titlepage.tex, v1162 2010-01-12 05:44:44Z (murray)

Networked Sensing, Estimation and Control Systems

Vijay Gupta
University of Notre Dame

Richard M. Murray
California Institute of Technology

Ling Shi
Hong Kong University of Science and Technology

Bruno Sinopoli
Carnegie Mellon University

DRAFT v1.1b, 11 January 2010

Chapter 2
State Estimation and Sensor Fusion

In this chapter, we provide an overview of Kalman filter and linear quadratic Gaussian (LQG) control. We will first provide a quick summary of basic theories of probability and stochastic process, which will be used to derive the Kalman filter equations. Some properties of Kalman filter and its steady-state error covariance matrix will also be provided. After that, we will introduce LQG control and derive the optimal control law using a dynamic programming approach.

The material of this chapter will be the foundation of most subsequent chapters including Chapters 4, 6 and 8. In particular, Chapter 6 considers the effect of data packet drops on Kalman filtering and LQG control and Chapter 8 considers distributed Kalman filtering.

Chapter 11 investigates how to improve state estimation using a Kalman filter in a networked environment, and Chapter 12 considers Kalman filtering using a sensor network subject to limited sensor communication bandwidth and energy. Supplement

2.1 Review of Probability and Random Process

We assume the readers have some exposure to the theory of probability and random process. The material presented in this section only serves as a quick review of some basic concepts and tools from probability and random process that will be helpful to understand and derive some important results in subsequent sections and chapters. Good introductory books on probability and random process are [GS01] and [LG93].

Random Variables

Consider an experiment with many (possibly infinite) outcomes. All these outcomes form the *sample space* Ω. A subset $A \subset \Omega$ is called an *event*. Two events A_1, A_2 are called *mutually disjoint* if $A_1 \cap A_2 = \emptyset$. The *complement* of an event A is defined as $\bar{A} = \Omega \setminus A$. A *probability measure* $P(\cdot)$ is a mapping from Ω into the interval $[0, 1]$ such that the following axioms of probability are satisfied:

1. $P(A) \geq 0$ for all $A \subset \Omega$.
2. $P(\Omega) = 1$.
3. If $\{A_i, i = 1, 2, \ldots\}$ is a collection of disjoint members of $\mathcal{F}$, i.e., $A_i \cap A_j = \emptyset$ for all i, j, then $P(\cup A_i) = \sum_i P(A_i)$.

From the axioms of probability, it follows that

$$P(A) \leq 1, \quad P(\emptyset) = 0, \quad P(\bar{A}) = 1 - P(A), \quad P(\cup A_i) \leq \sum_i P(A_i).$$

The joint probability of two events A and B is $P(A \cap B)$ which is often written as $P(AB)$ for simplicity. The conditional probability of A given B i.e., the probability that A occurs if B occurs in an experiment is

$$P(A|B) = \frac{P(AB)}{P(B)}, \quad \text{assuming } P(B) \neq 0.$$

A and B are *mutually independent* if

$$P(AB) = P(A)P(B).$$

If $P(B) \neq 0$, the conditional probability $P(A|B)$ can be calculated from *Bayes' Rule* as

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

If $A_i, i = 1, 2, \ldots$ are mutually disjoint and $\cup A_i = \Omega$, then

$$P(B) = \sum_i P(B|A_i)P(A_i)$$

and

$$P(A_j|B) = \frac{P(B|A_j)P(A_j)}{\sum_i P(B|A_i)P(A_i)}.$$

A *random variable* is a function $X : \Omega \to \mathbb{R}$. The *cumulative distribution function* of a random variable X is a function $F_X : R \to [0, 1]$ given by

$$F_X(x) = P(X \leq x).$$

The cumulative distribution function F has the following properties

1. $\lim_{x \to -\infty} F_X(x) = 0$ and $\lim_{x \to \infty} F_X(x) = 1$.
2. If $x \leq y$, then $F_X(x) \leq F_X(y)$.
3. F_X is right-continuous.

When F_X is differentiable, we can define the associated *probability density function* $p_X(x)$ as

$$p_X(x) = \frac{dF_X(x)}{dx}.$$

The *joint cumulative distribution function* of two random variables X and Y, denoted as $F_{XY}(x, y)$, is given by

$$F_{XY}(x, y) = P(X \leq x) \cap P(Y \leq y).$$

If its derivative exists, the associated joint probability density function is given by

$$p_{XY}(x, y) = \frac{\partial^2}{\partial x \partial y} F_{XY}(x, y).$$

Given $F_{XY}(x, y)$, the *marginal distribution functions* of X and Y can be calculated as

$$F_X(x) = P(X \leq x) = F_{XY}(x, \infty), \; F_Y(y) = P(Y \leq y) = F_{XY}(\infty, y).$$

It follows that the *marginal density functions* of X and Y are

$$p_X(x) = \int_{-\infty}^{\infty} F_{XY}(x,y)dy, \;\; p_Y(y) = \int_{-\infty}^{\infty} F_{XY}(x,y)dx.$$

The *conditional density function* of X given Y is given by

$$p_{X|Y}(x|y) = \frac{p_{XY}(x,y)}{p_Y(y)}.$$

The density function of X can also be calculated as

$$p_X(x) = \int_{-\infty}^{\infty} p_{X|Y}(x|y)p_Y(y)dy.$$

If X and Y are independent random variables, then the following statements holds and are equivalent to each other:

1. $F_{XY}(x,y) = F_X(x)F_Y(y)$.
2. $p_{XY}(x,y) = p_X(x)p_Y(y)$.
3. $p_{X|Y}(x|y) = p_X(x)$.

A random variable X is completely specified by its distribution function $F_X(x)$ or density function $p_X(x)$. In many situations, $F_X(x)$ or $p_X(x)$ are difficult to obtain. It turns out the *mean* μ_X and *variance* σ_X^2 may provide us enough (useful) information about X. The mean and variance of a random variable X are defined as follows:

$$\begin{aligned} \mu_X &= \mathbb{E}[X] = \int_{-\infty}^{\infty} x p_X(x)dx, \\ \sigma_X^2 &= \mathbb{E}\big[(X - \mathbb{E}[X])^2\big] = \int_{-\infty}^{\infty} (X - \mathbb{E}[X])^2 p_X(x)dx. \end{aligned}$$

We denote $\mathbb{E}[\cdot]$ as the *expectation operator*. Since $\mathbb{E}[\cdot]$ is a linear operator, σ_X^2 can also be calculated as

$$\sigma_X^2 = \mathbb{E}[X^2] - \big(\mathbb{E}[X]\big)^2.$$

If X is a zero-mean random variable, i.e., $\mathbb{E}[X] = 0$, then $\sigma_X = \mathbb{E}[X^2]$. The kth *moment* of X is $m_k = \mathbb{E}[X^k]$ and the kth *central moment* is $\mu_k = \mathbb{E}\big[(X - \mathbb{E}[X])^k\big]$.

The *covariance* of two random variables X and Y is defined as $\mathbb{E}\big[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\big]$. X and Y are *uncorrelated* if $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$. If X and Y are uncorrelated, it is easy to verify that the covariance of X and Y is equal to zero. Clearly if X and Y are independent, then they are uncorrelated. However the converse does not hold in general.

Introduce conditional distribution function first. **LS**

The conditional expectation of X given $Y = y$ is

$$\mathbb{E}[X|Y = y] = \int_{-\infty}^{\infty} x p_{X|Y}(x|y)dx$$

which is a number that depends on the value of y. Similarly, the conditional expectation of X given Y is

$$\mathbb{E}[X|Y] = \int_{-\infty}^{\infty} x p_{X|Y}(x|Y) dx$$

which is also a random variable that depends on Y, i.e., it is a *function of the random variable* Y. The following property is very important and has great practical value in evaluating $\mathbb{E}[X]$:

$$\mathbb{E}[X] = \mathbb{E}_Y\big[\mathbb{E}_X[X|Y]\big],$$

i.e., we first find the conditional expectation of X (conditioned on Y), and then remove the condition by taking the expectation with respect to Y. From this property, one can easily verify that if X and Y are independent, then

$$\mathbb{E}[X|Y] = \mathbb{E}[X].$$

Furthermore if X and Y are jointly independently of Z, then

$$\mathbb{E}[XY|Z] = \mathbb{E}[X|Z]\mathbb{E}[Y|Z].$$

Random Processes

A random process $X(t)$ is a generalization of a random variable. For a random variable, each experiment leads to a number (or a vector), while for a random process, each experiment leads to a function. For a fixed outcome $\omega \in \Omega$, one obtains the function $X(t, \omega)$, which is also called the *sample path* or *sample function* of the process. For a fixed t, $X(t, \omega)$ is a random variable with the underlying probability space Ω. The *mean process* of $X(t)$ is the time function $\mathbb{E}[X(t)]$. The autocorrelation of $X(t)$ is $\mathbb{E}[X(t_1)X(t_2)^T]$ and the autocovariance of $X(t)$ is $\mathbb{E}\big[\big(X(t_1) - m(t_1)\big)\big(X(t_2) - m(t_2)\big)^T\big]$.

A random process $X(t)$ is called a *Gaussian random process* if for any finite set $\{t_1, t_2, \ldots, t_N\}$, the random variables $\{X(t_1), X(t_2), \ldots, X(t_N)\}$ have a joint Gaussian distribution, i.e., their joint probability density function is given by

$$p_X(x) = \frac{1}{(2\pi)^{N/2}\sqrt{\det[\mathcal{C}_X]}} \exp\left[-\frac{1}{2}(x - m_X)^T \mathcal{C}_X^{-1}(x - m_X)\right] \tag{2.1}$$

where $m_X = [m_X(t_1)\ m_X(t_2)\ \ldots\ m_X(t_N)]^T$ is the mean vector and $\mathcal{C}_X = \big[\mathrm{cov}\big(X(t_i), X(t_j)\big)\big]$ is the covariance matrix. Gaussian processes have the following properties.

Theorem 2.1. *Let $X(t)$ be a Gaussian process. Then $X(t)$ is completely determined by m_X and $\mathcal{C}_X$.*

Theorem 2.2. *Let X and Y have a joint Gaussian distribution with mean and covariance given by*

$$\mu = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} \text{ and } \Sigma = \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix}.$$

Then X conditioned on $Y = y$ is Gaussian with mean and covariance given by

$$\mu_{X|Y=y} = \bar{x} + \Sigma_{xy}\Sigma_y^{-1}(y - \bar{y}) \text{ and } \Sigma_{X|Y=y}\Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{yx}.$$

In other words,

$$\mathbb{E}[X|Y=y] = \bar{x} + \Sigma_{xy}\Sigma_y^{-1}(y-\bar{y}). \tag{2.2}$$

The proof can be found in Anderson and Moore [AM90].

Stability of stochastic systems

This subsection should describe the different types of stability for a stochastic system (mean square, almost surely, convergence in distribution). I have put this in a separate file, `stability.tex`, just in case it gets large and needs to turn into its own section. We should include the definition of the Ricatti operator here (if it hasn't come up already) and also make sure to include Jensen's inequality. **LS**

Consider the following system dynamics:

$$x_{k+1} = f(x_k, w_k), \tag{2.3}$$

where x_0 and w_k are random vectors. System (2.3) is said to be

1. *second moment stable* if
$$\lim_{k\to\infty} \mathbb{E}[||x_k||^2] = 0,$$
2. *almost sure stable* if
$$P(\lim_{k\to\infty} ||x_k|| = 0]) = 1,$$

where the expectation is taken with respect to x_0 and $w_i, i = 0, \ldots, k$.

For a convex function f, $x_1, \ldots, x_n$ in its domain, and positive weights α_i, Jensen's inequality can be stated as:

$$f\left(\frac{\sum \alpha_i x_i}{\sum \alpha_i}\right) \leq \frac{\sum \alpha_i f(x_i)}{\sum \alpha_i}. \tag{2.4}$$

Jensen's inequality can also be stated in probabilistic form. Let X be a random variable and f be a convex function. Then

$$f\left(\mathbb{E}[X]\right) \leq \mathbb{E}\left(f(X)\right). \tag{2.5}$$

The above two inequalities are reversed if f is concave.

Markov Chains

Write up this subsection, which should include the relevant results that we will need in later chapters. I have create this as a separate file, `markov.tex`, in case we want to move it around later. **VG**

2.2 Optimal Estimation

Minimum mean square error estimator

Suppose we wish to know some quantity X, and we are not able to make a direct and accurate measurement of X. However we can make some indirect measurement Y that is related to X. Our task is to get an "optimal" estimate of X from Y.

One question that immediately arises before we attempt to solve the estimation problem is: what is a good estimate and when an estimate is "optimal"?

Intuitively a "good" estimate should make the estimation error $\hat{X} - X$ "small" since we wish to reconstruct X as perfectly as possible. An "optimal" estimate should make $\hat{X} - X$ the "smallest" among all other estimates. Many metrics can be used to define the size of the error $\hat{X} - X$ (hence we are able to say if it is "small" or not). Since $\hat{X} - X$ is typically a random variable, the metric that we shall use throughout the book is the following *mean squared error* (MSE)

$$\mathbb{E}[(\hat{X} - X)^T(\hat{X} - X)].$$

Therefore given $Y = y$ (i.e., the measurement that we take), our task is to construct the optimal estimate $\hat{X}$ that minimizes

$$\mathbb{E}[(\hat{X} - X)^T(\hat{X} - X)|Y = y].$$

It turns out that the optimal $\hat{X}$ has a very simple form, given in the following theorem.

Theorem 2.3. *The optimal estimate $\hat{X}^*$ that minimizes*

$$\mathbb{E}[(\hat{X} - X)^T(\hat{X} - X)|Y = y]$$

is given by the following conditional expectation of X

$$\hat{X}^* = \mathbb{E}[X|Y = y].$$

Proof. We can rewrite $\mathbb{E}[(\hat{X} - X)^T(\hat{X} - X)|Y = y]$ as follows

$$\begin{aligned} & \mathbb{E}[(\hat{X} - X)^T(\hat{X} - X)|Y = y] \\ = \; & \mathbb{E}[X^T X|Y = y] - 2\hat{X}^T\mathbb{E}[X|Y = y] + \hat{X}^T\hat{X} \\ = \; & \big(\hat{X} - \mathbb{E}[X|Y = y]\big)^T\big(\hat{X} - \mathbb{E}[X|Y = y]\big) + \mathbb{E}\big[X^T X - \mathbb{E}[X]^T\mathbb{E}[X^T]|Y = y\big]. \end{aligned}$$

Since $\mathbb{E}\big[X^T X - \mathbb{E}[X]^T\mathbb{E}[X^T]|Y = y\big]$ is independent of $\hat{X}$, we conclude that

$$\hat{X}^* = \mathbb{E}[X|Y = y].$$

□

$\hat{X}^* = \mathbb{E}[X|Y = y]$ is also called the *minimum mean squared error* (MMSE) estimate of X.

Example 2.1 Estimate a Gaussian random variable
Consider the following equation

$$Y = X + N \tag{2.6}$$

where X and N are both scalar zero-mean Gaussian random variables with covariances σ_x and σ_n respectively. Further assume X and N are uncorrelated. Suppose we make a measurement of X and get y. The MMSE estimate of X is then given by

$$\hat{X} \quad = \quad \mathbb{E}[X|Y = y] = \frac{\sigma_x}{\sigma_x + \sigma_n} y.$$

∇

Sampling of a continuous-time system

Note: This text is now in separate file, `sampling.tex`, so that we can pull it out more easily if we decide we don't wan to include it. [RMM, 21 Dec 09]

A wide variety of physical systems are modeled in the continuous-time domain. In this book, we focus on continuous-time systems with dynamics of the form

$$\frac{dx}{dt} = A_c x + B_c u + w, \quad y = C_c + v, \tag{2.7}$$

where $x(t) \in \mathbb{R}^n$ is the state vector with unknown initial value $x(0)$, $u(t) \in \mathbb{R}^p$ is the input vector, $y(t) \in \mathbb{R}^m$ is the observation vector, and $w(t)$ and $v(t)$ are process disturbance and measurement noise. We assume $w(t)$ and $v(t)$ are mutually uncorrelated zero-mean Gaussian processes with autocovariances

$$\mathbb{E}[w(s)w(t)^T] = \delta_{st}\Sigma_{wc}, \quad \mathbb{E}[v(s)v(t)^T] = \delta_{st}\Sigma_{vc},$$

where $\delta_{st} = 1$ if $s = t$ and $\delta_{st} = 0$ otherwise.

As more and more controllers are implemented digitally, we need a procedure to convert the continuous-time system (2.7) into an equivalent discrete-time system. This procedure is called *sampling* or *discretization*. A frequently seen approach to implement the control law on a digital computer is to use a digital to analogue converter that holds the analog signal until the next time step, called *zero-order-hold* control.

Consider the following periodic sampling scheme: we sample the system (2.7) at time instances $t = k\tau,\ k = 0, 1, \ldots$, where $\tau > 0$ is the *sampling period*. It can be shown (see Astrom-Wittenmark) that the equivalent discrete-time system of (2.7) is given by

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad y_k = Cx_k + v_k, \tag{2.8}$$

where x_k and y_k correspond to $x(t)$ and $y(t)$ at time $t = k\tau$, and A, B and C are given by

$$A = e^{A_c\tau}, \quad B = \int_0^\tau e^{A_c t} dt B_c, \quad C = C_c. \tag{2.9}$$

In the discrete-time setting, the process and measurement noises are also uncorrelated zero-mean Gaussian random processes with covariance

$$\mathbb{E}[w_s w_k] = \delta_{sk}\Sigma_w, \quad \mathbb{E}[v_s v_k^T] = \delta_{sk}\Sigma_v,$$

where

$$\Sigma_w = \int_0^\tau e^{A_c t}\Sigma_{wc}e^{A_c^T t}dt, \quad \Sigma_v = \Sigma_{vc}.$$

LS The following method from wikipedia for computing Σ_w needs to be verified.

Computing Σ_w directly from the above formula is sometimes difficult due to the integral of matrix exponentials. An easier approach to compute it is given as follows. Define M and N as

$$M = \begin{bmatrix} -A_c & \Sigma_{wc} \\ 0 & A_c^T \end{bmatrix}\tau, \quad N = e^M.$$

Then can be shown that

$$N = \begin{bmatrix} * & X^{-1}\Sigma_w \\ 0 & X^T \end{bmatrix}.$$

Therefore Σ_w can be computed from

$$\Sigma_w = (X^T)^T X^{-1}\Sigma_w,$$

i.e., Σ_w is obtained by multiplying the transpose of the lower-right submatrix of N with the upper-right submatrix of N.

Most of the results developed in this book also extend to cases where the sensor measurement y_k involves a direct input term, i.e.,

$$y_k = Cx_k + Du_k + v_k. \tag{2.10}$$

For simplicity, we shall use the system model as described by (2.8) for the remainder of the book unless otherwise explicitly stated.

Kalman filtering

Consider the following system as described by equation(2.8):

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad y_k = Cx_k + v_k, \tag{2.11}$$

where $x_k \in \mathbb{R}^n$ is the state vector with unknown initial value x_0, $u_k \in \mathbb{R}^p$ is the input vector, $y_k \in \mathbb{R}^m$ is the observation vector, and w_k and v_k are process and measurement noises (or disturbances).

Clearly nothing can be said on any estimator without defining a structure on w_k and v_k. In this book, we are particularly interested in w_k and v_k that have the following properties:

- w_k and v_k are zero-mean Gaussian random vectors;
- $\mathbb{E}[w_k w_j^T] = \delta_{kj}\Sigma_w$ with $\Sigma_w \geq 0$;
- $\mathbb{E}[v_k v_j^T] = \delta_{kj}\Sigma_v$ with $\Sigma_v > 0$;
- $\mathbb{E}[w_k v_j^T] = 0\ \forall j, k$,

where $\delta_{kj} = 0$ if $k \neq j$ and $\delta_{kj} = 1$ otherwise. We also assume the initial value x_0 of system (2.11) is a zero-mean Gaussian random vector that is uncorrelated with w_k and v_k for all $k \geq 0$. The covariance of x_0 is given by $\Pi_0 \geq 0$. Furthermore we assume $(A, \sqrt{Q})$ is stabilizable and (A, C) is detectable.

Let $Y_k = \{y_0, y_1, \ldots, y_k\}$ be the measurements available at time k and $U_k = \{u_0, u_1, \ldots, u_k\}$ be the input applied to the system up to time k. We are interested in looking for the MMSE $\hat{x}_k$ of x_k at each time $k \geq 0$ given Y_k and U_{k-1}. From Theorem 2.3, we know that $\hat{x}_k$ is given by

$$\hat{x}_k = \mathbb{E}[x_k | Y_k, U_{k-1}], \tag{2.12}$$

and the corresponding error covariance P_k is given by

$$P_k = \mathbb{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T | Y_k, U_{k-1}]. \tag{2.13}$$

Calculating $\hat{x}_k$ and P_k according to equation (2.12) and (2.13) is not trivial and is computationally intensive as k increases. The celebrated Kalman filter provides a simple and elegant way to compute $\hat{x}_k$ and P_k recursively.

The Kalman filter [Kal60] is a well-established methodology for model-based fusion of sensor data [GA93, Gus00, May79, KSH00, AM90] that has played a central role in systems theory and has found wide applications in many fields such as control, signal processing, and communications.

Assume that $\hat{x}_{k-1}$ and P_{k-1} defined as in equation (2.12) and (2.13) are available. Consider the one-step state prediction $\hat{x}_{k|k-1}$ (also called the *a priori state estimate*) given by

$$\hat{x}_{k|k-1} \quad = \quad \mathbb{E}[x_k | Y_{k-1}, U_{k-1}]$$

and the associated estimation error covariance (also called the *a priori error covariance*) $P_{k|k-1}$ given by

$$P_{k|k-1} \quad = \quad \mathbb{E}[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^T | Y_{k-1}, U_{k-1}].$$

From (2.11), we have

$$\begin{aligned} \hat{x}_{k|k-1} &= \mathbb{E}[x_k | Y_{k-1}, U_{k-1}] \\ &= \mathbb{E}[Ax_{k-1} + Bu_{k-1} + w_{k-1} | Y_{k-1}, U_{k-1}] \\ &= A\hat{x}_{k-1} + Bu_{k-1}, \end{aligned} \tag{2.14}$$

where we use the fact that w_{k-1} is independent of any y_t $(t \leq k-1)$ and the expectation operator is linear. Consequently,

$$P_{k|k-1} = AP_{k-1}A^T + \Sigma_w. \tag{2.15}$$

Now consider y_k conditioned on Y_{k-1} and U_{k-1} which has mean

$$\mathbb{E}[y_k | Y_{k-1}, U_{k-1}] = \mathbb{E}[Cx_k + v_k | Y_{k-1}, U_{k-1}] = C\hat{x}_{k|k-1}$$

and covariance

$$\mathbb{E}\big[\big(y_k - \mathbb{E}[y_k]\big)\big(y_k - \mathbb{E}[y_k]\big)^T | Y_{k-1}, U_{k-1}\big] = CP_{k|k-1}C^T + \Sigma_v,$$

where we have used the fact that v_k is independent of Y_{k-1}. The cross covariance of x_k and y_k conditioned on Y_{k-1} and U_{k-1} is given by

$$\mathbb{E}\big[\big(x_k - \mathbb{E}[x_k]\big)\big(y_k - \mathbb{E}[y_k]\big)^T | Y_{k-1}, U_{k-1}\big] = P_{k|k-1}C^T.$$

From the above analysis, we see that the random vector $[x_k' \quad y_k']'$ conditioned on Y_{k-1} and U_{k-1} is Gaussian with mean and covariance

$$\begin{bmatrix} \hat{x}_{k|k-1} \\ C\hat{x}_{k|k-1} \end{bmatrix} \text{ and } \begin{bmatrix} P_{k|k-1} & P_{k|k-1}C^T \\ CP_{k|k-1} & CP_{k|k-1}C^T + \Sigma_v \end{bmatrix}.$$

Therefore from Theorem 2.2, x_k conditioned on y_k (and on Y_{k-1} and U_{k-1}, i.e., conditioned on Y_k and U_{k-1}) has mean

$$\mathbb{E}[x_k|Y_k, U_{k-1}] = \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1})$$

and covariance

$$(I - K_kC)P_{k|k-1}$$

where $K_k = P_{k|k-1}C^T[CP_{k|k-1}C^T + \Sigma_v]^{-1}$ is the *Kalman gain*.

Let us summarize what we have said so far. Given the system (2.11), the MMSE estimate $\hat{x}_k$ of x_k is given by $\hat{x}_k = \mathbb{E}[x_k|Y_k, U_{k-1}]$, which can be computed recursively as follows

1. *time update:*

$$\begin{aligned} \hat{x}_{k|k-1} &= A\hat{x}_{k-1} + Bu_{k-1}, \\ P_{k|k-1} &= AP_{k-1}A^T + \Sigma_w. \end{aligned}$$

2. *measurement update:*

$$\begin{aligned} K_k &= P_{k|k-1}C^T[CP_{k|k-1}C^T + \Sigma_v]^{-1}, \\ \hat{x}_k &= \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}), \\ P_k &= (I - K_kC)P_{k|k-1}. \end{aligned}$$

The initial values of the recursion are set as $\hat{x}_0 = 0$ and $P_0 = \Pi_0$. The Kalman filter essentially consists of the above two update steps. The same set of equations are obtained if A, C, Q, R are time-varying (i.e., they are replaced by A_k, C_k, Q_k, R_k).

Lemma 2.1. *The Kalman gain K_k and the error covariance P_k satisfy*

$$K_k = P_kC^T\Sigma_v^{-1}. \tag{2.16}$$

Proof. Since $P_k = (I - K_kC)P_{k|k-1}$, it suffices to show

$$(I - K_kC)P_{k|k-1}C^T\Sigma_v^{-1} = K_k$$

which is equivalent to

$$\begin{aligned} & P_{k|k-1}C^T\Sigma_v^{-1} = K_k(I + CP_{k|k-1}C^T\Sigma_v^{-1}) \\ \iff\ & P_{k|k-1}C^T\Sigma_v^{-1} = P_{k|k-1}C^T[CP_{k|k-1}C^T + \Sigma_v]^{-1}(I + CP_{k|k-1}C^T\Sigma_v^{-1}) \\ \impliedby\ & \Sigma_v = (I + CP_{k|k-1}C^T\Sigma_v^{-1})^{-1}(CP_{k|k-1}C^T + \Sigma_v) \\ \iff\ & \Sigma_v = \Sigma_v(\Sigma_v + CP_{k|k-1}C^T)^{-1}(CP_{k|k-1}C^T + \Sigma_v) \end{aligned}$$

where the last equation holds trivially. □

Supplement

Note: Alternative version of the proof (from my OBC notes) [RMM, 21 Dec 09]

Alternate proof. K_k is defined as

$$K_k = P_{k|k-1}C^T(\Sigma_V + CP_{k|k-1}C^T)^{-1}.$$

Multiplying through by the inverse term on the right and expanding, we have

$$\begin{aligned} K_k(\Sigma_V + CP_{k|k-1}C^T) &= P_{k|k-1}C^T, \\ K_k\Sigma_V + K_kCP_{k|k-1}C^T &= P_{k|k-1}C^T, \end{aligned}$$

and hence

$$\begin{aligned} K_k\Sigma_V &= P_{k|k-1}C^T - K_kCP_{k|k-1}C^T, \\ &= (I - K_kC)P_{k|k-1}C^T = P_{k|k}C^T. \end{aligned}$$

The desired results follows by multiplying on the right by Σ_V^{-1}. □

Let $\mathbb{S}_+^n$ be the set of n by n positive semi-definite matrices. To simplify the analysis, define the function $h : \mathbb{S}_+^n \to \mathbb{S}_+^n$ as

$$h(X) \triangleq AXA^T + \Sigma_w, \tag{2.17}$$

and $\tilde{g} : \mathbb{S}_+^n \to \mathbb{S}_+^n$ as

$$\tilde{g}(X) \triangleq X - XC^T[CXC^T + \Sigma_v]^{-1}CX. \tag{2.18}$$

Further define $g : \mathbb{S}_+^n \to \mathbb{S}_+^n$ as

$$g(X) \triangleq h \circ \tilde{g} = AXA^T + \Sigma_w - AXC^T[CXC^T + \Sigma_v]^{-1}CXA. \tag{2.19}$$

For functions $f, f_1, f_2 : \mathbb{S}_+^n \to \mathbb{S}_+^n$, $f_1 \circ f_2$ is defined as

$$f_1 \circ f_2(X) \triangleq f_1\big(f_2(X)\big), \tag{2.20}$$

and f^t is defined as

$$f^t(X) \triangleq \underbrace{f \circ f \circ \cdots \circ f}_{t \text{ times}}(X). \tag{2.21}$$

With these definitions, it can be verified that in the Kalman filter time update and measurement update equations, $P_{k+1|k}$ and P_{k+1} satisfy

$$\begin{aligned} P_{k+1|k} &= h(P_k), \\ P_{k+1|k} &= g(P_{k|k-1}), \\ P_{k+1} &= \tilde{g}(P_{k+1|k}), \\ P_{k+1} &= \tilde{g} \circ h(P_k). \end{aligned}$$

The equation $g(X) = X$ or

$$AXA^T + \Sigma_w - AXC^T[CXC^T + \Sigma_v]^{-1}CXA = X \tag{2.22}$$

is called the *discrete-time algebraic Riccati equation* (DARE). When $(A, \sqrt{\Sigma_w})$ is stabilizable and (A, C) is detectable, P_k converges to a unique positive semi-definite matrix $\overline{P}$ which satisfies $\overline{P} = \tilde{g} \circ h(\overline{P})$. $\overline{P}$ is called the steady-state error covariance, and it reflects how well the estimate $\hat{x}_k$ approximates x_k in the steady state.

Properties of the Kalman filter

We first introduce a few well-known lemmas without proofs.

Lemma 2.2 (Matrix Inversion Lemma)**.** *Let $X > 0$. If $X = B^{-1} + CD^{-1}C'$, then the inverse of X can be written as*

$$X^{-1} = B - BC(D + C'BC)^{-1}C'B.$$

The second lemma is the Schur Complement lemma. It provides a set of equivalent relationships for a positive definite matrix M.

Lemma 2.3 (Schur Complement)**.** *Let*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

Then the following three conditions are equivalent to each other.

1. $M > 0$.
2. $A > 0$ *and* $S_A \triangleq D - CA^{-1}B > 0$.
3. $D > 0$ *and* $S_D \triangleq A - BD^{-1}C > 0$.

The last one is the Block Matrix Inversion lemma, which, as its name suggests, inverts a block matrix using the Schur complement of the matrix.

Lemma 2.4 (Block Matrix Inversion)**.** *Let*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} > 0.$$

Then M^{-1} can be computed as

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BS_A^{-1}CA^{-1} & -A^{-1}BS_A^{-1} \\ -S_A^{-1}CA^{-1} & S_A^{-1} \end{bmatrix},$$

or it can be computed as

$$M^{-1} = \begin{bmatrix} S_D^{-1} & -S_D^{-1}BD^{-1} \\ -D^{-1}CS_D^{-1} & D^{-1} + D^{-1}CS_D^{-1}BD^{-1} \end{bmatrix}.$$

Many useful properties of the functions h, $\tilde{g}$ and g are presented below.

Lemma 2.5. *For any $X, Y \in \mathbb{S}_+^n$, and $X \leq Y$,*

1. $h(X) \leq h(Y)$.
2. $g(X) \leq g(Y)$.
3. $\tilde{g}(X) \leq \tilde{g}(Y)$.
4. $\tilde{g}(X) \leq X$.

5. $g(X) \leq h(X)$.

When the measurement matrix C is invertible, the function g exhibits a very nice property. When we apply g to any $X \geq 0$, we have a bounded value. The following lemma gives this bound.

Lemma 2.6. *Assume C^{-1} exists and let $\overline{M} = C^{-1}RC^{-1'}$. Then for any $X \geq 0, \tilde{g}(X) \leq \overline{M}$.*

Proof. For any $t > 0$, we have $\tilde{g}(t\overline{M}) = \frac{t}{t+1}\overline{M} \leq \overline{M}$. For all $X \geq 0$, since $\overline{M} > 0$, it is clear that there exists $t_1 > 0$ such that $t_1\overline{M} > X$. Therefore $\tilde{g}(X) \leq \tilde{g}(t_1\overline{M}) \leq \overline{M}$. □

The steady-state error covariance $\overline{P}$ has the following property.

Lemma 2.7. $\overline{P} \leq h(\overline{P})$.

Proof. Let P^* satisfy $P^* = g(P^*)$. Then one can verify that $\overline{P} = \tilde{g}(P^*)$. Since $g = h \circ \tilde{g}$, we have

$$\overline{P} = \tilde{g}(P^*) \leq P^* = g(P^*) = h \circ \tilde{g}(P^*) = h(\overline{P}).$$

□

Let $0 \leq \lambda \leq 1$. Consider the following modified DARE.

$$g_\lambda(X) \triangleq AXA^T + \Sigma_w - \lambda AXC^T[CXC^T + \Sigma_v]^{-1}CXA = X. \tag{2.23}$$

The modified DARE will be studied in detail in Chapter 6 and the parameter λ will represent data packet arrival rate. Some preliminary results on the modified DARE are stated in the following lemmas. The proofs are omitted and can be found in the appendix of [SSS+03].

Lemma 2.8. *Let the operator*

$$\phi(K, X) = (1 - \lambda)(AXA' + \Sigma_w) + \lambda(FXF' + V) \tag{2.24}$$

where $F = A + KC, \quad V = \Sigma_w + K\Sigma_v K'$. Assume $X \in \mathbb{S}_+^n$, $\Sigma_v > 0$, $\Sigma_w \geq 0$, and $(A, \Sigma_w^{\frac{1}{2}})$ is controllable. Then the following facts are true:

1. *With $K_X = -AXC' (CXC' + \Sigma_v)^{-1}$, $g_\lambda(X) = \phi(K_X, X)$*
2. *$g_\lambda(X) = \min_K \phi(K, X) \leq \phi(K, X)$, $\forall K$*
3. *If $X \leq Y$, then $g_\lambda(X) \leq g_\lambda(Y)$*
4. *If $\lambda_1 \leq \lambda_2$ then $g_{\lambda_1}(X) \geq g_{\lambda_2}(X)$*
5. *If $\alpha \in [0, 1]$, then*
   $g_\lambda(\alpha X + (1 - \alpha)Y) \geq \alpha g_\lambda(X) + (1 - \alpha)g_\lambda(Y)$
6. *$g_\lambda(X) \geq (1 - \lambda)AXA' + \Sigma_w$*
7. *If $\bar{X} \geq g_\lambda(\bar{X})$, then $\bar{X} > 0$*

8. *If X is a random variable, then*
   $(1-\lambda)A\mathbb{E}[X]A' + \Sigma_w \leq \mathbb{E}[g_\lambda(X)] \leq g_\lambda(\mathbb{E}[X])$

The next two Lemmas show that when the modified DARE has a solution $\hat{P}$, this solution is also stable, i.e., every sequence based on the difference Riccati equation $P_{t+1} = g_\lambda(P_t)$ converges to $\hat{P}$ for all initial positive semidefinite conditions $P_0 \geq 0$.

Lemma 2.9. *Define the linear operator*

$$\mathcal{L}(Y) = (1-\lambda)(AYA') + \lambda(FYF')$$

Suppose there exists $\overline{Y} > 0$ such that $\overline{Y} > \mathcal{L}(\overline{Y})$.

1. *For all $W \geq 0$,*
$$\lim_{k\to\infty} \mathcal{L}^k(W) = 0$$

2. *Let $U \geq 0$ and consider the linear system*
$$Y_{k+1} = \mathcal{L}(Y_k) + U \quad \text{initialized at} \quad Y_0.$$
   Then, the sequence Y_k is bounded.

Lemma 2.10. *Consider the operator $\phi(K, X)$ defined in equation (6.28). Suppose there exists a matrix $\overline{K}$ and a positive definite matrix Z such that*

$$Z > 0 \quad \text{and} \quad Z > \phi(\overline{K}, Z).$$

Then, for any P_0, the sequence $P_k = g_\lambda^k(P_0)$ is bounded, i.e., there exists $M_{P_0} \geq 0$ dependent of P_0 such that

$$P_k \leq M \quad \text{for all} \quad k.$$

2.3 Linear Quadratic Optimal Control

The optimal linear quadratic regulator problem is posed as follows. Consider the process

$$x_{k+1} = Ax_k + Bu_k$$

with the initial condition x_0, where $x_k \in \mathbf{R}^n$ is the state and $u_k \in \mathbf{R}^m$ is the control input that needs to be designed to minimize the cost

$$J_K = \sum_{k=0}^{K} \left(x_k^T Q x_k + u_k^T R u_k\right) + x_{k+1}^T P_{k+1} x_{k+1},$$

with $Q > 0$ and $R \geq 0$. If the parameter K is finite, the problem is termed the finite horizon LQR problem. The case when $K \to \infty$ is termed the infinite horizon LQR problem. We shall assume that the par (A, B) is controllable. In general, the results given below extend to the case when the matrices A, B, Q and R are time varying.

The solution to the finite horizon problem can be obtained through standard dynamic programming arguments. The following theorem summarizes the results.

Theorem 2.4. *Consider the finite horizon LQR problem posed above. The optimal control law is a linear function of the state*

$$u_k = -\left(B^T P_{k+1} B + R\right)^{-1} B^T P_{k+1} A x_k,$$

where the matrix P_k evolves according to the backward Riccati recursion

$$P_k = A^T P_{k+1} A + Q - A^T P_{k+1} B \left(B^T P_{k+1} B + R\right)^{-1} B^T P_{k+1} A$$

with the initial condition P_{k+1}. Moreover, the achieved cost is given by $x_0^T P_0 x_0$.

Proof. We begin by rewriting the cost function J_K to identify terms in the cost that depend on x_k and u_k:

$$\begin{aligned} J_K &= \sum_{k=1}^{K-1} \left(x_k^T Q x_k + u_k^T R u_k\right) + T_k \\ T_k &= x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T P_{k+1} x_{k+1}. \end{aligned}$$

The only term in the cost that can be affected by the choice of u_k is T_k. To choose u_k, we minimize T_k by a completion of squares argument. We obtain

$$\begin{aligned} T_k &= x_k^T Q x_k + u_k^T R u_k + (A x_k + B u_k)^T P_{k+1} (A x_k + B u_k) \\ &= x_k^T P_k x_k + (u_k + S_k^{-1} B^T P_{k+1} A x_k)^T S_k (u_k + S_k^{-1} B^T P_{k+1} A x_k), \end{aligned}$$

where

$$\begin{aligned} S_k &= B^T P_{k+1} B + R \\ P_k &= Q + A^T P_{k+1} A - A^T P_{k+1} B S_k^{-1} B^T P_{k+1} A. \end{aligned}$$

Thus, the optimal choice of u_k is

$$u_k = -S_k^{-1} B^T P_{k+1} x_k.$$

With the optimal choice of u_k the term T_k reduces to $x_k^T P_k x_k$. Thus, the cost function J_K can be rewritten as

$$\begin{aligned} J_K &= \sum_{k=1}^{K-2} \left(x_k^T Q x_k + u_k^T R u_k\right) + T_{K-1} \\ T_{K-1} &= x_{K-1}^T Q x_{K-1} + u_{K-1}^T R u_{K-1} + x_k^T P_k x_k. \end{aligned}$$

Thus, the problem of choosing u_{K-1} is formally identical to the problem that we solved above for choosing u_k, and the same argument can be repeated at any time step recursively. At a general time k, the control input u_k given $r_k = i$ is given by

$$u_k = -\left(B^T P_{k+1} B + R\right)^{-1} B^T P_{k+1} A x_k,$$

where the matrix P_k evolves according to the backward Riccati recursion

$$P_k = A^T P_{k+1} A + Q - A^T P_{k+1} B \left(B^T P_{k+1} B + R\right)^{-1} B^T P_{k+1} A$$

with the final condition P_{k+1}. Moreover, the cost J_0 obtained through this procedure equals $x_0^T P_0 x_0$. □

For the infinite horizon case, we provide the solution without proof below.

Theorem 2.5. *Consider the infinite horizon LQR problem posed above. The optimal control law is a linear function of the state*

$$u_k = -\left(B^T P B + R\right)^{-1} B^T P A x_k,$$

where the matrix P is the unique positive semi-definite solution of the Riccati equation

$$P = A^T P A + Q - A^T P B \left(B^T P B + R\right)^{-1} B^T P A.$$

Moreover, the achieved cost is given by $x_0^T P x_0$.

2.4 LQG Problem

The finite horizon Linear Quadratic Guassian optimal control problem is posed as follows. Consider the process

$$x_{k+1} = Ax_k + Bu_k + w_k$$

with the initial condition x_0 as zero mean Gaussian, where $x_k \in \mathbf{R}^n$ is the state, $u_k \in \mathbf{R}^m$ is the control input that needs to be designed, and w_k is the process noise modeled Gaussian and white with mean zero and covariance $\Sigma_w > 0$. The process is observed using a sensor that generates measurements of the form

$$y_k = Cx_k + v_k,$$

where the sensor noise v_k is modeled Gaussian and white with mean zero and covariance $\Sigma_v > 0$. The noise sequences $\{w(j)\}$, $\{v(j)\}$ and the initial condition x_0 are assumed to be independent. The cost function that needs to be minimized is

$$J_K = \sum_{k=0}^{K} \mathbb{E}[\left(x_k^T Q x_k + u_k^T R u_k\right)] + \mathbb{E}[x_{k+1}^T P_{k+1} x_{k+1}],$$

with $Q > 0$ and $R \geq 0$. The expectation is taken with respect to all the random parameters in the system. The controller at time k is allowed access to the measurements until time k and control inputs until time $k-1$. If the parameter K is finite, the problem is termed the finite horizon LQG problem. As $K \to \infty$, the cost would necessarily diverge. Thus, the infinite horizon LQG problem considers the cost

$$J_\infty = \lim_{K \to \infty} \frac{1}{K} J_K.$$

We shall assume that the par (A, B) is controllable and (A, C) is observable. In general, the results given below extend to the case when the matrices A, B, C, Q and R are time varying.

The solution to the finite horizon problem is provided by the separation principle. The principle essentially states that the optimal control input is calculated as the input in the LQR problem, but with the state x_k replaced by the minimum mean squared error (MMSE) estimate of the state x_k based on the measurements

until time k and control inputs until time $k-1$. Note that the estimate can be calculated recursively through the Kalman filter. The following theorem summarizes the results.

Theorem 2.6. *Consider the finite horizon LQG problem posed above. The optimal control law is a linear function of the state*

$$u_k = -\left(B^T P_{k+1} B + R\right)^{-1} B^T P_{k+1} A \hat{x}_k,$$

where the matrix P_k evolves according to the backward Riccati recursion

$$P_k = A^T P_{k+1} A + Q - A^T P_{k+1} B \left(B^T P_{k+1} B + R\right)^{-1} B^T P_{k+1} A$$

with the final condition P_{k+1}, and $\hat{x}_k$ is the MMSE estimate of the state x_k based on the measurements until time k and control inputs until time $k-1$, calculated, e.g., using a Kalman filter.

Proof. The proof again follows dynamic programming arguments. We begin by rewriting the cost function J_K to identify terms in the cost that depend on x_k and u_k:

$$\begin{aligned} J_K &= \mathbb{E}[\sum_{k=1}^{K-1} \left(x_k^T Q x_k + u_k^T R u_k\right)] + T_k \\ T_k &= \mathbb{E}[x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T P_{k+1} x_{k+1}]. \end{aligned}$$

The only term in the cost that can be affected by the choice of u_k is T_k. To choose u_k, we minimize T_k by a completion of squares argument. We obtain

$$\begin{aligned} T_k &= \mathbb{E}[x_k^T Q x_k + u_k^T R u_k + (A x_k + B u_k + w_k)^T P_{k+1} (A x_k + B u_k + w_k)] \\ &= \mathbb{E}[x_k^T P_k x_k + w_k^T P_{k+1} w_k + (u_k + S_k^{-1} B^T P_{k+1} A x_k)^T S_k (u_k + S_k^{-1} B^T P_{k+1} A x_k)], \end{aligned}$$

where we have used the fact that the process noise is white (hence w_k is independent of both x_k and u_k) and zero mean, and have defined

$$\begin{aligned} S_k &= B^T P_{k+1} B + R \\ P_k &= Q + A^T P_{k+1} A - A^T P_{k+1} B S_k^{-1} B^T P_{k+1} A. \end{aligned}$$

Note that the controller does not have access to x_k and hence the quadratic term cannot be minimized to zero. Instead, the controller estimates (based on the measurements until time k and control inputs until time $k-1$) the term $S_k^{-1} B^T P_{k+1} A x_k$ in the MMSE sense. Thus, the optimal choice of u_k is

$$u_k = -S_k^{-1} B^T P_{k+1} \hat{x}_k.$$

Denote by $\Lambda_{e,k}$ the error covariance thus obtained. Since the controller utilizes all control inputs until time $K-1$ while calculating u_k, $\Lambda_{e,k}$ does not depend on the choice of control inputs $u_0, \cdots, u_{K-1}$. With the optimal choice of u_k, the term T_k

reduces to $T_k = \mathbb{E}[x_k^T P_k x_k + w_k^T P_{k+1} w_k + \Lambda_{e,k}]$. Thus, the cost function J_K can be rewritten as

$$J_K = \mathbb{E}[\sum_{k=1}^{K-2} \left(x_k^T Q x_k + u_k^T R u_k\right)] + T_{K-1} + w_k^T P_{k+1} w_k + \Lambda_{e,k}$$
$$T_{K-1} = \mathbb{E}[x_{K-1}^T Q x_{K-1} + u_{K-1}^T R u_{K-1} + x_k^T P_k x_k].$$

Now note that the terms $w_k^T P_{k+1} w_k$ and $\Lambda_{e,k}$ are not impacted by the choice of $u[K-1]$ and hence may be dropped from the minimization problem. Thus, the problem of choosing $u[K-1]$ is formally identical to the problem that we solved above for choosing u_k, and the same argument can be repeated at any time step recursively. At a general time k, the control input u_k given $r[k] = i$ is given by

$$u_k = -\left(B^T P_{k+1} B + R\right)^{-1} B^T P_{k+1} A \hat{x}_k,$$

where the matrix P_k evolves according to the backward Riccati recursion

$$P_k = A^T P_{k+1} A + Q - A^T P_{k+1} B \left(B^T P_{k+1} B + R\right)^{-1} B^T P_{k+1} A$$

with the final condition P_{k+1}. □

The separation principle also holds for the infinite horizon case. In particular, under the assumptions above, both the backward Riccati recursion in the control calculation and the forward Riccati recursion in the Kalman filter are replaced by the corresponding Riccati equations.

2.5 Further Reading

Exercises

RMM: This doesn't appear to be showing up in the exercises. Fix.

2.1 Show $\mathbb{E}[X|Y = y] = \frac{\sigma_x}{\sigma_x + \sigma_n} y$ in Example 2.1.

2.2 Recall that $\overline{P}$ is the steady-state error covariance of the Kalman filter. Prove the following.

(a) If $1 \leq t_1 \leq t_2$, then $h^{t_1}(\overline{P}) \leq h^{t_2}(\overline{P})$.

(b) $h(\overline{P}) \neq \overline{P}$.

Chapter 7
Information Flow and Consensus

In this chapter we move from the problem of estimation and control of a single system across a communications channel to the challenge of sensing, estimation and control of a multi-agent system, with the information available to the agents represented by a graph of interconnections. We begin with a review of the relevant concepts in graph theory, focused on the use of algebraic techniques to characterize the properties of the interconnection structure. We then apply these concepts to study the problem of a group of agents reaching consensus on a shared property of the system.

The contents of this chapter are currently based on slides from the EECI course, which were generated using some notes from Reza's course at Caltech. Need to go through and make sure that I am not directly making use of any of his material. **RMM**

7.1 Graph Theory

In this section we give a brief overview of the field of graph theory, focused on some of the algebraic methods that characterize the properties of the graph in terms of a set of matrices associated with it. These techniques will be very important for helping understand the interactions between dynamic agents across a graph, including the consensus problem in this chapter and the distributed estimation and control problems in the subsequent chapters. More detailed treatments are available in a number of textbooks, including Diestel [**?**], Godsil and Royle [**?**], and Horn and Johnson [**?**]. This section is based in part on a set of course notes originally developed by Reza Olfati-Saber [**?**].†

RMM: Need to run this section by Reza and make sure he is OK with the contents.

Basic Definitions

We define a *directed graph* as a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a set of *vertices* $\mathcal{V}$ and a set of *edges* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We represent a vertex (or *node*) as an element $v_i \in \mathcal{V}$ and an edge (or *link*) as a connection between two vertices, $e_{ij} = (v_i, v_j) \in \mathcal{E}$. We write $|\mathcal{V}|$ for the number of vertices in the graph, also known as the *order* of the graph. qAn edge has an orientation given by the ordering of the vertices, so the edge e_{ij} is distinct from the edge e_{ji}. We call v_i the head of the edge and v_j the tail. A directed graph is also referred to as a *digraph*.

We say that two vertices v_i and v_j are *adjacent* if there exists and edge $e = (v_i, v_j)$ and vertex v_j is called a *neighbor* of v_i. We write $\mathcal{N}_i =$ set of all neighbors of v_i and we say that a graph $\mathcal{G}$ is *complete* if all vertices are adjacent to each other. We define the *out-degree* of a vertex v_i, written $\deg_{\text{out}}(V_i)$, as the number of edges

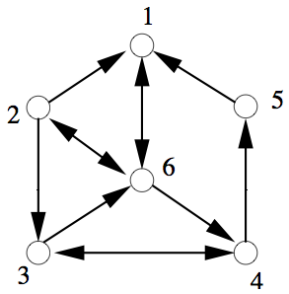


Figure 7.1: A graph with 6 vertices.

whose head is v_i. Similarly, the *in-degree* of a vertex v_i, $\deg_{\text{in}}(v_i)$ is the number of edges with tail v_i.

Example 7.1 6 node graph
Consider a graph given by the vertices $\mathcal{V} = \{1, 2, 3, 4, 5, 6\}$ and a set of edges

$$\mathcal{E} = \{(1,6), (2,1), (2,3), (2,6), (6,2), (3,4), \\ (3,6), (4,3), (4,5), (5,1), (6,1), (6,2), (6,4)\},$$

as shown in Figure 7.1. Node 1 has an in-degree of 3 and an out-degree of 1. Its neighbor set is given by $\mathcal{N}_1 = \{v_6\}$. Node 2 has an in-degree of 1 and an out-degree of 3. Its neighbor set is given by $\mathcal{N}_2 = \{v_1, v_2, v_6\}$. ∇

In many instances the orientation of the edges in a graph will not matter and we can ignore the ordering of the verticies in an edge $e_{ij} = (v_i, v_j)$. Formally, we will consider a graph to be *undirected* if $e_{ij} \in \mathcal{E}$ implies that $e_{ji} \in \mathcal{E}$. In these cases it will often be easier to simply say that the graph is undirected and consider an edge e_{ij} to be equivalent to an edge e_{ji}. For an undirected graph the indegree and outdegree are the same, so we simply refer to the *degree of a vertex*. An undirected graph is *regular* (or *k-regular*) if all vertices of a graph have the same degree k. A directed graph is *balanced* if the out-degree is equal to the in-degree at each vertex.

Example 7.2
Figure 7.3 shows three examples of graphs. ∇

Connectedness of Graphs

A key set of properties of a graph have to do with whether there exists paths that connect its nodes. Formally, a *path* is a subgraph $\pi = (\mathcal{V}, \mathcal{E}_\pi) \subset \mathcal{G}$ with distinct vertices $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$ and

$$\mathcal{E}_\pi := \{(v_1, v_2), (v_2, v_3), \ldots, (v_{m-1}, v_m)\}.$$

The *length* of π is defined as $|\mathcal{E}_\pi| = m - 1$. A *cycle* (or m-cycle) $C = (\mathcal{V}, \mathcal{E}_C)$ is a path (of length m) with an extra edge $(v_m, v_1) \in \mathcal{E}$. We define the *distance* between two vertices v and w as the length of the shortest path between them.

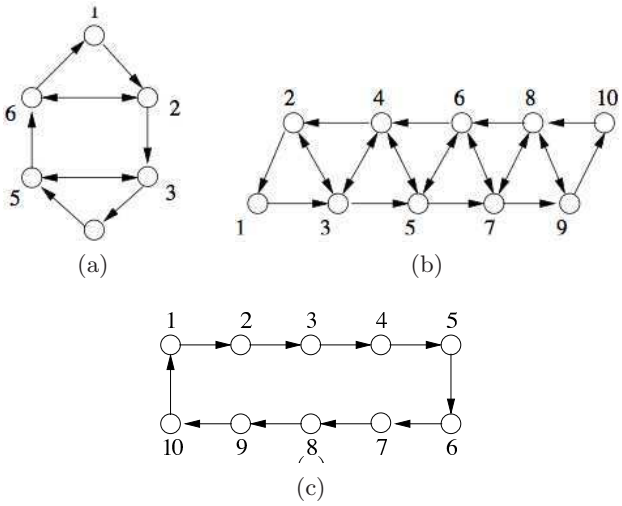


Figure 7.2: Examples of graphs with different properties.

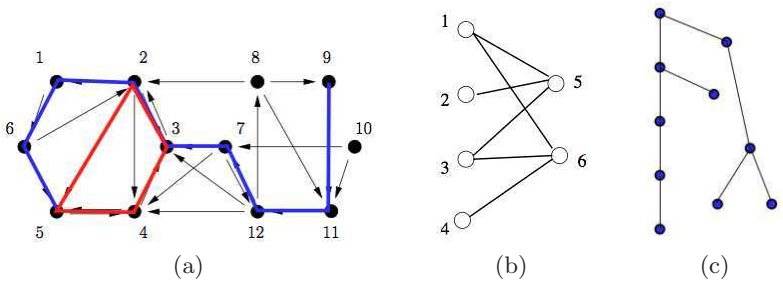


Figure 7.3: Examples of graphs with different properties.

An undirected graph $\mathcal{G}$ is called *connected* if there exists a path π between any two distinct vertices of $\mathcal{G}$. For a connected graph $\mathcal{G}$, the length of the maximum distance between two vertices is called the *diameter* of $\mathcal{G}$. A graph with no cycles is called *acyclic*. A *tree* is a connected acyclic graph.

A digraph is called *strongly connected* if there exists a directed path π between any two distinct vertices of $\mathcal{G}$. A digraph is called *weakly connected* if there exists an undirected path between any two distinct vertices of $\mathcal{G}$.

Example 7.3
Figure **??** shows examples of graphs and their connectedness properties. ∇

Matrices Associated with a Graph

In order to characterize the properties of a graph, we will use matrices to represent the structure of the graph. The properties of these matrices can then be related back to the properties of the graph.

The *adjacency matrix* $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ of a graph $\mathcal{G}$ of order n is given by:

$$a_{ij} := \begin{cases} 1 & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

The *degree matrix* of a graph as a diagonal $n \times n$ ($n = |\mathcal{V}|$) matrix

$$\Delta = \text{diag}\{\deg_{\text{out}}(v_i)\}$$

with diagonal elements equal to the out-degree of each vertex and zero everywhere else. The *Laplacian matrix* L of a graph is defined as

$$L = \Delta - A$$

. It follows from the definition that the row sums of the Laplacian are all 0.

Example 7.4 6 node graph
Consider the graph shown in Example **??**. The adjancy matrix and Laplacian are given by

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}, \qquad L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & 0 & 0 & -1 \\ 0 & 0 & 2 & -1 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & -1 & 0 & -1 & 0 & 3 \end{bmatrix}.$$

∇

Periodic Graphics and Weighted Graphs

A graph with the property that the set of all cycle lengths has a common divisor $k > 1$ is called *k-periodic*. A graph without cycles is said to be *acyclic*.

A *weighted graph* is graph $(\mathcal{V}, \mathcal{E})$ together with a map $\varphi : \mathcal{E} \to \mathbb{R}$ that assigns a real number $w_{ij} = \varphi(e_{ij})$ called a *weight* to an edge $e_{ij} = (v_i, v_j) \in \mathcal{E}$. The set of all weights associated with $\mathcal{E}$ is denoted by $\mathcal{W}$. A weighted graph can be represented as a triplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$.

In some applications it is natural to "normalize" the Laplacian by the outdegree. We define the *weighted Laplacian* as

$$\tilde{L} := \Delta^{-1} L = I - \tilde{A} = I - \Delta^{-1} A$$

, where $\tilde{A} = \Delta^{-1} A$ (weighted adjacency matrix).

Example 7.5 Weighted Laplacian for formation graph
Consider the graph in Figure 7.4. The weighted Laplacian is given by

$$L = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

∇

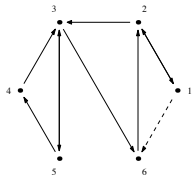


Figure 7.4: Formation control graph.

Gershgorin Disk Theorem

Add some explanatory text indicating what we are going to use all of this for. **RMM**

Theorem 7.1 (Gershgorin Disk Theorem)**.** *Let $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ and define the deleted absolute row sums of A as*

$$r_i := \sum_{j=1, j\neq i}^{n} |a_{ij}| \tag{7.1}$$

Then all the eigenvalues of A are located in the union of n disks

$$G(A) := \bigcup_{i=1}^{n} G_i(A), \text{ with } \quad G_i(A) := \{z \in \mathbb{C} : |z - a_{ii}| \leq r_i\} \tag{7.2}$$

Furthermore, if a union of k of these n disks forms a connected region that is disjoint from all the remaining $n - k$ disks, then there are precisely k eigenvalues of A in this region.

Sketch of proof. Let λ be an eigenvalue of A and let v be a corresponding eigenvector. Choose i such that $|v_i| = \max_j |v_j > 0$. Since v is an eigenvector,

$$\lambda v_i = \sum_i A_{ij} v_j \quad \Longrightarrow \quad (\lambda - a_{ii}) v_i = \sum_{i\neq j} A_{ij} v_j$$

Now divide by $v_i \neq 0$ and take the absolute value to obtain

$$|\lambda - a_{ii}| = |\sum_{j\neq i} a_{ij} v_j| \leq \sum_{j\neq i} |a_{ij}| = r_i$$

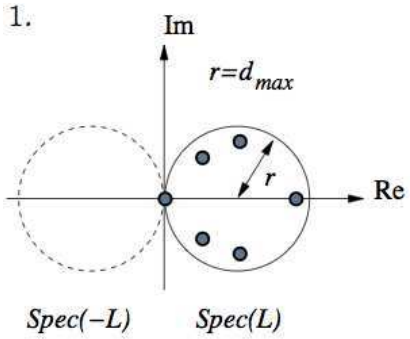
□

We can use the Gershgorin disk theorem to reason about the eigenvalues of the Laplacian and the weighted Laplacian.

Proposition 7.2. *Let L be the Laplacian matrix of a digraph $\mathcal{G}$ with maximum vertex out–degree of $d_{max} > 0$. Then all the eigenvalues of $A = -L$ are located in a disk*

$$B(\mathcal{G}) := \{s \in \mathbb{C} : |s + d_{max}| \leq d_{max}\} \tag{7.3}$$

that is located in the closed LHP of s-plane and is tangent to the imaginary axis at $s = 0$.



RMM: Figure is currently not referenced in the text

Figure 7.5: Graphical description of the Gershgorin disk theorem.

Proposition 7.3. *Let $\tilde{L}$ be the weighted Laplacian matrix of a digraph $\mathcal{G}$. Then all the eigenvalues of $A = -L$ are located inside a disk of radius 1 that is located in the closed LHP of s-plane and is tangent to the imaginary axis at $s = 0$.*

Another property of the Laplacian is that its rank determines the connectivity of the graph.

Theorem 7.4 (Olfati-Saber)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ be a weighted digraph of order n with Laplacian L. If $\mathcal{G}$ is strongly connected, then rank$(L) = n - 1$.*

RMM: check

The proof for the directed case can be found in standard textbooks on graph theory, such as those listed at the beginning of this section.†. The proof for the undirected case is available in [OSM04]. Note that for directed graphs, we require that $\mathcal{G}$ be strongly connected; the converse statement is not true.

Perron-Frobenius Theory

The spectrum of a matrix A is defined as $\text{spec}(A) = \{\lambda_1, \ldots, \lambda_n\}$, where λ_i, $i = 1, \ldots, n$ are the eigenvalues of A. The distance to the largest eigenvalue $\rho(A) = |\lambda_n| = \max_k |\lambda_k|$ is called the *spectral radius* of A.

Theorem 7.5 (Perron's Theorem, 1907)**.** *If $A \in \mathbb{R}^{n\times n}$ is a positive matrix ($A > 0$), then*

1. *$\rho(A) > 0$;*
2. *$r = \rho(A)$ is an eigenvalue of A;*
3. *There exists a positive vector $x > 0$ such that $Ax = \rho(A)x$;*
4. *$|\lambda| < \rho(A)$ for every eigenvalue $\lambda \neq \rho(A)$ of A, i.e. $\rho(A)$ is the unique eigenvalue of maximum modulus; and*
5. *$[\rho(A)^{-1}A]^m \to R$ as $m \to +\infty$ where $R = xy^T$, $Ax = \rho(A)x$, $A^Ty = \rho(A)y$, $x > 0$, $y > 0$, and $x^Ty = 1$.*

Theorem 7.6 (Perron's Theorem for Non–Negative Matrices)**.** *If $A \in \mathbb{R}^{n\times n}$ is a non-negative matrix ($A \geq 0$), then $\rho(A)$ is an eigenvalue of A and there is a non–negative vector $x \geq 0$, $x \neq 0$, such that $Ax = \rho(A)x$.*

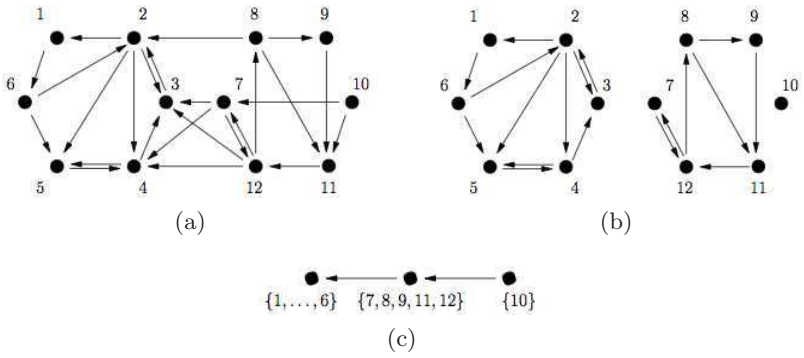


Figure 7.6: Irreducibility of a graph.

A directed graph is irreducible if, given any two vertices, there exists a path from the first vertex to the second. (Irreducible = strongly connected) A matrix is irreducible if it is not similar to a block upper triangular matrix via a permutation. A digraph is irreducible if and only if its adjacency matrix is irreducible.

Example 7.6
Consider the graph in Figure 7.6.

Complete example **RMM**

∇

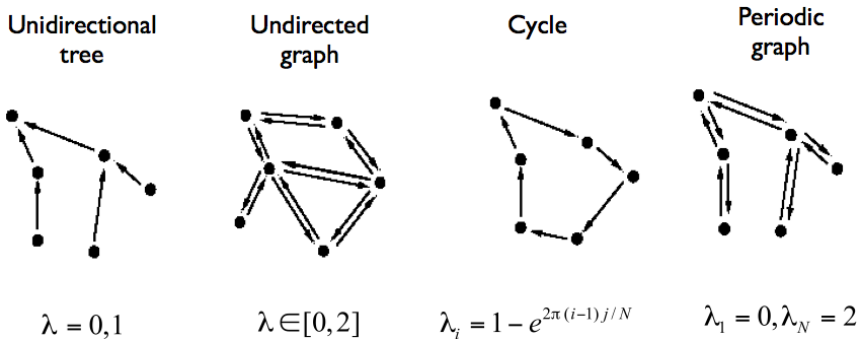
Theorem 7.7 (Frobenius)**.** *Let $A \in \mathbb{R}^{n\times n}$ and suppose that A is irreducible and non-negative. Then*

1. *$\rho(A) > 0$;*
2. *$r = \rho(A)$ is an eigenvalue of A;*
3. *There is a positive vector $x > 0$ such that $Ax = \rho(A)x$;*
4. *$r = \rho(A)$ is an algebraically simple eigenvalue of A; and*
5. *If A has h eigenvalues of modulus r, then these eigenvalues are all distinct roots of $\lambda^h - r^h = 0$.*

Using the Perron and Frobenius theorems, we can study additional properties of the Laplacian matrix of a graph. In particular, it can be shown that If $\mathcal{G}$ is strongly connected, the zero eigenvalue of L is simple. If $\mathcal{G}$ is aperiodic, all nonzero eigenvalues lie in the interior of the Gershgorin disk. If $\mathcal{G}$ is k-periodic, L has k evenly spaced eigenvalues on the boundary of the Gershgorin disk.

Theorem 7.8 (Variant of Courant-Fischer)**.** *Let $A \in \mathbb{R}^{n\times n}$ be a Hermitian matrix with eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ and let w_1 be the eigenvector of A associated with the eigenvalue λ_1. Then*

$$\lambda_2 = \min_{\substack{x \neq 0,\, x \in \mathbb{C}^n, \\ x \perp w_1}} \frac{x^* A x}{x^* x} = \min_{\substack{x^* x = 1, \\ x \perp w_1}} x^* A x \tag{7.4}$$



RMM: Figure is currently not referenced

Figure 7.7: Spectra of the Laplacian for classes of graphs.

Proof. Since A is Hermitian matrix, it is unitary diagonalizable (see Theorem **??**), i.e. $A = U\Lambda U^*$ where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Let $U = [w_1|w_2|\cdots|w_n]$ (w_k is the kth column of U). Then

$$\begin{aligned} x^*Ax &= x^*U\Lambda U^*x = (U^*x)^*\Lambda(U^*x) \\ &= \sum_{i=1}^{n} \lambda_i |(U^*x)_i|^2 = \sum_{i=1}^{n} \lambda_i |w_i^*x|^2 = \sum_{i=2}^{n} \lambda_i |w_i^*x|^2 \quad (x \perp w_1) \\ &\geq \lambda_2 \sum_{i=2}^{n} |w_i^*x|^2 = \lambda_2 \sum_{i=1}^{n} |w_i^*x|^2 \quad (x \perp w_1) = \lambda_2 \sum_{i=1}^{n} |(U^*x)_i|^2 \\ &= \lambda_2 (x^*UU^*x) = \lambda_2 x^*x \end{aligned} \tag{7.5}$$

Thus, for $x \perp w_1$ and $x \neq 0$

$$x^*Ax \geq \lambda_2 x^*x$$

where the equality is achieved with $x = w_2$. □

The second eigenvalue of the Laplacian λ_2 is called the *algebraic connectivity* of L.

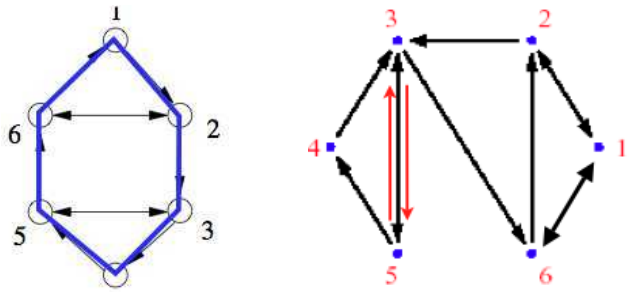
Cyclically Separable Graphs

Definition 7.1 (Cyclic separability)**.** A digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *cyclically separable* if and only if there exists a partition of the set of edges $\mathcal{E} = \cup_{k=1}^{n_c} \mathcal{E}_k$ such that each partition $\mathcal{E}_k$ corresponds to either the edges of a cycle of the graph, or a pair of directed edges ij and ji that constitute an undirected edge. A graph that is not cyclically separable is called *cyclically inseparable*.

Lemma 7.1. *Let L be the Laplacian matrix of a cyclically separable digraph $\mathcal{G}$ and set $u = -Lx, x \in \mathbb{R}^n$. Then $\sum_{i=1}^{n} u_i = 0, \forall x \in \mathbb{R}^n$ and $\mathbf{1} = (1, \ldots, 1)^T$ is the left eigenvector of L.*

Proof. The proof follows from the fact that by definition of cyclic separability. We have

$$-\sum_{i=1}^{n} u_i = \sum_{ij \in \mathcal{E}} (x_j - x_i) = \sum_{k=1}^{n_c} \sum_{ij \in \mathcal{E}_k} (x_j - x_i) = 0$$



(a) (b)

Figure 7.8: Cyclic separability.

because the inner sum is zero over the edges of cycles and undirected edges of the graph. □

Example 7.7 Cyclic separability

∇

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a digraph. We say $\mathcal{G}$ is *balanced* if and only if the in–degree and out–degree of all vertices of $\mathcal{G}$ are equal, i.e.

$$\deg_{\text{out}}(v_i) = \deg_{\text{in}}(v_i), \quad \forall v_i \in \mathcal{V} \tag{7.6}$$

Let $\mathcal{G}$ be a digraph with a weighted adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ that has the property $a_{ii} = 0$. Then, $\mathcal{G}$ is balanced if and only if $w_l = \mathbf{1}$.

Theorem 7.9. *A digraph is cyclically separable if and only if it is balanced.*

Proof. Assume the graph is cyclically separable. Then any arbitrary vertex v_i of the graph belongs to a finite number of cycles and/or undirected edges. The main property of a cycle is that corresponding to any directed edge arriving at a vertex, there is one edge leaving that vertex and therefore the in–degree and out-degree of any vertex are equal, i.e. the graph is balanced.

Now, let us assume that the graph is balanced, we show that it is cyclically separable. Suppose the opposite holds, meaning that the graph is not cyclically separable. Then there exists a directed edge (v_k, v_l) of the graph which does not belong to any cycles and/or undirected edges. Set $x_i = 0, \forall i \neq l$ and let $x_l = 1$. Define $u = -Lx$, we have $u_i = 0, \forall i \neq k$ and $u_k = x_l - x_k = 1$ (notice that $u_l = 0$ since k is not an out–neighbor of l). Thus $\sum_{i=1}^{n} u_i = \mathbf{1}^T u = 1 \neq 0$. But we know that $\mathbf{1}$ eigenvector of L for any balanced graph, thus $\mathbf{1}^T u = -\mathbf{1}^T L x = 0, \forall x$. This is a contradiction which means every directed edge of a balanced graph belongs to a cycle or an undirected edge, i.e. the graph is cyclically separable. □ □

7.2 Consensus algorithms

The *consensus problem* involves a group of agents reaching an agreement on a decision in a decentralized problem. In this sectoin we describe one approach to

RMM: Rewrite solving this problem, with the agents communicating on a graph.†

Average Consensus

Consider a collection of N agents that communicate along a set of undirected links described by a graph $\mathcal{G}$. Each agent has a state x_i with initial value $x_i(0)$ and together they wish to determine the average of the initial states $\text{Ave}(x_0) = 1/N \sum x_0^i$.†

RMM: Ave not defined

The agents implement the following consensus algorithm:

$$x_{k+1}^i = \epsilon \sum_{j \in \mathcal{N}_i} (x_k^j - x_k^i) = -\epsilon |\mathcal{N}_i| (x_k^i - \text{Ave}(x_k^{\mathcal{N}_i}))$$

which is equivalent to the dynamical system

$$x_{k+1} == -\epsilon L x_k.$$

Proposition 7.10. *If the graph is connected, there exists an ϵ such that the state of the agents converges to $x_i^* = Ave(x_0)$ exponentially fast.*

- Proposition 1 implies that the spectra of L controls the stability (and convergence) of the consensus protocol.
- To (partially) prove this theorem, we need to show that the eigenvalues of L are all positive.

$$\dot{x} = -Lx \qquad L = \Delta - A$$

Note first that the subspaced spanned by $\mathbf{1} = (1, 1, \ldots, 1)^T$ is an invariant subspace since $L \cdot \mathbf{1} = 0$ Assume that there are no other eigenvectors with eigenvalue 0. Hence it suffices to look at the convergence on the complementary subspace $\mathbf{1}^\perp$.

Let δ be the disagreement vector

$$\delta = x - \text{Ave}(x(0))\,\mathbf{1}$$

and take the square of the norm of δ as a Lyapunov function candidate, i.e. define

$$V(\delta) = \|\delta\|^2 = \delta^T \delta \tag{7.7}$$

Differentiating $V(\delta)$ along the solution of $\dot{\delta} = -L\delta$, we obtain

$$\dot{V}(\delta) = -2\delta^T L \delta < 0, \quad \forall \delta \neq 0, \tag{7.8}$$

where we have used the fact that $\mathcal{G}$ is connected and hence has only 1 zero eigenvalue (along $\mathbf{1}$). Thus, $\delta = 0$ is globally asymptotically stable and $\delta \to 0$ as $t \to +\infty$, i.e. $x^* = \lim_{t \to +\infty} x(t) = \alpha_0 \mathbf{1}$ because $\alpha(t) = \alpha_0 = \text{Ave}(x(0)), \forall t > 0$. In other words, the average–consensus is globally asymptotically achieved. □

For an undirected graph with Laplacian L, the rate of convergence for the consensus protocol is bounded by the second smallest eigenvalue λ_2

Corollary 7.10.1. *Consider a network of integrators with a directed information flow $\mathcal{G}$ and vertices that apply the consensus protocol. Then, $\alpha = Ave(x)$ is an invariant quantity if and only if $\mathcal{G}$ is balanced.*

Remarks.

- Balanced graphs generalized undirected graphs and retain many key properties

Consensus on Directed Graphs

Talk through the case where the graph is directed. This includes balanced graphs, for which we recover the directed case, but also non-balanced graphs, where we reach a consensus but the value is not the average. Can also talk here about using different link weights, though can't talk about the effect on rate since that is not covered until the next section. **RMM**

Consensus over Communication Channels

The plan for this subsection is to talk about modifications to the basic consensus algorithm that take into account packet losses, rate limits and delays. Need to look through the literature to make sure we get the right basic results here to be useful. **RMM**

Consensus for Idempotent Functions

Look at extensions of consensus for computing min, max and other idempotent functions. Can also talk about what happens when we get join/rejoin actions, ala Charpentier and Chandy, though this might go better in a later section. **RMM**

7.3 Effects of Information Topology

This section will cover some of the effects of the information topology on the consensus problem. Need to think of a better title, though. **RMM**

Outline:

- Fixed graphs—rates of convergence (λ_2)
- Nearest neighbor graphs
- Gossip algorithms
- Eigenvalues of the Laplacian (including small word, scale free, etc)

7.4 Applications of Consensus

This section will cover some of the applications of consensus algorithms. **RMM**

Outline:

- Distributed computation (Tsitsiklis, PageRank [Ishi and Tempo])
- Flocking
- Load balancing
- Intrusion detection

7.5 Further Reading

Exercises