)^2 + c - \frac{b^2}{4a} = 0 \end{aligned}$$

□

Lemma 1.2 (Completing the Square for Quadratic Forms)

Given a quadratic form $x^T A x + b^T x + c$, where A is a symmetric positive definite matrix, b is a vector, and c is a scalar, the expression can be completed to a perfect square as follows:

$$x^T A x + b^T x + c = (x + A^{-1}b/2)^T A (x + A^{-1}b/2) + c - \frac{1}{4}b^T A^{-1}b$$

1.2 The Z-transform

The Z-transform converts a discrete-time signal, which is a sequence of real or complex numbers, into a complex frequency domain representation.

The Z-transform of a discrete-time signal $x[k]$ is defined as:

$$X(z) = \mathcal{Z}\{x[k]\} = \sum_{k=-\infty}^{\infty} x[k]z^{-k},$$

where z is a complex variable. The Z-transform is particularly useful for analyzing linear time-invariant (LTI) systems.

1.2.1 Properties of the Z-transform

1. Linearity:

$$\mathcal{Z}\{ax[k] + by[k]\} = aX(z) + bY(z).$$

2. Time Shifting:

$$\mathcal{Z}\{x[k - n]\} = z^{-n}X(z).$$

3. Convolution:

$$\mathcal{Z}\{x[k] * y[k]\} = X(z)Y(z).$$

4. Initial Value Theorem:

$$x[0] = \lim_{z \rightarrow \infty} X(z).$$

5. Final Value Theorem:

$$\lim_{k \rightarrow \infty} x[k] = \lim_{z \rightarrow 1} (1 - z^{-1})X(z),$$

provided the limits exist.

1.2.2 Differences Between Z-transform and Laplace Transform

1. **Domain:** The Z-transform is used for discrete-time signals, while the Laplace transform is used for continuous-time signals.
2. **Definition:** The Z-transform is defined as a summation:

$$X(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k},$$

while the Laplace transform is defined as an integral:

$$X(s) = \int_0^{\infty} x(t)e^{-st} dt.$$

3. **Complex Variable:** The Z-transform uses the complex variable z , typically represented as $z = e^{sT}$ where T is the sampling period. The Laplace transform uses the complex variable s .
4. **Application:** The Z-transform is applied to discrete-time systems and signals, making it useful for digital signal processing and discrete control systems. The Laplace transform is applied to continuous-time systems and signals, making it useful for analog signal processing and continuous control systems.

2 The Shortest Path Problem

Definition 2.1 (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

$$J^* = \min_{\{n_i \in \mathcal{S}_i\}_{i=0}^{N+1}} \sum_{t=0}^N a_{n_t, n_{t+1}}^t. \quad (2.1)$$

Definition 2.2 (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

$$J_k(i) = \min_{\{n_j \in \mathcal{S}_j | j=k+1, \dots, N, n_k=i\}} \sum_{j=k}^N a_{n_j, n_{j+1}}^j, \quad \forall i \in \mathcal{S}_k. \quad (2.2)$$

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 2.3 (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

$$J_{N-k}(j) = \min_{\{n_i \in \mathcal{S}_i | i=1, \dots, k-1, n_k=j\}} \sum_{i=1}^k a_{n_{i-1}, n_i}^i, \quad \forall j \in \mathcal{S}_k. \quad (2.3)$$

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

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

3 Markov Decision Processes (MDPs)

Definition 3.1 (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

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, \dots, N-1. \quad (3.1)$$

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

$$S_\pi(x_0) = \mathbb{E}[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k)], \quad (3.2)$$

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

Definition 3.2 (History-dependent policy)

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

$$\mu_k : \mathcal{S}_1 \times \dots \times \mathcal{S}_k \times \mathcal{U}_1 \times \dots \times \mathcal{U}_{k-1} \rightarrow \mathcal{U}_k, \quad (3.3)$$

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

Definition 3.3 (Markovian policy)

A Markovian policy is defined by a sequence of functions:

$$\mu_k : \mathcal{S}_k \rightarrow \mathcal{U}_k, \quad (3.4)$$

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

Remark 3.1 (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

$$P_k(x_{k+1}|x_k, u_k) = P_s^u(s') \quad (3.5)$$

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

$$P(x_{k+1}|x_1, \dots, x_k, u_1, \dots, u_k) = P_k(x_{k+1}|x_k, u_k). \quad (3.6)$$

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.

4 Linear Systems

Definition 4.1 (Linear System)

A linear system is given by

$$x_{t+1} = Ax_t, \quad t = 0, 1, \dots \quad (4.1)$$

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 4.1 (State and Decoupling in Linear Systems)

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

$$x_t = TD^tT^{-1}x_0. \quad (4.2)$$

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

$$z_t = D^tz_0, \quad (4.3)$$

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

Remark 4.1

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 4.2 (Stable System)

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

5 Linear Systems with Control

Definition 5.1 (Linear System with Control)

A linear system with control is given by

$$x_{t+1} = Ax_t + Bu_t, \quad t \geq 0, \quad x_0 \in \mathbb{R}^n, \quad (5.1)$$

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 5.2 (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 5.3 (State-Feedback, Time-Invariant, Linear Controller)

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

$$u_t = -Kx_t.$$

Definition 5.4 (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 5.5 (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 5.1 (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 5.2 (Poles Placement in Controllable System)

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

6 The Linear Quadratic Regulator (LQR)

Definition 6.1 (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.

Lemma 6.1 (LQR matrix formulation)

$$\begin{bmatrix} x_0 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \end{bmatrix} + \begin{bmatrix} I \\ A \\ \vdots \\ A^N \end{bmatrix} x_0$$

The matrices above can be written in short as $x = Gu + Hx_0$.

Lemma 6.2 (LQR Closed-Form Solution)

The optimal control input \mathbf{u} that minimizes the cost function $\mathbf{u}^T \mathbf{u} + \mathbf{x}^T \mathbf{x}$ can be achieved with

$$\mathbf{u} = -(I + G^T G)^{-1} G^T H x_0.$$

Proof.

$$\begin{aligned} \min_{\mathbf{u}} \mathbf{x}^T \mathbf{x} + \mathbf{u}^T \mathbf{u} &= \min_{\mathbf{u}} (G\mathbf{u} + Hx_0)^T (G\mathbf{u} + Hx_0) + \mathbf{u}^T \mathbf{u} \\ &= \min_{\mathbf{u}} [\mathbf{u}^T G^T G \mathbf{u} + 2x_0^T H^T G \mathbf{u} + x_0^T H^T H x_0 + \mathbf{u}^T \mathbf{u}] \\ &= \min_{\mathbf{u}} [\mathbf{u}^T (I + G^T G) \mathbf{u} + 2x_0^T H^T G \mathbf{u} + x_0^T H^T H x_0] \\ &= \min_{\mathbf{u}} [(\mathbf{u} + (I + G^T G)^{-1} G^T H x_0)^T (I + G^T G) (\mathbf{u} + (I + G^T G)^{-1} G^T H x_0) \\ &\quad - x_0^T H^T G (I + G^T G)^{-1} G^T H x_0 + x_0^T H^T H x_0] \\ &= \min_{\mathbf{u}} [(\mathbf{u} + (I + G^T G)^{-1} G^T H x_0)^T (I + G^T G) (\mathbf{u} + (I + G^T G)^{-1} G^T H x_0) \\ &\quad + x_0^T H^T (I - G(I + G^T G)^{-1} G^T) H x_0] \\ &= x_0^T H^T (I - G(I + G^T G)^{-1} G^T) H x_0 \\ &= x_0^T H^T (I + G G^T)^{-1} H x_0, \end{aligned}$$

where the optimal control input is

$$\mathbf{u} = -(I + G^T G)^{-1} G^T H x_0.$$

□

Remark 6.1

This solution is the optimal solution. However, it is not efficient since we should compute the inverse of $I + G G^T$ that grows linearly with N , i.e., $O(N^3)$ computations.

Definition 6.2 (Cost-to-Go Function)

The cost-to-go function (Value-function) is defined as

$$V_t(z) = \min_{u_t, u_{t+1}, \dots, u_{N-1}} \left[\sum_{i=t}^{N-1} (x_i^T Q x_i + u_i^T R u_i) + x_N^T Q_f x_N \right]$$

for $x_t = z$.

Theorem 6.1 (Properties of the Value Function)

The value function satisfies the following properties.

1. $V_t(z)$ is a quadratic function (of the variable z).

That is, we can write $V_t(z) = z^T P_t z$ with some $P_t \succeq 0$.

2. The optimal controller u_t is given by

$$u_t = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x_t.$$

3. The sequence P_t can be computed recursively with

$$P_t = \begin{cases} Q_f & t = N \\ Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A & t < N. \end{cases}$$

Note that we can compute P_t and K_t offline prior to the control.
(Tools used in the proof - completion of the square, induction.)

Definition 6.3 (Stabilizability)

The pair (A, B) is stabilizable if

$$\exists K \in \mathbb{R}^{m \times n} : \rho(A - BK) < 1,$$

where $\rho(A - BK)$ denotes the spectral radius of $A - BK$, i.e., the largest absolute value of its eigenvalues.

An equivalent characterization of stabilizability is

$$\nexists(x, \lambda) \text{ s.t. } xA = \lambda x \wedge |\lambda| \geq 1 \wedge xB = 0.$$

In other words, we can control the unstable modes.

If we want $J^*(x_0) < \infty$ for all x_0 , a necessary and sufficient condition is that of stabilizability.

7 State-space models as mappings

Definition 7.1 (Linear Time-Invariant (LTI) System)

A discrete-time linear time-invariant (LTI) system is represented by the difference equation:

$$y_k + a_1 y_{k-1} + a_2 y_{k-2} + \cdots + a_n y_{k-n} = b_0 u_k + b_1 u_{k-1} + b_2 u_{k-2} + \cdots + b_{n-1} u_{k-n+1}.$$

The inputs are the u_i while the outputs are the y_i .

Example 7.1 (Moving average)

This system outputs the average of the current and past input values.

If $a_1 = a_2 = \cdots = a_n = 0$, the equation simplifies to:

$$y_k = b_0 u_k + b_1 u_{k-1} + b_2 u_{k-2} + \cdots + b_{n-1} u_{k-n+1}.$$

This form indicates that the output y_k is a weighted sum of the current and previous inputs, hence the term "moving average."

Example 7.2 (Auto-regressive)

This system models the output as a function of its previous values.

If $b_1 = b_2 = \cdots = b_n = 0$, the equation simplifies to:

$$y_k + a_1 y_{k-1} + a_2 y_{k-2} + \cdots + a_n y_{k-n} = b_0 u_k.$$

This form indicates that the output y_k depends on its past values and the current input, making it "auto-regressive."

Example 7.3 (ARMA (Auto-regressive Moving Average))

This system combines both auto-regressive and moving average models. Sometimes used as ARMA(i, j) to include the order, it captures dependencies on both past outputs and past inputs:

$$y_k + a_1 y_{k-1} + a_2 y_{k-2} + \cdots + a_n y_{k-n} = b_0 u_k + b_1 u_{k-1} + b_2 u_{k-2} + \cdots + b_{n-1} u_{k-n+1}.$$

Here, the output y_k is influenced by both its previous values (auto-regressive part) and previous inputs (moving average part).

Definition 7.2 (Transfer Function $H(z)$)

The transfer function $H(z)$ of a discrete-time linear time-invariant (LTI) system represents the relationship between the Z-transform of the output $Y(z)$ and the Z-transform of the input $U(z)$. It provides a way to analyze the system's behavior in the frequency domain and is defined as:

$$H(z) = \frac{Y(z)}{U(z)}.$$

For an LTI system, this can be expressed as:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n-1} z^{-n+1}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}.$$

7.1 Controllable Canonical Form

Definition 7.3 (Controllable Canonical Form)

For the discrete-time transfer function $H(z)$, the state-space representation in controllable canonical form is given by the equations:

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t), \\ y(t) &= C_c x(t) + D_c u(t), \end{aligned}$$

where the state vector $x(t)$ and input $u(t)$ are defined as:

$$x(t) = \begin{bmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_{n-1}(t) \\ z_n(t) \end{bmatrix}, \quad u(t) = z_n(t),$$

and the matrices are defined as follows:

$$A_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

$$C_c = [b_{n-1} \quad b_{n-2} \quad \cdots \quad b_1 \quad b_0], \quad D_c = 0.$$

Add the proof of the controllable canonical form + observable canonical form + Jordan canonical form.

8 The Asymptotic Observer

Definition 8.1 (Linear System State Estimation)

The setting is given by a linear system:

$$\begin{aligned} x_{i+1} &= Ax_i + Bu_i, \quad x_0 \\ y_i &= Cx_i \end{aligned}$$

where:

1. x_i is the state sequence.
2. u_i is the control sequence (which is available to us).
3. y_i is the measurement process that we observe.

The objective is to estimate the states.

Claim 8.1

The optimal closed-loop estimate for a linear system is of the form

$$\hat{x}_{i+1} = A\hat{x}_i + Bu_i + K(y_i - C\hat{x}_i), \quad \hat{x}_0.$$

Claim 8.2 (Dynamics of the Estimation Error)

The estimation error $\tilde{x}_i \triangleq x_i - \hat{x}_i$ evolves according to

$$\tilde{x}_i = (A - KC)^i \tilde{x}_0.$$

Proof. We have

$$\tilde{x}_{i+1} = x_{i+1} - \hat{x}_{i+1} = Ax_i + Bu_i - A\hat{x}_i - Bu_i - K(y_i - C\hat{x}_i) = A\tilde{x}_i - KC\tilde{x}_i = (A - KC)\tilde{x}_i.$$

By induction, we conclude the claim. □

9 Kalman Filter

Definition 9.1 (Kalman Filter State Space)

$$\begin{aligned} x_{i+1} &= Fx_i + Gw_i \\ y_i &= Hx_i + v_i \end{aligned} \quad (9.1)$$

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 9.2 (Kalman Filter Covariance Matrix)

Formally, the following covariance matrix describes the model:

$$\mathbb{E} \left[\begin{pmatrix} w_i \\ v_i \\ x_0 \end{pmatrix} \begin{pmatrix} w_j^* & v_j^* & x_0^* & 1 \end{pmatrix} \right] = \begin{pmatrix} \begin{pmatrix} Q & S \\ S^* & R \end{pmatrix} \delta_{ij} & 0 & 0 \\ 0 & \Pi_0 & 0 \end{pmatrix}, \quad (9.2)$$

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

$$P_i \triangleq (x_i - \hat{x}_i)(x_i - \hat{x}_i)^T \quad (9.3)$$

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

$$R_{e,i} \triangleq HP_iH^* + R \quad (9.4)$$

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

$$K_{p,i} \triangleq (FP_iH^* + GS)R_{e,i}^{-1} \quad (9.5)$$

Kalman Filter Optimality

We suggest the following predictor:

$$\hat{x}_{i+1|i} = F\hat{x}_{i|i-1} + K_{p,i}(y_i - H\hat{x}_{i|i-1}) \quad (9.6)$$

Lemma 9.1

$$\tilde{x}_{i+1} = (F - K_{p,i}H)\tilde{x}_i + (G - K_{p,i}) \begin{pmatrix} w_i \\ v_i \end{pmatrix}. \quad (9.7)$$

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 9.2

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

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