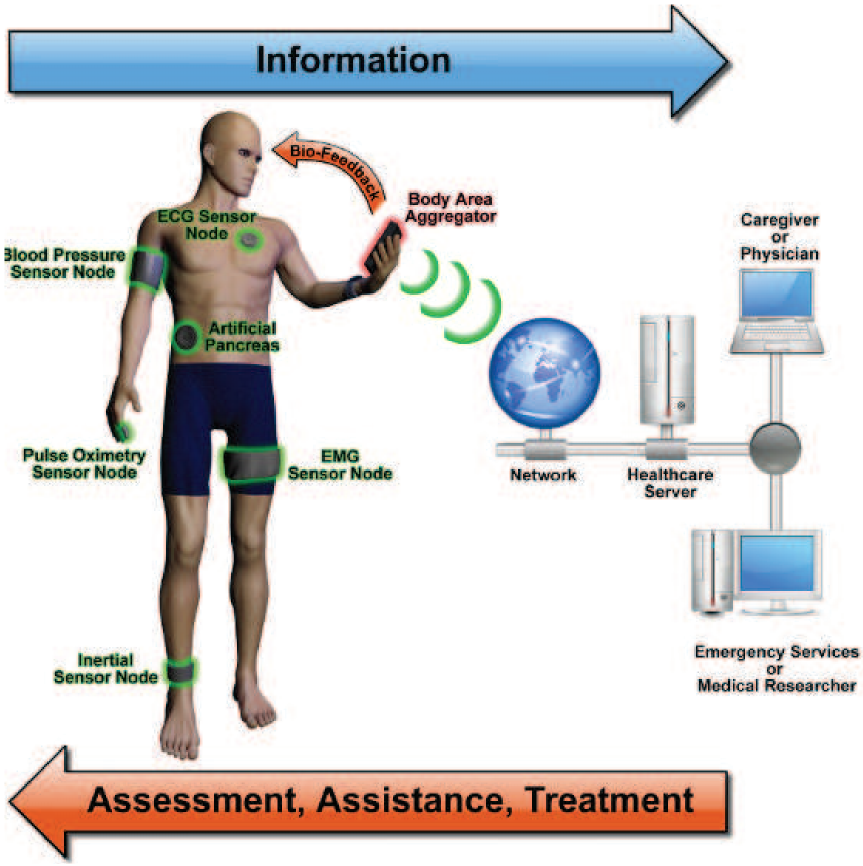
ROYAL INSTITUTE
OF TECHNOLOGY

EL 2745

Principles of Wireless Sensor Networks

Euhanna Ghadimi, Yuzhe Xu, and Carlo Fischione
January 2013



Automatic Control Lab, School of Electrical Engineering
KTH-Royal Institute of Technology, Stockholm, Sweden

Preface

The present compendium has been developed by Euhanna Ghadimi, Yuzhe Xu, and Carlo Fischione during 2011 and 2012 for the course EL 2745 Principles of Wireless Sensor Networks, given at KTH Royal Institute of Technology, Stockholm. In many cases exercises have been borrowed from other sources. In these cases, the original source has been cited.

CONTENTS

Exercises

1 Introductory Exercises: Sensor Modeling, Random Variables and Optimization Theory

EXERCISE 1.1 Gaussian Q function

(a) Consider a random variable X having a Gaussian distribution with zero mean and unit variance. The probability that X is larger than x, or distribution function, is

$$\mathbf{P}(X > x) = Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt\,,$$

where $Q(\cdot)$ is called the Q function. Plot the distribution function in the variable x. Recalling that a function is convex when the second derivative is strictly positive, find a region of x in which the function is convex.

(b) Consider a Gaussian random variable $X \sim \mathcal{N}(\mu, \sigma)$ of average μ and standard deviation σ. Such a random variable has a distribution function given by a translated and reshaped Q function:

$$Q\left(\frac{x-\mu}{\sigma}\right)\,.$$

Discuss about convexity region of this function.

(c) A function f is log-concave if $f(x) > 0$ and for all x in its domain $-\log f(x)$ is convex. Show that the twice differentiable function Q is log-concave.

EXERCISE 1.2 Binary hypothesis testing: application of the Q function

Assume a couple of sensor nodes are randomly deployed in a region of interest and are connected to a sink. The task of each sensor is to detect if an event happened or not, namely taking a binary decision. Each sensor measures noisy signals from the environment and whenever the measured signal is strong enough the sensor will decide that an event has occurred. We assume that the measurement noises at sensor i are identically and independently distributed (i.i.d) and follows a Gaussian distribution $n_i \sim \mathcal{N}(0, 1)$. The binary hypothesis testing problem for sensor i is as follows:

$$H1 : s_i = a_i + n_i$$

$$H0 : s_i = n_i$$

where s_i is the measured signal at sensor i, and $a_i \in \mathbb{R}_+$ is the signal amplitude associated to the event. Assume that all sensors use a common threshold τ to detect the event, i.e., if the measured signal at sensor i is larger than τ, then the sensor will decide that the event happened and will report this decision to the sink.

(a) Characterize the probability of *false alarm* p_f, namely the probability that a local sensor decides that there was an event while there was not one.

(b) Characterize the probability of *detecting* an event p_d, namely the probability that an event occurs and the sensor detects it correctly.

EXERCISE 1.3 Miscellanea of discrete random variables (Ex. 3.24 in [1])

Let X be a real-valued random variable that takes discrete values in $\{a_1, a_2 \dots, a_n\}$ where $a_1 < a_2 < \dots < a_n$, with probability $\mathbf{P}(X = a_i) = p_i$, $\forall i = 1, 2, \dots, n$. Characterize each of following functions of $\mathbf{p} = [p_i]$ $\{\mathbf{p} \in \mathbb{R}^n_+ | \mathbf{1}^T\mathbf{p} = 1\}$ (where $\mathbf{1}$ is the all ones vector) and determine wether the function is convex or concave.

(a) Expectation: $\mathbf{E}X$.

(b) Distribution function: $\mathbf{P}(X \geq \alpha)$.

(c) Probability of interval: $\mathbf{P}(\alpha \leq X \leq \beta)$.

(d) Negative entropy distribution: $\sum_{i=1}^n p_i \log p_i$.

(e) Variance: $\mathbf{var}X = \mathbf{E}(X - \mathbf{E}X)^2$.

(f) Quartile: $\mathbf{quartile}(X) = \inf\{\beta \,|\, \mathbf{P}(X \leq \beta) \geq 0.5\}$.

EXERCISE 1.4 Amplitude quantization

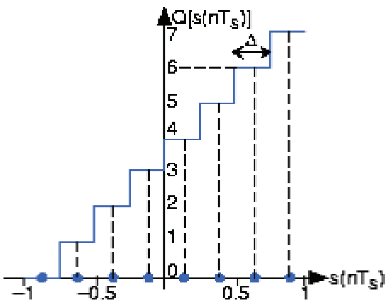
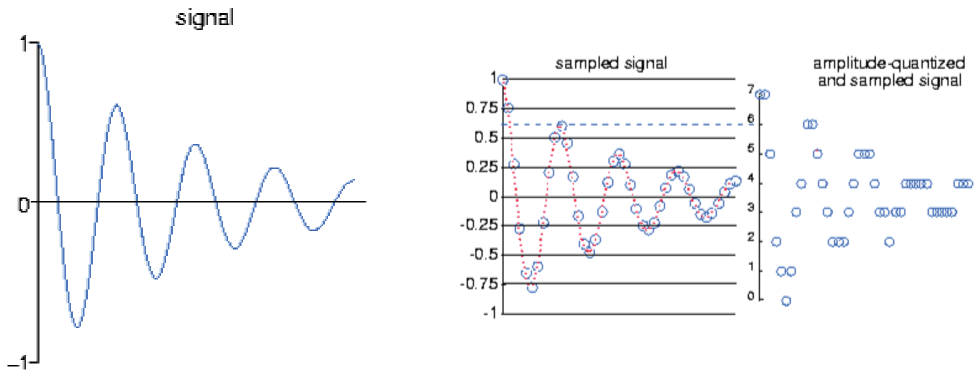


Figure 1.4.1: (a)A three-bit Analog to Digital (A/D) converter assigns voltage in the range $[-1, 1]$ to one of eight integers between 0 and 7. For example, all inputs having values lying between 0.5 and 0.75 are assigned the integer value six and, upon conversion back to an analog value, they all become 0.625. The width of a single quantization interval Δ is $2/2^B$.

The analog-to-digital (A/D) conversion is a standard operation performed in sensors and many electronic devices. Itworks as follows: Consider a sensor that samples a bandlimited continuos time signal $s(t)$. According to sampling theory, if the sensor samples the signal fast enough at time nT_s, where n is the sample number and T_s is the sampling time, it can be recovered without error from its samples $s(nT_s)$, $n \in \{\dots, -1, 0, 1, \dots\}$. The processing of the data further requires that the sensor samples be quantized: analog values are converted into digital form. The computational round-off prevents signal amplitudes from being converted with no errors into a binary number representation.

In general, in A/D conversion, the signal is assumed to lie within a predefined range. Assuming we can scale the signal without affecting the information it expresses, we will define this range to be $[-1, 1]$. Furthermore, the A/D converter assigns amplitude values in this range to a set of integers. A B-bit converter produces one of the integers $\{0, 1, \dots, 2^B - 1\}$ for each sampled input. Figure 1.4.1 shows how a three-bit A/D converter assigns input values to the integers. We define a quantization interval to be the range of values assigned to the same integer. Thus, for our example three-bit A/D converter, the quantization interval Δ is 0.25; in general, it is $2/2^B$.

Since values lying anywhere within a quantization interval are assigned the same value for processing, the original amplitude value is recovered with errors. The D/A converter, which is the device that converts integers to amplitudes, assigns an amplitude equal to the value lying halfway in the quantization interval. The integer 6 would be assigned to the amplitude 0.625 in this scheme. The error introduced by converting a signal from analog to digital form by sampling and amplitude quantization then back again would be half the quantization interval for each amplitude value. Thus, the so-called A/D error equals half the width of a quantization interval: $1/2^B$. As we have fixed the input-amplitude range, the more bits available in the A/D converter, the smaller the quantization error.



(a) (b)

Figure 1.4.2: (a) Shows a signal going through the analog-to-digital, where B is the number of bits used in the A/D conversion process (3 in the case depicted here). First it is sampled (b), then amplitude-quantized to three bits. Note how the sampled signal waveform becomes distorted after amplitude quantization. For example the two signal values between 0.5 and 0.75 become 0.625. This distortion is irreversible; it can be reduced (but not eliminated) by using more bits in the A/D converter.

To analyze the amplitude quantization error more deeply, we need to compute the signal-to-noise ratio, which is the ratio of the signal power and the quantization error power. Assuming the signal is a sinusoid, the signal power is the square of the root mean square (*rms*) amplitude: $\mathrm{power(s)} = (1/\sqrt{2})^2 = 1/2$. Figure 1.4.2 shows the details of a single quantization interval.

Its width is Δ and the quantization error is denoted by ϵ. To find the power in the quantization error, we note that no matter into which quantization interval the signal's value falls, the error will have the same characteristics. To calculate the *rms* value, we must square the error and average it over the interval.

$$\mathrm{rms}(\epsilon) = \sqrt{\frac{1}{\Delta}\int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \epsilon^2 \, \mathrm{d}\epsilon} = \left(\frac{\Delta^2}{12}\right)^{1/2}$$

Since the quantization interval width for a B-bit converter equals $2/2^B = 2^{1-B}$, we find that the signal-to-noise ratio for the analog-to-digital conversion process equals

$$\mathrm{SNR} = \frac{\frac{1}{2}}{\frac{2^{2(1-\mathrm{B})}}{12}} = \frac{3}{2}2^{2\mathrm{B}} = 6\mathrm{B} + 10\log 1.5\mathrm{dB}$$

Thus, every bit increase in the A/D converter yields a 6 dB increase in the signal-to-noise ratio. The constant term $10\log 1.5$ equals 1.76.

(a) This derivation assumed the signal's amplitude lay in the range $[-1, 1]$. What would the amplitude quantization signal-to-noise ratio be if it lay in the range $[-A, A]$?

(b) How many bits would be required in the A/D converter to ensure that the maximum amplitude quantization error was less than 60 db smaller than the signal's peak value?

(c) Music on a CD is stored to 16-bit accuracy. To what signal-to-noise ratio does this correspond?

EXERCISE 1.5 Accelerometer system design and system scale estimate (Ex.4.1 in [2])

An accelerometer is a sensor that measures acceleration. Consider the design of an accelerometer that is intended to meet specific acceleration sensitivity goals over a specific bandwidth given a position sensor sensitivity. The designer may adjust mass, spring constant, proof mass value, and resonance quality factor to achieve these goals.

(a) Consider an accelerometer with an electronic displacement sensor having a position sensitivity of $1\mathrm{pm}/(\mathrm{Hz})^{1/2}$. For a target acceleration sensitivity of $10^{-5}\ \mathrm{m/s^2}/(\mathrm{Hz})^{1/2}$ in the bandwidth from 0.001 to 100 Hz, find the largest sensor resonance frequency that may meet this objective while ignoring the effect of thermal noise.

(b) Now, include the effect of thermal noise and compute the required proof mass value for this accelerometer for Q values of 1, 100, and 10^4 (consider parameters $K_b = 1.38 \times 10^{-23}$ and $T = 300$).

(c) If this mass were to be composed of a planar Si structure, of thickness 1μ, what would be the required area of this structure.

EXERCISE 1.6 Signal dependent temperature coefficients (Ex.4.4 in [2])

A silicon pressure microsensor system employs a piezoresistive strain sensor for diaphragm deflection having a sensitivity to displacement of $\alpha = 1\mathrm{V}/\mu$ (at $T = 300$K). Further, this displacement is related to pressure with a pressure-dependent deflection of K = $0.01\mu/\mathrm{N/m^2}$. This is followed by an amplifier having a gain $G = 10$ (at $T = 300K$). This amplifier further shows an input-referred offset potential, $V_{\mathrm{offset}} = 0$ at 300K. Each of these characteristics include temperature coefficients. These temperature coefficients are listed here:

α	$10^{-2}/\mathrm{K}$
K	$10^{-4}/\mathrm{K}$
G	$-10^{-3}/\mathrm{K}$
V_{offset}	$-10\mu\mathrm{V/K}$

(a) Consider that the pressure sensor is exposed to no pressure difference. Find an expression for its output signal for temperature. Compute the temperature coefficient that describes the operation.

(b) Consider that the pressure sensor is exposed to a pressure difference signal of 0.1 N/m^2. Find an expression for its output signal for temperature and plot this. Estimate the temperature coefficient that describes its operation at the specific temperatures in the neighborhood of 250K and 350K.

(c) Consider that the pressure sensor is exposed to a pressure difference signal of 10 N/m^2. Find an expression for its output signal for temperature and plot this. Estimate the temperature coefficient that describes its operation at the specific temperatures in the neighborhood of 250K and 350K.

2 Programming Wireless Sensor Networks

EXERCISE 2.1 Hello world

Implement a Hello world program in TinyOS. Implement a timer and toggle the blue LED every 2 sec.

EXERCISE 2.2 Counter

Implement a counter using 3 LEDs. Use binary code to count-up every 1 seconds. Change the application to reset after it reaches 7.

EXERCISE 2.3 Ping Pong

(a) Develop an application where two sensor nodes start to exchange a message in a ping pong manner. For this task you are not allowed to use Node IDs. (hint: probably you need to use broadcast message once. then upon receiving the message use unicast to ping pong message between sender and receiver.)

(b) Change the application such that only two nodes out of many nodes can ping pong. (hint: you might use a sequence number inside the packet!)

EXERCISE 2.4 Dissemination Protocol

(a) The task is propagating a command in the sensor network. The command could be toggling a LED. Node ID 1 every 10 second sends a command to turn ON/OFF a selected LEDs. Receivers act accordingly and re-broadcast the command.

(b) How to avoid redundant commands? (hint: use a sequence counter to detect duplicate commands).

3 Wireless Channel

EXERCISE 3.1 The noisy sensor (Ex.14.6 in [2])

Sensor nodes are laid out on a square grid of spacing d as reported in Figure 3.1.1. Propagation losses go as the second power of distance. The source to be detected has a Gaussian distribution with zero mean and variance σ_n^2. The source is measured at each sensor by a noisy measurement having an independent Additive White Gaussian Noise (AWGN) with variance σ_S^2 . Sensor node 1 is malfunctioning, producing noise variance $10\sigma_n^2$. The two best nodes in terms of SNR cooperate to provide estimates of the source.

(a) Sketch the region of source locations over which node (1) will be among the two best nodes, assuming a long sequence of measurements are made of the source.

(b) For a single measurement, approximate the likelihood that a source at position $(0.25d, 0)$ will result in better SNR at sensor 5 than at sensor 1.

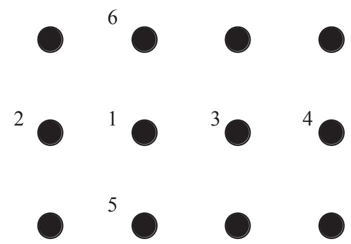


Figure 3.1.1: A sensor network.

EXERCISE 3.2 Radio power optimization

Consider the following model describing the required energy $E(A, B)$ to send a packet from node A to node B: $E(A, B) = d(A, B)^\alpha$. Here, $d(A, B)$ is the distance between node A and B and α is a system parameter with $\alpha > 2$. Assume that we are allowed to place a number of equidistant relay nodes between source node S and destination node T. Here, relay nodes serve as intermediate nodes to route packets from S to T. For instance, if S and T would use relay nodes A and B, the message would be sent from S to A, from A to B and finally from B to T.

(a) What is the ideal number of relay nodes in order to send a message from S to T with minimum energy consumption?

(b) How much energy would be consumed in the optimal case of the previous item?

(c) Assume now an energy model which determines the energy required to send a message from A to B as $E(A, B) = d(A, B)^\alpha + c$, with $c > 0$. Argue why this energy model is more realistic.

(d) Prove under the modified energy model introduced in previous item that there exists an optimal number n of equidistant intermediate nodes between S and D that minimizes the overall energy consumption when using these intermediate nodes in order to route a packet from S to T. [Assume n as a continuous variable for simplicity].

(e) Derive a closed-form expression on how much energy will be consumed when using this optimal number n of relay nodes. [Assume n as a continuous variable for simplicity].

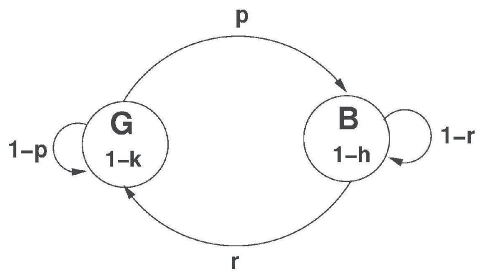


Figure 3.7.1: 2-state Markov chain describing to Gilbert Elliott model.

EXERCISE 3.3 Density of a Function of a Random Variable: the Rayleigh channel

Suppose that $\mathbf{x}$ has a chi-square distribution with the density

$$f(x) = \frac{1}{2^{n/2}\Gamma(n/2)} x^{n/2-1} e^{-x/2} U(x),$$

where

$$\Gamma(a+1) = \int_0^\infty x^a e^{-x} dx$$

is the gamma function and $U(x) = 1$ for $x \geq 0$ and $U(x) = 0$ otherwise. For a new random variable $\mathbf{y} = \sqrt{\mathbf{x}}$ compute its density function.

EXERCISE 3.4 Deriving the Density of a Function of a Random Variable: The step windowing

For a random variable $\mathbf{x}$ with density function f_x, compute the density function of $\mathbf{y} = \mathbf{x}U(\mathbf{x})$, where

$$U(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

EXERCISE 3.5 Deriving the Density of a Function of a Random Variable: The shadow fading

A log-normal distribution is a continuous probability distribution of a random variable whose logarithm has a Normal distribution. If $\mathbf{x}$ is a random variable with a normal distribution, then $\mathbf{y} = \exp(\mathbf{x})$ has a log-normal distribution. For $\mathbf{x} \sim \mathcal{N}(\mu, \sigma)$, compute the density function of $\mathbf{y} = \exp(\mathbf{x})$.

EXERCISE 3.6 Mean and Variance of Log-normal Distribution

For $\mathbf{x} \sim \mathcal{N}(\mu, \sigma)$, compute mean and variance of $\mathbf{y} = \exp(\mathbf{x})$.

EXERCISE 3.7 Gillbert-Elliott Model for Wireless Channels

The Gillbert-Elliott model is a 2-state Markov chain to model the wireless channel behavior when sending packet losses. This model consists of two channel states denoted as Good and Bad with corresponding error probabilities. In Fig. 3.7.1 each state may introduce errors for independent events with state dependent error rates $1 - k$ in the good and $1 - h$ in the bad state. In our framework, we interpret the event as the arrival of a packet and an error as a packet loss.

(a) Based on the given error rates and transition probabilities p and r, formulate π_G and π_B to be the stationary state probabilities of being in each state.

(b) Obtain error rate p_E in stationary state.

(c) Consider the Average Error Length (AEL) and Average number of Packet Drops (APD) as two statistics of channel. Derive π_G and π_B.

EXERCISE 3.8 Gillbert-Elliott model application

We have two sensor nodes that share a wireless channel. The state of the channel follows the Gillbert-Elliott model. Suppose that the transition probabilities in Fig. 3.7.1 are $p = 10^{-5}$ and $r = 10^{-1}$.

(a) Find the average length of an error burst.

(b) Obtain the average length of an error-free sequence of message transmission.

(c) Assume that the error probability in Good and Bad states is negligible and almost sure, respectively. Compute the average message loss rate of the channel.

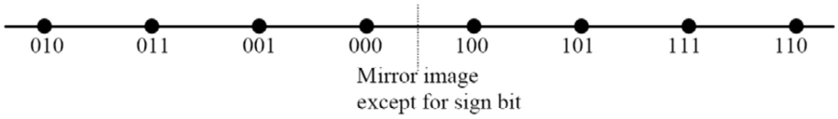


Figure 4.1.1: Gray-coded 8-PAM.

4 Physical Layer

EXERCISE 4.1 Gray Code (Ex. 6.5 in [2])

The property of the Gray code is that the code-words of adjacent symbols only differ in one bit. For example, the code words of 8-**PAM** (pulse amplitude modulation) symbols are as illustrated in Figure 4.1.1. This results in a minimum expected number of bit errors per symbol error, in conditions of low symbol error probability. Devise a Gray code for 16-**QAM** (quadrature amplitude modulation) symbols.

EXERCISE 4.2 Network reconfiguration (Ex.14.7 in [2])

A rectangular grid is also used for relaying packets. For the electronics used, it costs two times the energy of a hop among nearest neighbors (separated by distance d) to hop diagonally across the square (e.g. node 2 to 5) and eight times the energy to go a distance of 2d in one hop (e.g. node 2 to 3). In normal operation, packet dropping rates are negligible and routes that use the least energy are chosen.

(a) Considering only energy consumption, at what packet dropping rate is it better to consider using two diagonal hops to move around a malfunctioning node?

(b) Now suppose delay constraints are such that we can only tolerate the probability of needing three transmission attempts being less than 0.01. In this case, what error rate is acceptable, assuming packing dropping events are independent?

EXERCISE 4.3 Bit error probability for BPSK over AWGN channels

Compute the probability of error for binary phase shift keying (**BPSK**) with Additive white Gaussian noise (AWGN) channel model.

EXERCISE 4.4 Bit error probability for QPSK over AWGN channels

Compute the probability of error for Quadrature phase-shift keying (**QPSK**) modulation with Additive white Gaussian noise (AWGN) channel model.

EXERCISE 4.5 Error probability for 4-PAM over AWGN channels

Compute the probability of error for Pulse amplitude modulation (PAM) with Additive white Gaussian noise (AWGN) channel model.

EXERCISE 4.6 Average error probability for Rayleigh fading

Compute the average probability of error for a Rayleigh fading channel given the error probability of AWGN channel model.

EXERCISE 4.7 Detection in a Rayleigh fading channel

In a Rayleigh fading channel the detection of symbol x from y is based on the sign of the real sufficient statistic

$$r = |h|x + z,$$

where $z \sim \mathcal{N}(0, N_0/2)$. It means that, If the transmitted symbol is $x = \pm a$, then, for a given value of h, the error probability of detecting x is

$$\mathbf{Q}\left(\frac{a|h|}{\sqrt{N_0/2}}\right) = \mathbf{Q}\left(\sqrt{2|h|^2\text{SNR}}\right),$$

where SNR $= a^2/N_0$ is the average received signal-to-noise ratio per symbol time (note that we normalized the channel gain such that $\mathbf{E}[|\mathbf{h}|^\mathbf{2}] = \mathbf{1}$.) For Rayleigh fading when $|h|$ has Rayleigh distribution with mean 0 and variance 1, calculate the average probability of error. Approximate the solution for high SNR regions.

EXERCISE 4.8 Average error probability for log-normal fading

Consider a log-normal wireless channel with AWGN receiver noise. We know that the probability of error in AWGN is

$$\mathbf{Q}(\gamma) = \Pr\{\mathbf{x} > \gamma\} = \int_\gamma^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$$

The average probability of error with respect to the log-normal distribution is the average of $\mathbf{Q}(\gamma)$ with respect to the log-normal distribution. It is difficult to compute because $\mathbf{Q}$ is highly non linear. Suppose to perform a Stirling approximation of the $\mathbf{Q}$ function, which is

$$\mathbf{E}\left\{f(\theta)\right\} \sim \frac{2}{3}f(\mu) + \frac{1}{6}f(\mu + \sqrt{3}\sigma) + \frac{1}{6}f(\mu - \sqrt{3}\sigma)$$

where $f(\theta)$ is any function of a random variable θ having mean μ and variance σ^2. Compute the average probability of error of log-normal channel by using the Stirling approximation.

EXERCISE 4.9 Probability of error at the message level

In a WSN communication platform, consider a Rayleigh Channel over a AWGN receiver noise. The message is a frame of size f bits and is composed of the preamble, network payload, and a CRC code.

(a) Compute p the probability that the message is correctly received.

(b) Assume that the received signal level at the receiver decays inversely with the squared of the distance, i.e.,

$$\text{SNR} \approx \frac{\alpha E_b}{N_0 d^2}.$$

For messages of size 10 bits and the values $E_b/N_0 = 100$ and $\alpha = 0.1$, compute the farthest distance to deploy a receiver such that the probability of successfully message reception is at least $p = 0.9^{10} \approx 0.35$.

EXERCISE 4.10 The rate $2/3$ parity check code

A parity check code forms a modulo 2 sum of information bits and then appends this bit as the parity check. Consider, e.g., a scheme in which there are two information bits and one party bit. The codewords are then the set $000, 011, 101,$ and 101, which have even parity, while the odd-parity possibilities $001, 010, 100$ and 111 are excluded. There are four code-words and thus two bits of information, compared with the eight uncoded possibilities which would take three bits to represent. The code rate is thus $2/3$. Suppose this coding scheme is used in conjunction with binary phase shift keying (**BPSK**). Compute the coding gain assuming **ML** soft coding.

EXERCISE 4.11 Hard vs. soft decisions (Ex. 6.7 in [2])

In previous exercise, if hard-decisions are used instead of soft-decisions, answer to the following questions:

(a) How many errors can the code detect and correct, respectively?

(b) Compute the error probability, i.e., the probability that the decoder cannot make the correct decision.

(c) Compare the error probability with that resulting from soft-decisions.

5 Medium Access Control

EXERCISE 5.1 Slotted Aloha

In this exercise we analyze the Slotted Aloha when the number of stations n is not exactly known. In each time slot each station transmits with probability p. The probability that the slot can be used (i.e. the probability that exactly one station transmits) is

$$\Pr(\text{success}) = n \cdot p(1-p)^{n-1}\,.$$

If n is fixed, we can maximize the above expression and get the optimal p. Now assume that the only thing we know about n is $A \leq n \leq B$, with A and B being two known constants.

(a) What is the value of p that maximizes $\Pr(\text{success})$ for the worst $n \in [A, B]$?

(b) What is this "worst case optimal" value for p if $A = 100$ and $B = 200$?

EXERCISE 5.2 ARQ Ex.8.10 in [2]

Consider a simple ARQ scheme through a single transmission link of data rate R. The ARQ scheme works as follows. The sender transmits a data packet across the link. Once the receiver receives the whole packet, it checks if data have been corrupted. If there is no error, a packet is sent to the sender to acknowledge the correct reception of the data packet. If there is an error, an ARQ is sent for a retransmission. The sender resends the packet immediately after it receives the ARQ packet. Assume the lengths of data and ARQ packets are L and L_{ARQ} respectively, and the propagation delay along the link is t_d . Neglect the turn-around time at the sender and the receiver. Suppose that the probability the data packet is corrupted during transmission is P_e and ARQ packets are always correctly received.

(a) Determine the average number of transmissions required for a packet to be correctly received.

(b) Find the average delay a packet experiences. The delay is defined as the time interval between the start of the first packet transmission and the end of the correct packet reception, and note that it does not include the transmission of the last acknowledgement packet.

EXERCISE 5.3 Analysis of CSMA based MAC in WSNs

In this exercise we evaluate the performance of slotted CSMA protocol with fixed contention window size. Such mechanism is supported by protocols such as IEEE 802.15.4 in non-beacon enabled mode.

Assume a network of N sensors with a single channel and all the nodes are in the communication range of each other. The nodes use slotted CSMA scheme with fixed contention size M. Nodes sense the channel and if the channel is free they enter to the contention round. In contention round each node draws a random slot number in $[1, M]$ using uniform distribution and sets its counter with this integer number. In successive slots times t_{slot} each contender counts down until when its counter expires then it senses the channel and if there is no transmission in the channel it will send the packet immediately at beginning of the next slot. Assume t_d is the required time to transmit the data packet. t_{slot} is determined by physical layer parameters like propagation time of the packet (it also called vulnerable time) which is defined by the distance between the nodes. In this exercise t_{data} depends on data length is assumed to be much larger than t_{slot}. Each contention round will finish by a packet transmission that might be either successful or collided. Collision happens if at least two nodes draw the same minimum slot number, otherwise the transmission would be successful.

(a) Define P_s as the probability of having a successful transmission after a contention round with M maximum window size and N contenders. Also denote $p_s(m)$ as the probability of success at slot m. Find $p_s(m)$ and P_s.

(b) Similarly denote P_c as the probability of collision after contention round and $p_c(m)$ as the probability of collision at slot m. Propose an analytical model to calculate P_c and $p_c(m)$. Note that based on our system model, a collision happens at slot m, if at least two sensors pick the same slot m to transmit given that nobody has selected a smaller slot.

EXERCISE 5.4 MAC optimization for distributed estimation

Consider N nodes randomly deployed in a field. A node periodically checks with period S if there is an event of interest. Whenever node k detects such an event x_k, it starts broadcasting a monitoring message $m_k(x_k)$, which is called "state vector", to a fusion center. Nodes uses the slotted Alhoa medium access control protocol and transmit over the same wireless channel. In particular, each node transmits $m_k(x_k)$ in a randomly chosen time slot within the range $[1, S]$ units where S is the total number of slots per second. The node transmits a message within the slot boundaries following any slot. Hence, each node starts transmission with probability

$$\tau = \frac{z}{S}, \qquad 0 < \tau < 1,$$

where z is the rate of state vector transmissions per second. The probability that a node does not start transmission is $1 - \tau$. Collision at the fusion center happens when two nodes simultaneously transmit in a time slot.

The state vector transmission interval is $T_u = 1/z$. Each node wants to minimize the state vector transmission interval so to have often and more reliable information about the event of interest. However, this increases the collision probability.

(a) Pose an optimization problem which copes with such a tradeoff and argue if it is a convex one.

(b) Calculate the optimal rate of state vector transmissions per second that minimizes T_u.

EXERCISE 5.5 Broadcast

Three students discuss the broadcasting problem with collision detection in graphs of constant diameter. Student A claims that there is a deterministic protocol that allows to broadcast messages of length l in time $O(l)$. He says that it is possible since all nodes act synchronously and can detect collisions, which allows to transmit information one bit per round (slot) using the collision detection mechanism, i.e. detecting a transmission or collision in a slot means bit 1, detecting a free channel means 0. Student B says that this is not possible because he can prove the existence of a lower bound of $\Omega(\log n)$ for deterministic algorithms, which can be much larger than the length of a message l in general. He says that this can be done in the same way as for the lower bound of n for the deterministic broadcast without collision detection for graphs of diameter 2, i.e. using golden and blue nodes in the middle layer. Student C claims that A's idea works in principle but all nodes need to know the length l of the message. Who is right?

(a) If you believe A is right, give an algorithm that performs the broadcast.

(b) If you believe B is right, give a proof.

(c) If you believe C is right, describe an algorithm given that all nodes know the message length l and explain why the message length l is needed.

EXERCISE 5.6 $M/M/1$ queues (Ex.8.9 in [2])

Consider the infinite length $M/M/1$ queue.

(a) Given that the probability of n customers in the queue is $p(n) = (1-\rho)\rho^n$, where $\rho = \lambda/\mu$, show that the average number of customers in the queue is

$$N = \mathbf{E}(n) = \sum_{n=0}^{\infty} np(n) = \frac{\rho}{1-\rho}$$

.

(b) Plot N as a function of ρ when $0 \leq \rho < 1$. What happens when $\rho \geq 1$?

(c) Find the average delay that customers experience and the average waiting time that customers spend in queue. (Hint: use Little's theorem.)

6 Routing

EXERCISE 6.1 Shortest path routing: Bellman-Ford algorithm (Ex.8.4 in [2])

The network topology of Figure 6.1.1 is used to illustrate the Bellman-Ford algorithm for finding the shortest route to a node. In the figure, the number beside a node serves as the label for the node, and the number near an arc indicates the length of the arc. For instance, the arc connecting nodes 1 and 2 has length 1. Define d_{ij} to be the length of the direct arc connecting nodes i and j . If there is no direct arc connecting the two nodes, we set $d_{ij} = \infty$. By doing this, d_{ij} has meaning for any pair of nodes i and j in the network.

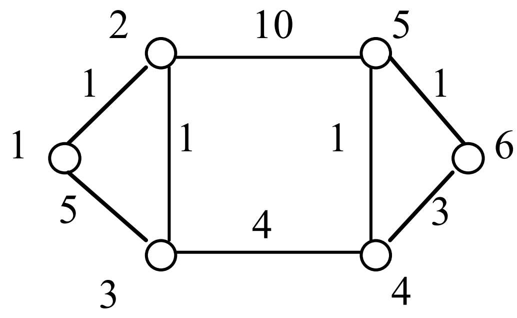


Figure 6.1.1: A simple network.

Consider node 1 as the destination node. The shortest path from node i to node 1 that traverses at most h arcs and goes through node 1 only once is called a shortest ($\leq h$) walk, and its length is denoted by D_i^h. Note there are two special cases. If all paths between node i and 1 consist of more then h arcs, $D_i^h = \infty$. By convention, $D_1^h = 0$ for any h.

(a) Determine D_i^0 for $i = 1, 2, \ldots, 6$. Find d_{ij} for all possible $i, j = 1, 2, \ldots, 6$.

(b) The following iteration is used to generate the subsequent shortest walks:

$$D_i^{h+1} = \min_j [d_{ij} + D_j^h] \quad \text{for all} i \neq 1$$

Determine D_i^1 for $i \neq 1$.

(c) Use the iteration equation in (b) to compute D_i^2 , D_i^3 ,... for $i \neq 1$. Stop the iteration when $D_i^{h+1} = D_i^h$, for all $i \neq 1$. The minimum distance from node i to node 1 is D_i^h in the last iteration.

EXERCISE 6.2 Shortest path routing: Dijkstra algorithm (Ex.8.5 in [2])

We will use Figure 6.1.1 to illustrate the Dijkstra algorithm for finding the shortest route to a destination node. The length of the direct arc connecting nodes i and j is defined to be d_{ij}. For a detailed description of the figure and the definition of d_{ij}, refer to previous exercise. Denote by P the set of nodes whose shortest path to the destination node is known, and denote by D_j the current shortest distance from node j to the destination node. Note that only when node j belongs to the set P can we say D_j is the true shortest distance. Choose node 1 as the destination node. Initially, set $P = \{1\}$, $D_1 = 0$, and $D_j = \infty$ for $j \neq 1$.

(a) Update D_j for $j \neq 1$ using the following equation

$$D_j = \min[D_j, d_{j1}] .$$

(b) Find i such that

$$D_i = \min_{j \notin P}[D_j]$$

update $P := P \cup \{i\}$.

(c) Update D_j for $j \notin P$ by the following equation

$$D_j := \min[D_j, D_i + d_{ji}]$$

in which i is the i obtained in (b).

(d) Go back and compute steps (b) and (c) recursively until P contains all the nodes in the network. The resulting D_j is the shortest distance from node j to node 1.

EXERCISE 6.3 Shortest path routing in WSNs

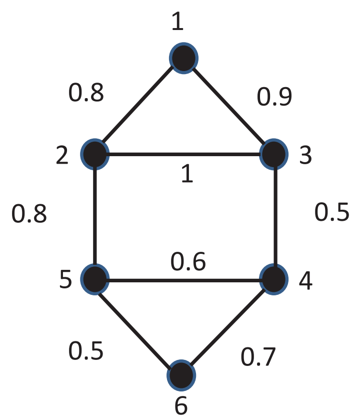


Figure 6.3.1: A sample WSN topology. Node 1 is the sink and link qualities (PRR) are depicted on each arc

One way of building routing tree in WSNs is based on ETX. ETX stands for expected number of transmissions. The Idea is to make a minimum spanning tree (MST) minimizing the expected number of transmissions for each node. This is done based on MAC layer functionalities (e.g., PRR). With PRR for each link between (i, j) nodes have a good estimate of packet reception rate from other party and hence can measure the temporal reliability of the link. Note that PRR is directional and the rate of packet reception for links (i, j) and (j, i) can be different. Having the values of PRR of direct neighbors available at each node, in a recursive fashion nodes can build a routing tree that minimizes the expected number of transmissions to the sink.

(a) Develop a sketch of the algorithm and the required equations to build the routing tree based on ETX metric.

(b) Consider Figure 6.3.1 and assume the PRR is bidirectional (links are undirected) where the values of the PRR are given on the arcs. Find the MST based on ETX metric.

EXERCISE 6.4 Anycast routing over WSNs

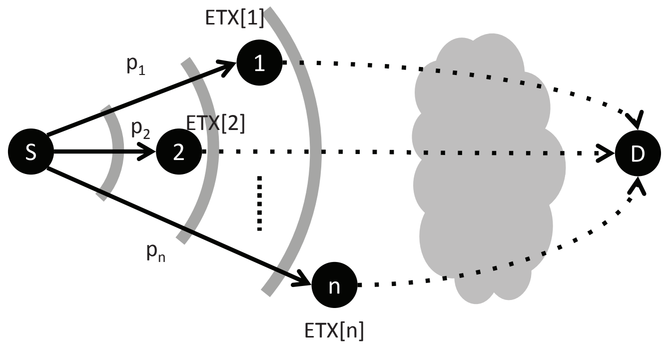


Figure 6.3.1: *Initial graph with link probabilities annotated. Each neighbor i of node s provides its ETX[i] to s.*

In WSNs, the expected number of transmissions of a node (ETX) is a routing metric, namely a metric used by a node to take the decision over which path the node routes packets. Denote by ETX$[s]$ the expected number of transmissions required for node s to send a packet to the destination D. Let $\mathcal{N}_s$, $\mathcal{P}_s$ and p_i be the neighbors set of s, parent set of s and probability of successful transmission from node s to neighboring node i, respectively. Given ETX$[i]$ and p_i for all $i \in \mathcal{N}_s$, ETX at s is defined as

$$\text{ETX}[s] = \min_{i \in \mathcal{N}_s} \left\{ \text{ETX}[i] + \frac{1}{p_i} \right\}$$

and the parent set of s is defined as $\mathcal{P}_s = \{i\}$,where i is the neighbor that minimizes ETX$[s]$ above. Note that the $\mathcal{P}_s$ has one component.

Now we want to extend this scheme to consider multiple parents. Figure 5.6.1 illustrates such network. The routing scenario is as follows. Node s looks at its parents set $\mathcal{P}_s = \{1 \dots n\}$ as an ordered set. It broadcasts a packet to all the parents and waits for an acknowledgement (ack) packet. If parent 1 receives the packet (with probability p_1) then node 1 will forward the packet to D (with cost ETX[1]). Now if node 1 fails to receive the packet and node 2 receives it, then node 2 will forward it. So within this scheme node i is allowed to forward a packet if 1) it successfully receives the packet from s with probability p_i and 2) if all the nodes with higher priority $1, \dots, i-1$ fail to get the packet. Assume that an efficient message passing scheme handles this structure.

(a) Calculate the new ETX metric for s and a given ordered set of parents $\mathcal{P}_s = \{1 \dots n\}$. [hint: first you can calculate the probability that a packet from s is received by at least one of the parents. Then, conditioned on that you are in one of the parents (the first hop transmission is successful), calculate the average ETX from one of the parents to the destination.]

(b) In Figure 5.6.1, assume that s has 3 neighbors with success probabilities $(p_1, p_2, p_3) = (1/2, 1/3, 1)$ and ETX of $(2, 2, 4)$, respectively. Calculate the ETX$[s]$ for two cases: with single parent and three parents with priority order $(1, 2, 3)$.

(c) For the second case of the previous point, find the optimal parent set (note that there are $2^3 - 1$ possible parent sets) that minimizes ETX$[s]$.

Exercise 6.5 Spanning tree (Ex.8.7 in [2])

Find all possible spanning trees for the two graphs in Figure 6.4.1 subject to the constraint that node 1 must be the root. Determine the number of nodes N and arcs A in each of these spanning trees. Can you see a relation between N and A?

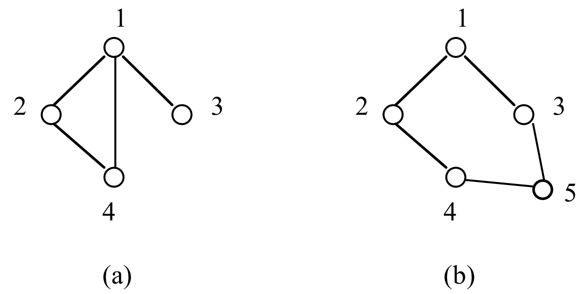


Figure 6.4.1: Spanning tree.

EXERCISE 6.6 Directed diffusion (Ex.8.8 in [2])

Consider the situation in Figure 6.6.1. The solid lines represent transmission links between nodes, and dashed lines indicate boundaries of tiers. Here node A wants to transmit to node D. Suppose the transmission takes the branches within the same tier with one third of the probability of branches in the next tier, and the packets do not back-track. Determine the likelihood of packets flowing through node B and C to reach D.

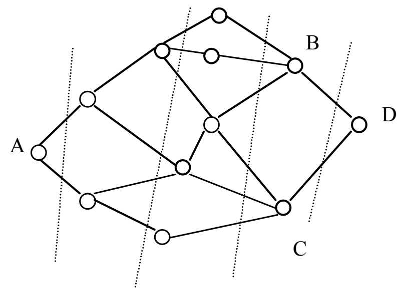


Figure 6.6.1: Directed diffusion.

7 Detection

EXERCISE 7.1 Binary choice in Gaussian noise

A signal voltage z can be zero (hypothesis H_0) or k (hypothesis H_1), each hypothesis with a probability $1/2$, The voltage measurement is perturbed by additive white Gaussian noise (AWGN) of variance σ^2. Compute the decision threshold for MAP criterion, and the error probabilities $\Pr(D_1|H_0)$ and $\Pr(D_0|H_1)$, where D_1 means that H_1 was decided, and D_0 means H_0 was decided.

EXERCISE 7.2 Binary hypothesis test and SNR (Ex.5.2 in [2])

Consider the binary choice in Gaussian noise, as shown in previous exercise with the threshold of $k/2$, the SNR is also maximized at the decision point. Since the possible signal values are known, the maximization of SNR means that the hypothesized noise power $E[n^2(t)]$ is minimized when the decision boundaries are optimally chosen. Prove that SNR is maximized when the threshold is $k/2$.

EXERCISE 7.3 MAP and the LRT (Ex.5.4 in [2])

Show that the MAP decision rule is equivalent to the likelihood ratio test.

EXERCISE 7.4 Binary decisions with unequal a priori probabilities(Ex.5.5 in [2])

For the binary choice in Gaussian noise in Exercise 1, compute the threshold when the probabilities of H_0 and H_1 are 1/3 and 2/3 respectively.

EXERCISE 7.5 Detection of known mean in Gaussian noise (Example D.1 in [6])

The simplest possible problem is to decide whether there is a known mean A in an observed signal or not:

$$H_0 : y_k = e_k \,,$$
$$H_1 : y_k = s_k + e_k \,.$$

Suppose to detect a general known signal s_k observed with Gaussian noise as $y_k = s_k + e_k$. Using a matched filter defined as

$$\bar{y} = \frac{1}{N}\sum_{k=0}^{N-1} y_k s_k = A + \bar{e}\,,$$

show that

$$A = \frac{1}{N}\sum_{k=0}^{N-1} s_k^2$$

and $\bar{e} \sim \mathcal{N}(0, \sigma^2/N)$. Here we assume that $\sum s_k = 1$.

EXERCISE 7.6 Fault detection

Suppose to detect a signal s_k observed with Gaussian noise as $y_k = s_k + e_k$, where $e_k \sim \mathcal{N}(0, \sigma^2)$. Assume there exist fault alarms for the signal, that is, the alarms occur when the measurement of signal beyond the interval $[-3\sigma, 3\sigma]$. Here assume that s_k equals 0 with $p_0 = 0.9$, and $t = 3\sigma$ as fault with $p_t = 0.1$. Find the probability of the correct fault alarms.

EXERCISE 7.7 Optimal Data Fusion in Multiple Sensor Detection Systems [7]

Let consider a binary hypothesis problem with the following two hypotheses: H_0 signal is absent, H_1 signal is present. The priori probabilities of the two hypotheses are denoted by $\Pr(H_0) = P_0$ and $\Pr(H_1) = P_1$. Assume that there are n detectors and the observations at each detector are denoted by $y_i, i = 1, \ldots, n$. Furthermore, assume that the observations are statistically independent and that the conditional probability density function is denoted by $p(y_i|H_j), i = 1, \ldots, n$, while $j = 1, 2$. Each detector employs a decision rule $g_i(y_i)$ to make a decision $u_i, i = 1, \ldots, n$, where

$$u_i = \begin{cases} -1 & \text{if} \quad \mathrm{H}_0 \quad \text{is} \quad \text{declared} \\ +1 & \text{if} \quad \mathrm{H}_1 \quad \text{is} \quad \text{declared} \end{cases}$$

We denote the probabilities of the false alarm and miss of each detector by P_{F_i} and P_{M_i} respectively. After processing the observations locally, the decisions u_i are transmitted to the data fusion center. The data fusion center determines the overall decision for the system u based on the individual decisions, i.e., $u = f(u_1, \ldots, u_n)$.

1. Show that
$$\log \frac{\Pr(H_1|\mathbf{u})}{\Pr(H_0|\mathbf{u})} = \log \frac{P_1}{P_0} + \sum_{S_+} \log \frac{1 - P_{M_i}}{P_{F_i}} + \sum_{S_-} \log \frac{P_{M_i}}{1 - P_{F_i}} \,,$$
where S_+ is the set of all i such that $u_i = +1$ and S_- is the set of all i such that $u_i = -1$.

2. Find the optimum data fusion rule using likelihood ratio.

EXERCISE 7.8 Counting Rule [8]

Consider the same situation in Exercise 7. An alternative scheme would be that the fusion center counting the number of detections made by local sensors and then comparing it with a threshold T:

$$\Lambda = \sum_{S_+} u_i \underset{H_0}{\overset{H_1}{\gtrless}} T \,,$$

which is called "counting rule". Now assume that each sensor has the same $P_{F_i} = P_f$ and $P_{M_i} = P_m$, find the probability of false alarm P_F and detection P_D at the fusion center level.

EXERCISE 7.9 Matched filter and SNR (Ex.5.12 in [2])

Prove the matched filter maximizes the output SNR and compute the maximum output SNR as a function of the energy of the signal $s(t)$ and N_0.

EXERCISE 7.10 Binary hypothesis testing and mutual information (Ex.5.3 in [2])

Consider the binary choice in Gaussian noise, as shown in Exercise 1. When $k = 1$ and the variance of the Gaussian distribution is 1, show numerically that the mutual information is maximized when $\gamma = 0.5$.

8 Estimation

EXERCISE 8.1

Given a vector of random variables Y that is related to another vector of random variables X, describe briefly what is the best linear estimator of X if one observes an outcome of Y.

EXERCISE 8.2 MMSE estimator

In many situations, one has to estimate x from some noisy measurements y that are a linear function of x plus some noise. Let X be a vector of random variables having zero mean. Suppose that Y is a vector of random variables related to X such that if x is an outcome of X, then an outcome of Y is $y = Hx + v$, where H is a constant matrix and v is a zero mean Gaussian noise having covariance R_v, with v independent of X. Then, the MMSE estimate of X is given that $Y = y$ is

$$P^{-1}\hat{x} = HR_v^{-1}y$$

with error covariance

$$P^{-1} = R_X^{-1} + H^T R_v^{-1} H\,.$$

Now, consider a network of n sensors. Let X be a random variables observed by each sensor by the noisy measurement $y_i = H_i x + v_i$ and $i = 1, \ldots, n$, where all the noises are uncorrelated with each other and with X. Let the estimate based on all the measurement be $\hat{x}$ and let $\hat{x}_i$ the estimate based on only the measurement y_i. Then,

$$P^{-1}\hat{x} = \sum_{i=1}^{n} P_i^{-1}\hat{x}_i$$

where P is the estimate error covariance corresponding to $\hat{x}$ and P_i is the estimate error covariance corresponding to $\hat{x}_i$, with

$$P^{-1} = \sum_{i=1}^{n} P_i^{-1} - (n-1)R_X^{-1}\,.$$

The above estimators, by the assumption that H_i is the i-th row of the matrix H, give the same estimate. Assume that R_X and R_v are diagonal matrixes. Motivate wether the first estimator requires more computations than the second estimator and suggest which one is best for a sensor network.

EXERCISE 8.3 Mean square (MS) estimation (Ex.5.21 in [2])

Let X be a real-valued RV with a pdf of $f_X(x)$. Find an estimate $\hat{x}$ such that the mean square error of x by $\hat{x}$ is minimized when no observation is available.

EXERCISE 8.4 Distributed MMSE estimator

We would like to estimate a vector of unknown constant parameters $x \in \mathbb{R}^{\mathrm{m}}$ using a network of n distributed sensors. Each sensor makes a noisy measurement

$$y_i = H_i x + v_i \qquad i = 1, \ldots, n.$$

Where H_i is an known matrix relating the unknown parameter to the measurement, v_i is a Gaussian noise with zero average and covariance matrix R_{v_i}. Moreover v_i's are assumed statistically independent noises. In vector notation one can formulate $y = Hx + v$, where y, H and v are $n \times 1$ vectors of y_i, H_i and v_i. Show that the maximum likelihood (or MMSE) estimate of x given y is

$$\hat{x} = \left(\sum_{i=1}^{n} H_i^T R_{v_i}^{-1} H_i \right)^{-1} \sum_{i=1}^{n} H_i^T R_{v_i}^{-1} y_i \, .$$

EXERCISE 8.5 Cramér-Rao bound (Ex.5.27 in [2])

Let $\overline{X}$ be the sample mean from n independent Gaussian random variables $X_1, X_2, \ldots, X_n$ with Gaussian distribution $N(\theta, \sigma^2)$. Assume σ^2 is known. First, derive the Cramér-Rao bound. Then, show that $\overline{X}$ is the most efficient unbiased estimate for θ (i.e. it attains the right-hand-side of the Cramér-Rao bound.)

EXERCISE 8.6 ML estimates of mean and variance of Gaussian random variables (Ex.5.28 in [2])

Consider n independent random samples from a Gaussian distribution $N(\mu, \sigma^2)$. Let $\theta = (\mu, \sigma)$, that is $\theta_1 = \mu$ and $\theta_2 = \sigma$. Find the Maximum-Likelihood (ML) estimates of μ and σ.

EXERCISE 8.7 Distributed detection/estimation

A set of N nodes is randomly deployed on a field. Every node makes observations on an unknown parameter $\theta \in [-1, 1]$. The observations are corrupted by an additive noise

$$x_k = \theta + v_k, \qquad k = 1, 2, \ldots, N \, ,$$

where v_k is the noise, which is modeled as a random variable. These random variables are assumed to be independent and identically distributed and with zero mean. In particular, they are uniformly distributed over $[-1, 1]$, with a probability distribution function (pdf)

$$p(v) = \frac{1}{2}, \qquad \text{if } v \in [-1, 1] \, .$$

To get an accurate estimate of the parameter θ, each node reports its observations to a fusion centre. However, due to message losses and medium access control protocol, each node is allowed to transmit a message composed only by one bit. In other words, each node reports a message $m_k(x_k) \in \{0, 1\}$ to the fusion center. The bit of the message is chosen as

$$m_k(x_k) = \begin{cases} 1, & \text{if } x_k \geq 0 \\ 0, & \text{if } x_k < 0 \, . \end{cases}$$

(a) Find the expectation $\mathbf{E}(m_k)$ and variance of the one-bit message $\mathbf{E}(m_k - \mathbf{E}(m_k))^2$ for node k.

(b) Prove that $\mathbf{E}(m_k - \mathbf{E}(m_k))^2$ is bounded above. Find the upper bound.

(c) Suppose that the fusion center uses a final fusion function f and estimator $\hat{\theta}$ to decide upon the parameter given by

$$\hat{\theta} := f(m_1, \ldots, m_N) = \frac{2}{N} \sum_{k=1}^{N} m_k - 1 \, .$$

Find $\mathbf{E}(\hat{\theta})$ and $\mathbf{E}(\hat{\theta} - \theta)^2$.

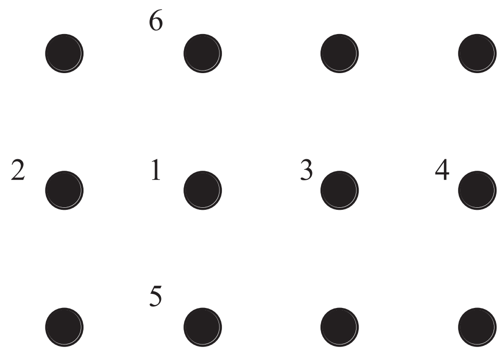


Figure 8.8.1: A grid of sensor nodes.

(d) Suppose we want the variance of estimate $\hat{\theta}$ less than ϵ. What is the minimum number of nodes N to deploy so that such a variance bound is satisfied?

EXERCISE 8.8 Distributed detection, MAC, and routing

Sensor nodes are laid out on a square grid of spacing d, as depicted in Figure 8.8.1. Ever sensor wants to detect a common source.

(a) Suppose that the source signal has a Gaussian distribution with zero mean and with variance σ_S^2. Moreover, every sensor measures such a signal with an additive Gaussian noise of zero average and variance σ_n^2. If the measured signal is positive, the sensor decides for hypothesis H_0, otherwise the sensor decides for hypothesis H_1. Based on the measured signal, characterize the probability of false alarm and the probability of miss detection per every sensor.

(b) Now, suppose that the source signal is constant and has a power S. Such a signal power is received at every sensor with an attenuation given by r_i^2, where r_i is the distance between the source and sensor i. Sensor node 1 is malfunctioning, producing noise variance $10\sigma_n^2$. The two best nodes in terms of SNR will cooperate to provide estimates of the source. Characterize the region of source locations over which node (1) will be among the two best nodes.

(c) The grid depicted in Figure 8.8.1 is also used for relaying. Assume it costs two times the energy of a hop among nearest neighbors (separated by distance d) to hop diagonally across the square (e.g. node 2 to 5) and eight times the energy to go a distance of 2d in one hop (e.g. node 2 to 3). Let p be the packet loss probability. Characterize p for which it is better to consider using two diagonal hops to move around the malfunctioning node.

(d) Under the same assumption of the previous item, suppose that there is an ARQ protocol, but the delay constraints are such that we can only tolerate three retransmission attempts. Let 0.99 be the probability of having up to three retransmissions. Assuming packing dropping events are independent, characterize the constraint that probability of packet losses per transmission should satifsy.

EXERCISE 8.9 Unknown mean in Gaussian noise (Example C.1 in [6])

Consider an unknown mean in Gaussian noise,

$$y_k = \theta + e_k, \quad e_k \in \mathcal{N}(0, \sigma^2)\,.$$

Find the mean and variance of the sample average. Show that the sample average is the minimum variance estimator.

Hint: use the CRLB.

EXERCISE 8.10 Moments method (Example C.9 and C.10 in [6])

The method of moments is general not efficient and thus inferior to the ML method. However, in many cases it is easier to derive and implement. For Gaussian mixtures, the MLE does not lead to analytical solutions so numerical algorithms have to applied directly to the definitions, where whole data vector has to be used. Using the method of moments, closed expressions can be derived as functions of reduced data statics.

The key idea is to estimate the first p moments of data, and match these to the analytical moments of the parametric distribution $p(y|\theta)$:

$$\begin{aligned}
\mu_i =& E[y_k^i] = g_i(\theta)\,, \quad i = 1, 2, \ldots, p \\
\hat{\mu}_i =& \frac{1}{N}\sum_{k=1}^{N} y_k^i\,, \quad i = 1, 2, \ldots, p \\
\mu =& g(\theta)\,, \\
\hat{\theta} =& g^{-1}(\hat{\mu})\,.
\end{aligned}$$

Now consider a Gaussian mixture

$$p(y|\theta) = \alpha\mathcal{N}(y; 0, \sigma_1^2) + (1-\alpha)\mathcal{N}(y; 0, \sigma_2^2),$$

where

$$\theta = (\alpha, \sigma_1^2, \sigma_2^2)^T\,.$$

Assume that those σ are known. Using the method of moments, find the estimation of α. If the variances are unknown, find the estimation of θ.

9 Positioning and Localization

EXERCISE 9.1 Timing Offset and GPS (Ex.9.1 in [2])

GPS uses a constellation of 24 satellites and their ground stations as reference points to calculate positions accurate to a matter of meters. Suppose we find our distance measurements from three satellites to be 18 000, 19 000, and 20 000 km respectively. Collectively this places the location at either of the two points where the 20 000 km sphere cuts through the circle that is the intersection of the 18 000 and 19 000 km spheres. Thus by ranging from three satellites we can narrow our position to just two points in space. To decide which one is our true location we could make a fourth measurement. However, usually one of the two points is a non-possible answer (either too far from Earth or moving at an impossible velocity) and can be rejected without a measurement. Now apply the above principle of location in a two-dimensional space. Assume that points A, B, and C are reference points with known locations, respectively at $(x1, y1)$, $(x2, y2)$, and $(x3, y3)$, and that the unknown position is 3.0 meters from point A, 4.0 meters from point B, and 5.0 meters from point C

(a) Suppose that accurate measurements are available. Then the three measurements can be used to uniquely determine the position. Let $(x1, y1) = (0, 3.0)$, $(x2, y2) = (4.0, 0)$, $(x3, y3) = (4.0, 3.0)$. Find the position.

(b) Now assume that all measurements include a single timing offset that corresponds to an error of 0.5 m. In other words, the position is observed to be 3.5 m from point A, 4.5 m from point B, and 5.5 m from point C. Develop a generic procedure to find the true position.

EXERCISE 9.2 Linearizing GPS Equations (Ex.9.2 in [2])

In order to find position using the GPS system, we need to know the location of at least three satellites and the distance to each of those satellites. Assume that the three satellites are located respectively at $(x1, y1, z1)$, $(x2, y2, z2)$, and $(x3, y3, z3)$, and that the distance between us and the three satellites are respectively $d1, d2, d3$. The following nonlinear system of equations needs to be solved,

$$\begin{aligned} (x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2 &= d_1^2 \\ (x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2 &= d_2^2 \\ (x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2 &= d_3^2 \end{aligned} \tag{9.2}$$

Obviously linearization is desirable in this case. Assume that the reference point is $(0, 0, 0)$. Prove that the resulting system after linearizing (9.2) is

$$2\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & y_3 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_1^2 + y_1^2 + z_1^2 - d_1^2 \\ x_2^2 + y_2^2 + z_2^2 - d_2^2 \\ x_3^2 + y_3^2 + z_3^2 - d_3^2 \end{bmatrix}$$

EXERCISE 9.3 Averaging to reduce error in TOA (Ex.9.3 in [2])

TOA is based upon the measurement of the arrival time of a signal transmitted from the to-be-located object to several reference nodes. For radio signals, the distance is ct, where c is the velocity of light and t is time of travel from the object to the reference node. This measurement thus indicates the object is on a circle of radius ct, centered at the reference node. There is always a need for at least three reference nodes to determine the location of the object correctly. The disadvantage of this technique is that processing delays and non-line-of-sight propagation can cause error, resulting in mistakes in the TOA estimation. Assume that t is a Gaussian distributed RV with mean at the real time of arrival $\bar{t}$ and a variance δ_t.

(a) Find the mean and variance of the resulting range of the object.

(b) Now assume that independent multiple measurements of range are available. That is, $t(n)$, $n = 1, 2, 3, ..$, is the measured time of arrival from the reference node to the to-be-located object, at time instant n. Show that multiple measurements help to reduce the error in the resulting range of the object.

EXERCISE 9.4 Weighted centroid computation (Ex.9.9 in [2])

Three beacons are located at $a = (1, 1)$, $b = (1, -1)$, and $c = (-1, 1)$. The received powers from nodes a, b, and c are 1.2, 1.5, and 1.7 respectively. Calculate the unknown position of the receiver through a weighted centroid computation.

EXERCISE 9.5 Collaborative multilateration

Consider Figure 9.5.1, suppose node U can estimate ranges only for nodes A, C, and V, and node V can estimate ranges only for nodes B, D, and U, where the unknown locations are U and V. One can begin with an initial guess at the position of U from either the centroids of the known positions in immediate range, or via the topology. Then multilateration is performed using the locations of all neighbors (estimated or known) to refine the positions, in a sequence that proceeds until locations stabilize. Compute the first estimate of the positions of $U(u_0)$ and $V(n_0)$ as the centroids of the nodes they can hear that have known position. Then iteratively calculate by multilateration the positions in the order u_1, n_1 assuming perfect range measurements.

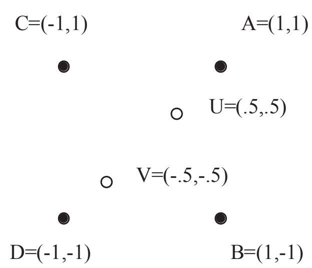


Figure 9.5.1: Four node multilateration.

EXERCISE 9.6 Linearization of angle of arrival (AOA) location determination (Ex.9.11 in [2])

The intersection of the angles from two or more sites may be used to provide an unknown location in the plane. For this triangulation problem, denote the position of the two known nodes as $r_i = [\ x_i \quad y_i\ ]^T, i = 1, 2$, and the unknown node's position as $r = [\ x \quad y\ ]^T$. The bearing angles can be expressed as

$$\theta_i = f_i(r, r_i) + n_i, \quad i = 1, 2, \tag{9.6}$$

where n_i is the angle measurement error, and the function $f_i()$ is defined as

$$f_i(r, r_i) = \arctan\left(\frac{x - x_i}{y - y_i}\right), i = 1, 2. \tag{9.6}$$

After collecting angle measurements from known nodes, the unknown node's position can be found by solving the nonlinear system of equations

$$\begin{aligned} \theta_1 &= \arctan\left(\frac{x-x_1}{y-y_1}\right) + n_1 \\ \theta_2 &= \arctan\left(\frac{x-x_2}{y-y_2}\right) + n_2 \end{aligned} \tag{9.6}$$

This triangulation problem can alternatively be solved by linearizing the $f_i()$ function by expanding it in a Taylor series around a reference point, denoted by r_0. Once the equation system is linearized, the ML estimator is used to provide the following unknown node position estimate

$$\begin{aligned} \hat{r} &= r_0 + \left(G^T N^{-1} G\right)^{-1} G^T N^{-1} \begin{bmatrix} \theta_1 - f_1(r_0) \\ \theta_2 - f_2(r_0) \end{bmatrix} \\ &= r_0 + G^{-1} \begin{bmatrix} \theta_1 - f_1(r_0) \\ \theta_2 - f_2(r_0) \end{bmatrix}. \end{aligned} \tag{9.6}$$

Matrix $N = E[nn']$ is the measurement error covariance matrix, and matrix G is the matrix of the resulting equation system after linearizing (9.6). Matrix G is equal to

$$G = \begin{bmatrix} (y_0 - y_1)/d_{01}^2 & -(x_0 - x_1)/d_{01}^2 \\ (y_0 - y_2)/d_{02}^2 & -(x_0 - x_2)/d_{02}^2 \end{bmatrix},$$

where angle $\theta_{0i} = f_i(r_0), i = 1, 2$, and d_{0i} is the distance between the i th node and r_0. Given $r_0 = [\ 0 \quad 0\ ]^T$, $r_1 = [\ -3 \quad 4\ ]^T, r_2 = [\ 4 \quad 3\ ]^T$, $\theta_1 = 45°, \theta_2 = 135°$, and

$$N = \begin{bmatrix} 1 & 0 \\ 0 & 0.9 \end{bmatrix}.$$

Use equation (9.6) to find the unknown node's position. Comment on the accuracy of the results.

10 Time Synchronization

EXERCISE 10.1 TOA with low-cost clocks (Ex.9.4 in [2])

In order to make accurate range measurements in a GPS system, the receiver and satellite both need clocks that can be synchronized down to the nanosecond, which potentially could require atomic clocks not only on all the satellites, but also in the receivers. However, atomic clocks are far too expensive for everyday consumer use. GPS sidesteps this problem by measuring the distance to four instead of the minimum three located satellites. Every satellite contains an expensive atomic clock, but the receiver uses an ordinary quartz clock, which it constantly resets. With four range measurements, the receiver can easily calculate the necessary adjustment that will cause the four spheres to intersect at one point. Based on this, it resets its clock to be in sync with the satellite's atomic clock, thus providing time as well as location. Explain mathematically how this fourth measurement provides these benefits.

EXERCISE 10.2 Time difference of arrival (TDOA) in a two-dimensional space (Ex.9.5 in [2])

TOA requires that all the reference nodes and the receiver have precise synchronized clocks and the transmitted signals be labeled with time stamps. TDOA measurements remove the requirement of an accurate clock at the receiver. Assume that five reference nodes have known positions $(0,0), (-1,-1), (0,1), (3,1)$, and $(1,4)$ respectively. We choose $(0,0)$ as the reference sensor for differential time-delays which are defined as

$$t_{1r} = t_1 - t_r = \frac{r_{s1} - r_{s2}}{v},$$

where v is the velocity of propagation, r_{si} is the distance between the unknown node and the ith node. Further assume that $t_{12} = -1.4s$, $t_{13} = 0.4s$, $t_{14} = -1.6s$, and $t_{15} = -2.6s$.

(a) Find the unknown location (x_t, y_t).

(b) Now assume that the propagation speed is known as 1.8 m/s. Find the unknown location (x_t, y_t).

EXERCISE 10.3 TDOA in a three-dimensional space (Ex.9.6 in [2])

Now assume that five reference nodes are known at $(0,3,0)$, $(6,0,0)$, $(3,4,0)$, $(-4,-3,0)$, and $(0,0,-8)$ respectively. Also, $t_{12} = 0s$, $t_{13} = 1s$, $t_{14} = 0.7s$, $t_{15} = 0.7s$, and $t_{16} = 1.7s$. The velocity of propagation is v.

(a) Find the unknown location (x_t, y_t, z_t) using (9.10) from lecture notes.

(b) Now assume that the propagation speed is known to be 8.7 m/s. Find the unknown location (x_t, y_t, z_t) using (9.12) from lecture notes.

EXERCISE 10.4 Ex.9.3 in [3]

Consider two nodes, where the current time at node A is 1100 and the current time at node B is 1000. Node A's clock progresses by 1.01 time units once every 1 s and node B's clock progresses by 0.99 time units once every 1 s. Explain the terms clock offset, clock rate, and clock skew using this concrete example. Are these clocks fast or slow and why?

EXERCISE 10.5 Ex.9.4 in [3]

Assume that two nodes have a maximum drift rate from the real time of 100 ppm each. Your goal is to synchronize their clocks such that their relative offset does not exceed 1 s. What is the necessary re-synchronization interval?

EXERCISE 10.6 Ex.9.6 in [3]

A network of five nodes is synchronized to an external reference time with maximum errors of 1, 3, 4, 1, and 2 time units, respectively. What is the maximum precision that can be obtained in this network?

EXERCISE 10.7 Ex.9.7 in [3]

Node A sends a synchronization request to node B at 3150 (on node A's clock). At 3250, node A receives the reply from node B with a times-tamp of 3120.

(a) What is node A's clock offset with respect to the time at node B (you can ignore any processing delays at either node)?

(b) Is node A's clock going too slow or too fast?

(c) How should node A adjust the clock?

EXERCISE 10.8 Ex.9.8 in [3]

Node A issues a synchronization request simultaneously to nodes B, C, and D(Figure 10.8.1). Assume that nodes B, C, and D are all perfectly synchronized to each other. Explain why the offsets between node A and the three other nodes may still differ?

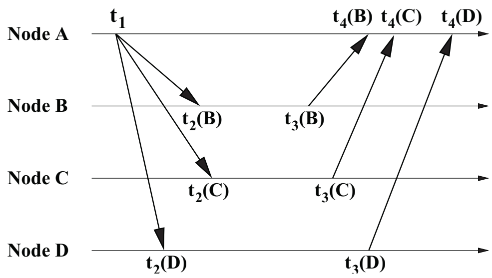


Figure 10.8.1: Pairwise synchronization with multiple neighboring nodes.

11 Networked Control Systems

EXERCISE 11.1 Matrix Exponential

Let A be an $n \times n$ real or complex matrix. The exponential of A, denoted by e^A or $\exp(A)$, is the $n \times n$ matrix. Find e^A using two different way, where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} .$$

EXERCISE 11.2 Stability

Given a bi-dimensional state space system

$$X_{t+1} = \Phi X_t ,$$

1. show how to compute the eigenvalues of Φ.
2. make some comments on the relationship between the eigenvalues of Φ and the stability.

EXERCISE 11.3 Modeling

Model the dynamics of a coordinated turn (circle movement) using Cartesian and polar velocity. Here we assume that the turn rate ω is piecewise constant.

EXERCISE 11.4 Linearized Discretization

In some cases, of which tracking with constant turn rate is one example, the state space model can be discretized exactly by solving sampling formula

$$x(t+T) = x(t) + \int_t^{t+T} a(x(\tau))d\tau ,$$

analytically. The solution can be written as

$$x(t+T) = f(x(t)) .$$

Using this method, discretize the models in Ex:11.3.

EXERCISE 11.5 Modeling of the Temperature Control

Assume that in winter, you'd like to keep the temperature in the room warm automatically by controlling a house heating system. Let T_i, T_o and T_r denote the temperature inside, outside and radiator. Thus the process model can be simplified as

$$\begin{aligned} \dot{T}_i =& \alpha_1(T_r - T_i) + \alpha_2(T_o - T_i) \\ \dot{T}_r =& \alpha_3(u - T_r) . \end{aligned}$$

1. Model the dynamics in standard state space form. Here assume that the outside temperature is aroud zero, $T_o = 0$.

2. Assume that the sampling time is h, model the continuous state space form to the discrete time standard form.

EXERCISE 11.6 PID Controller

One heuristic tuning method for PID controller is formally known as the Ziegler-Nichols method. In this method, the K_i and K_d gains are first set to zero. The K_p gain is increased until it reaches the ultimate gain, G, at which the output of the loop starts to oscillate. G and the oscillation period T_G are used to set the gains, let $K_p = 0.60G$, $K_i = 2K_p/T_G$ and $K_d = K_pT_G/8$. Now consider the system in Ex: 11.3 and the step response plot shown in Fig. 11.6.1. Find T_G, then design the PID controller for the system in continuous space using Ziegler-Nichols method. Here assume that $G = 10$.

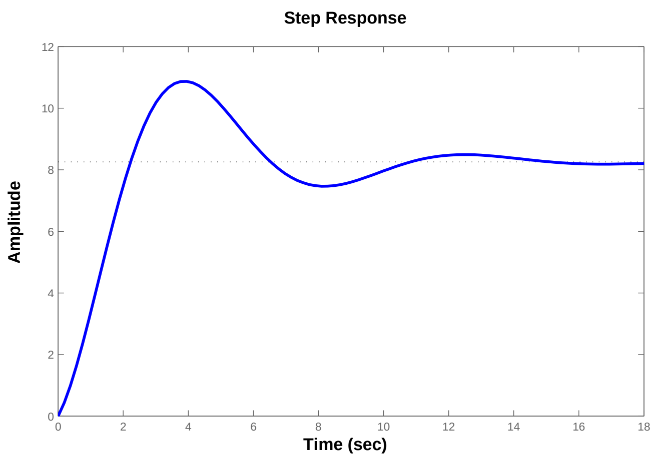


Figure 11.6.1: The step response for PID controller with $K_p = 12$.

EXERCISE 11.7 Stability of Networked Control Systems with Network-induced Delay.

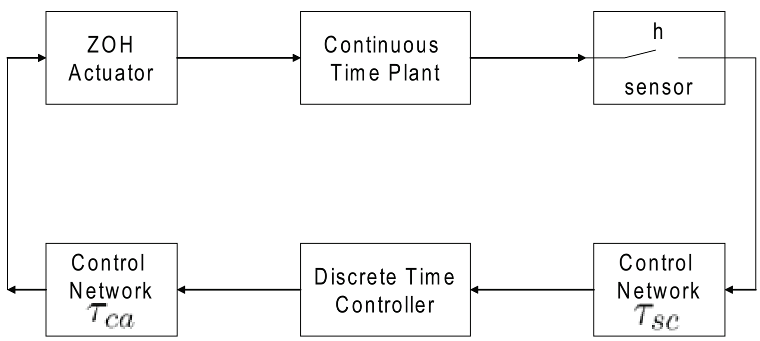


Figure 11.7.1: Networked Control System with communication delay.

Consider the Networked Control Systems (NCS) in Figure 11.7.1. The system consists of a continuous plant

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned},$$

and a discrete controller

$$u(kh) = -Kx(kh), \qquad k = 0, 1, 2, \ldots ,$$

where $A \in \mathbb{R}$, $B \in \mathbb{R}$, $C \in \mathbb{R}$.

Let $A = 0$, $B = I$. Illustrate the stability properties of the system as function of the network delays τ_{sc} and τ_{ca} under the assumptions that $\tau_{sc} + \tau_{ca} \leq h$ and that $h = 1/K$.

EXERCISE 11.8 Control with time-varying delay

A process with transfer function

$$P(z) = \frac{z}{z - 0.5}$$

is controlled by the PI-controller

$$C(z) = K_p + K_i \frac{z}{z - 1}$$

where $K_p = 0.2$ and $K_i = 0.1$. The control is performed over a wireless sensor network, as shown in Figure 11.10.1. Due to retransmission of dropped packets, the network induces time-varying delays. How large can the maximum delay be, so that the closed loop system is stable?

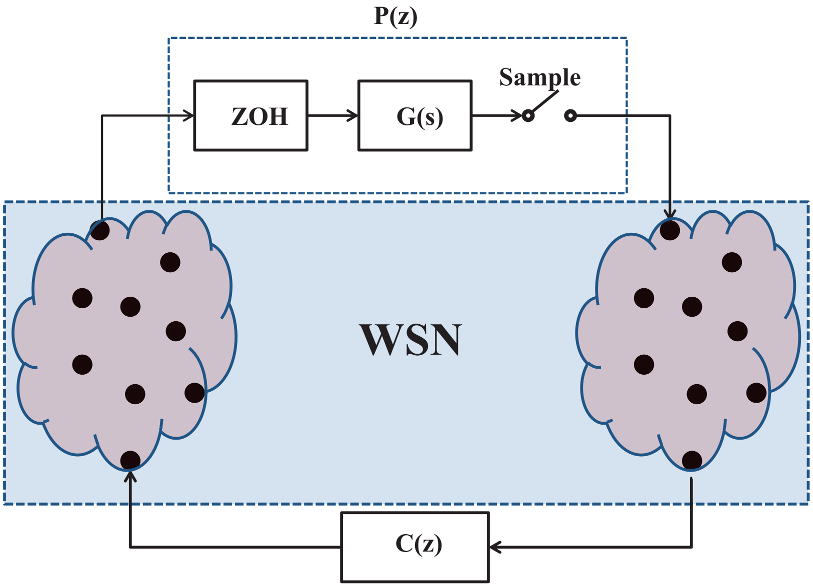


Figure 11.8.1: Closed loop system for Problem 11.2.

EXERCISE 11.9 Stability of Networked Control Systems with Packet Losses.

Consider the Networked Control System in Figure 11.9.1. It is assumed that the network is present only from the plant to the controller. The state space plant model is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

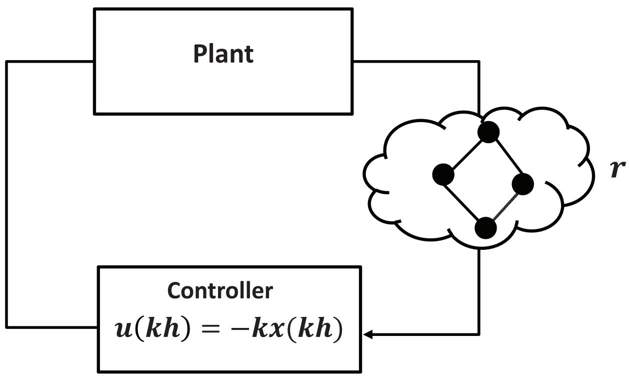


Figure 11.9.1: Networked Control System with packet losses.

The feedback controller is $u(kh) = -Kx(kh)$, where $K = [20, 9]$.

Suppose that packets sent over the network are received at rate $r = 1 - p$, where p is the packet loss rate, and that the system is sampled at rate $h = 0.3\,$s. What is the lower bound on reception rate r that still guarantee the stability of the system?

EXERCISE 11.10 Networked Control System

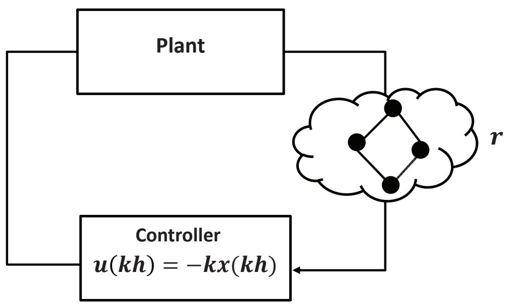


Figure 11.10.1: Closed loop system over a WSN.

Consider the Networked Control System (NCS) in Fig. 11.10.1. The system consists of a continuous plant

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{11.10a}$$
$$y(t) = Cx(t)\,, \tag{11.10b}$$

where $A = a$, $B = 1$, $C = 1$. The system is sampled with sampling time h, and the discrete controller is given by

$$u(kh) = -Kx(kh), \qquad k = 0, 1, 2, \ldots\,,$$

where K is a constant.

(a) Suppose that the sensor network has a medium access control and routing protocols that introduce a delay $\tau \leq h$. Derive a sampled system corresponding to Eq.(11.10) with a zero-order-hold.

(b) Under the same assumption above that the sensor network introduces a delay $\tau \leq h$, give an augmented state-space description of the closed loop system so to account for such a delay.

(c) Under the same assumption above that the sensor network introduces a delay $\tau \leq h$, characterize the conditions for which the closed loop system becomes unstable [Hint: no need of computing numbers, equations will be enough]

(d) Now, suppose that the network does not induce any delay, but unfortunately introduces packet losses with probability p. Let $r = 1 - p$ be the probability of successful packet reception. Give and discuss sufficient conditions for which the closed loop system is stable. If these conditions are not satisfied, discuss what can be done at the network level or at the controller level so to still ensure closed loop stability.

EXERCISE 11.11 Energy-Efficient Control of NCS over IEEE 802.15.4 Networks [5].

Consider the Networked control system over IEEE 802.15.4 network composed of 3 control loops depicted in the Figure 11.11, where each process is scalar of the form $\dot{x}_i = a_i x_i + b_i u_i, i = 1, 2, 3$, and where the

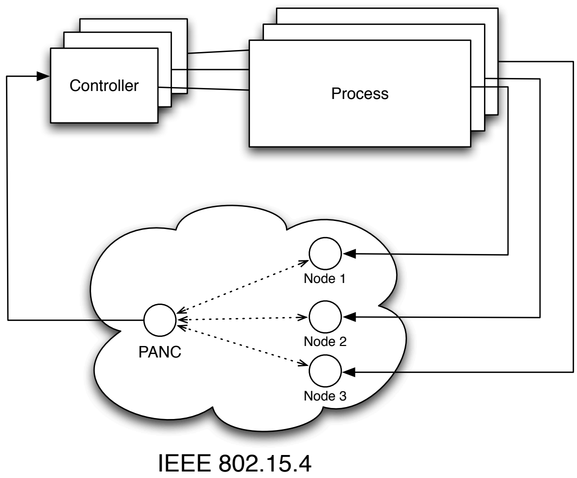


Figure 11.11.1: NCS over IEEE 802.15.4 network.

communication from the sensor nodes to the Personal Area Network Coordinator (PANC) is allowed only during the Guaranteed Time Slot (GTS) portion of the super-frame. Assume that there are no time delays, i.e. the transmissions from sensor i to the PANC and the respective control updates u_i are performed at the same instant $t = T_{i,k}$ and that each node can transmit only a packet per super-frame.

At each $t = T_{i,k}$, node i sends the values of $x_i(T_{i,k})$ and $t_{i,k+1}$ to the PANC, where $x_i(T_{i,k})$ is the measurement of the output of process i at time $t = T_{i,k}$, and $t_{i,k+1}$ is the time by which the next transmission from node i must be performed. The controller i updates the control input u_i with $u_i = -k_i x_i(T_{i,k})$ and it keeps it constant in the time interval $[T_{i,k}, T_{i,k+1})$. The transmissions are performed according to a self-triggered sampler that predicts the time in which the condition $|e_i(t)| := |x_i(T_{i,k}) - x_i(t)| \leq \delta_i$ is violated. The self-triggered sampler has the expression

$$t_{i,k+1} = T_{i,k} + \frac{1}{|a_i|} \ln \left(1 + \frac{|a_i|\delta_i}{|a_i - b_i k_i||x_i(T_{i,k})|} \right) . \tag{11.11}$$

Consider the numerical values of the described NCS as in the following table where $x_{i,0}$ denotes the initial condition of the process i. Determine:

(a) The values of k_1 such that practical-stability of the loop #1 is ensured.

(b) The values of δ_2 such that that practical-stability of the loop #2 is ensured.

	a_i	b_i	k_i	δ_i	$x_{i,0}$
Loop #1	2	1	?	$\frac{1}{2}$	5
Loop #2	3	-2	-2	?	8
Loop #3	2	$\frac{1}{2}$	6	$\frac{1}{2}$	?

(c) The values of $x_{3,0}$ such that that practical-stability of the loop #3 is ensured.

(d) For each control loop, find an upper-bound of the the practical-stability region size ε_i.

12 Scheduling

EXERCISE 12.1 Scheduling in Smart Grids

The latest wireless network, 3GPP Long Term Evolution (LTE), is used to transmit the measurements obtained by a smart grid. Every time slot, LTE need to allocate resources for N users in time- and frequency- domain. Assume that in each time slot, LTE has $N_{\text{TTI}} \times N_{\text{RB}}$ transmission resources, where N_{TTI} is the number in time domain and N_{RB} the number in frequency domain. Now, assume that each transmission resource can be used by one user only in one time slot. And the utility can be defined as $R_{i,j}^{(c)}$ for slot used by user c in time i and frequency j. Now pose the scheduler design by an optimization problem by maximize the sum of the utility of the transmission resources.

Since LTE transmits data flows not only in a smart grid but also for public use in a wide area, we would like choose a utility function in the scheduling problem so to give the highest value to messages from the devices in smart grid. Assume that the devices in smart grid have following features: constant data updating rates, equivalent data packets lengths, and approximately invariant channel qualities. Design a suitable utility function.

13 Security

EXERCISE 13.1 The Caesar code (Ex.14.1 in [2])

The Virginère family of codes for the alphabet is specified by

$$c_i = m_i + k_i \bmod 26\,,$$

where m_i is the numerical index of the letter in the alphabet (e.g., $B = 2$), and k_i is a component of a key of length d. When d goes to infinity and the key is chosen randomly, this is the Vernam code, otherwise known as the one-time pad. When d=1, this is one of the 26 possible Caesar ciphers. Typically it is broken by brute-force attacks. All possible keys are applied to the cipher text and the message that makes sense is selected. More precisely, a message that matches the statistical properties of the occurrences of letters and sequences of letters in a natural language is selected as being correct. Knowing that a Caesar code was employed, decode:

NBCMCMYUMS

EXERCISE 13.2 Unicity distance (Ex.14.2 in [2])

The length of text required to uniquely decode enciphered text when keys are randomly generated is known as the unicity distance. Denoting by $H(K)$ the entropy of the key and by D the redundancy of the text, then the unicity distance is defined as

$$\frac{H(K)}{D}\,.$$

There are $26!$ possible substitution ciphers for a 26-letter alphabet, while the redundancy of English expressed using $\log 10$ is 0.7. What length of text is sufficient to uniquely decode any substitution cipher? What does this say about the security of such ciphers?

EXERCISE 13.3 Euclid's Algorithm (Ex.14.3 in [2])

Euclid's algorithm can be used to find the greatest common divisor (gcd) of two integers a and b. Suppose $a > b$ and let $\{q_i\}$ and $\{r_i\}$ be integer quotients and remainders respectively. The algorithm rests on two principles. First, if b divides a, then $\mathrm{gcd(a,b)} = \mathrm{b}$. Second, if $a = qb + r$, then $\mathrm{gcd(a,b)} = \mathrm{gcd(b,r)}$. One can find $\mathrm{gcd(a,b)}$ by repeatedly applying the second relation:

$$\begin{aligned}
&a = q_0 b + r_1 \;\; 0 \le r_1 < |b| \\
&b = q_1 r_1 + r_2 \;\; 0 \le r_2 < r_1 \\
&r_1 = q_2 r_2 + r_3 \;\; 0 \le r_3 < r_2 \\
&\vdots \\
&r_k = q_{k+1} r_{k+1} + r_{k+2} \;\; 0 \le r_{k+2} < r_{k+1}
\end{aligned}\quad .$$

The process continues until one finds an integer n such that $rn + 1 = 0$, then $rn = \mathrm{gcd(a,b)}$. Find $\mathrm{gcd}(10480, 3920)$.

EXERCISE 13.4 Factoring products of primes (Ex.14.4 in [2])

Two ways to decode a public key system are to try out all possible keys or to attempt to factor $n = pq$ where p and q are both primes. Let $p(x)$ denote the number of primes less than or equal to x. We have

$$\lim_{x \to \infty} \frac{\pi(x)}{x/\ln(x)} = 1, \quad \text{hence } \pi(x) \approx \frac{x}{\ln x} \quad \text{for large x.}$$

Quadratic sieve factoring is the most efficient known algorithm for factoring primes less than about 110 decimal digits long (well within the range of keys in common use). Its complexity is sub exponential, in this case

$$O(e^{(\ln n)^{1/2}}(\ln \ln n)^{1/2})$$

For $n = 2^{64}$ and $n = 2^{128}$ determine the relative effort of running through the primes less than n or applying quadratic sieve factoring.

EXERCISE 13.5 Hash functions and sensing (Ex.14.5 in [2])

In message authentication codes, a family of hash functions h_k where k is the secret key, are employed. The properties of good families are:

(1) Ease of computation of $h_k(x)$, given k.

(2) Compression of an input x of arbitrary length into a sequence of fixed length, n.

(3) Even given many text-message-authentication-code pairs $(x_i, h_k(x_i))$ it is computationally infeasible to compute any text-message-authentication-code pair $(x, h_k(x))$ for a new input x not equal to x_i.

(a) What brute force attacks could result in determination of the hash function, and what is a simple countermeasure?

(b) How is hash function compression similar to and different from the mapping of data from the physical world into a decision by a sensor node?

EXERCISE 13.6 Fighting infection (Ex.14.8 in [2])

Suppose nodes face a constant probability p of being infected (e.g., malfunctioning, taken over) in every time epoch. A mobile agent can visit $1/100$ of the nodes in each epoch and fix/replace them if they are infected. It is further desired that less than $1/100$ of the data produced by nodes be corrupted. Determine the maximum tolerable infection probability if (a) nodes simply report their own data or (b) three nearby nodes with identical access to the data perform a majority vote.

EXERCISE 13.7 Information theft (Ex.14.9 in [2])

The compression approach in a sensor network has implications both for scalability and vulnerability to information theft. Consider the network depicted in Figure 13.7.1 Tier 1 produces 100 data per node; differing levels of compression are applied. Calculate the volume of data at risk when a node is compromised at any of the three levels with the following data compression schemes

(a) Tier 1 does no compression; tier 2 reduces the union of inputs to 10 units; tier 3 reduces union of inputs to 1 unit.

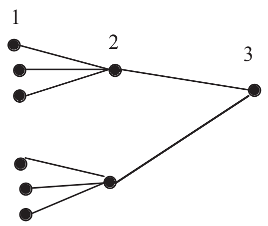


Figure 13.7.1: Three-tier network.

(b) Tier 1 compresses to 20 units through a hardwired algorithm; tier 2 to 5 units; tier 3 to 1 unit.

(c) Tiers 1 and 2 do no compression; tier 3 reduces to 1 unit.

EXERCISE 13.8 Physical security (Ex.14.10 in [2])

An important component in the overall security of a system is the difficulty of physical access for theft or tampering. A rule of thumb is that the electronic and physical security means should present comparable difficulty to would-be attackers, and should have cost commensurate with the value of what is being protected. Comment on the applicability/usefulness of the following systems for uses in ubiquitous small sensor nodes, nodes with cameras, gateways (including those in vehicles), aggregation points, CAs: tamper-detection devices, hardened casing, security bolts, security port for key updates, camouflage/embedding in structures, secured room, secured facility and redundant systems

EXERCISE 13.9 People and data reliability (Ex.14.11 in [2])

Data reliability will depend also on the users/applications just as security depends on human involvement. List the ways in which confidence (trust/reputation) is gained among people and comment on how reputation systems for people fit into the overall system integrity for a sensor network.

Solutions

1 Introductory exercises

SOLUTION 1.1

Random variables

(a) For a zero mean Gaussian distribution

$$\mathbf{P}(X > x) = Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \mathrm{dt}\,,$$

one can verify that the function is continuous in $x \in (-\infty, +\infty)$. Moreover, the first derivative is

$$Q_x^{'} = \frac{-1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\,,$$

and the second derivative is

$$Q_x^{''} = \frac{x}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \geq 0\,,$$

for $x \geq 0$. So the function is convex for $x \geq 0$.

(b) For the function

$$Q\left(\frac{x-\mu}{\sigma}\right) = \int_x^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(t-\mu)^2}{2\sigma^2}} \mathrm{dt}$$

A derivation similar to the previous exercise yields

$$Q^{''}\left(\frac{x-\mu}{\sigma}\right) = \frac{x-\mu}{\sigma^2\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \geq 0 \text{ for } x \geq \mu$$

(c) We prove that Q function in its entire domain is log-concave. According to the definition, a function f is log-concave if $f(x) > 0$ and for all x in its domain, log f is concave. Suppose f is twice differentiable entirely its domain $x \in \mathbb{R}$, so

$$\nabla^2 \log f(x) = \frac{f^{''}(x) \cdot f(x) - f^{'}(x)^2}{f^2(x)}$$

We conclude that f is log-concave if and only if $f^{''}(x) \cdot f(x) \leq f^{'}(x)^2$. For $Q(x)$, in previous exercise we computed the second derivative

$$f^{''}(x) = \frac{x}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} < 0$$

for $x < 0$. Moreover, $f(x) \geq 0$ for all x which indicates that the inequality holds for $x < 0$. What remains to show is that the inequality also holds for $x > 0$. For the case of Gaussian function the basic inequality reduces to

$$x\mathrm{e}^{-\frac{x^2}{2}} \cdot \int_x^{\infty} \mathrm{e}^{-\frac{t^2}{2}} \mathrm{dt} \leq \mathrm{e}^{-x^2}\,.$$

Hence

$$\int_x^{\infty} \mathrm{e}^{-\frac{t^2}{2}} \mathrm{dt} \leq \frac{\mathrm{e}^{-\frac{x^2}{2}}}{x}$$

To prove this inequality, remember the following general result for convex functions

$$g(t) \geq g(x) + g^{'}(x)(t-x)\,.$$

We now apply the above inequality for $g(t) = t^2/2$, so we have

$$\frac{t^2}{2} \geq \frac{x^2}{2} + x(t-x) = xt - \frac{x^2}{2}\,.$$

So, multiplying by -1 and taking exponential we will yield

$$\mathrm{e}^{-\frac{t^2}{2}} \leq \mathrm{e}^{-xt+\frac{x^2}{2}}\,.$$

Now take the integral and conclude

$$\int_x^\infty \mathrm{e}^{-\frac{t^2}{2}}\mathrm{dt} \leq \int_x^\infty \mathrm{e}^{-xt}\cdot \mathrm{e}^{\frac{x^2}{2}}\mathrm{dt} = \frac{\mathrm{e}^{-x^2}}{x}\cdot \mathrm{e}^{\frac{x^2}{2}} = \frac{\mathrm{e}^{-\frac{x^2}{2}}}{x}\,.$$

SOLUTION 1.2

Application of $Q(\cdot)$ function

Recalling the definition of p_f as the probability of *false alarm* and p_d as the probability of *detection*, we have

(a)

$$p_f = \mathbf{P}(s_i > \tau \,|\, \mathrm{H}_0) = \int_\tau^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}\mathrm{dt} = Q(\tau)$$

(b)

$$p_d = \mathbf{P}(s_i > \tau \,|\, \mathrm{H}_1) = \int_\tau^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{(t-a_i)^2}{2}}\mathrm{dt} = Q(\tau - a_i)$$

SOLUTION 1.3

(Ex. 3.24 in [1]) In each case we investigate the given function based on variable $\{\mathbf{p} \in \mathbb{R}^n_+ | \mathbf{1}^T\mathbf{p} = 1\}$.

(a) $\mathbf{E}x = a_1p_1 + a_2p_2 + \cdots + a_np_n$ is a linear function of $\mathbf{p}$, hence it is both convex and concave.

(b) $\mathbf{P}(X \geq \alpha)$. Let $j = \min\{a_i >= \alpha\}$, and then $\mathbf{P}(X \geq \alpha) = \sum_{i=j}^n p_i$, is a linear function of $\mathbf{p}$ and, hence is convex and concave.

(c) $\mathbf{P}(\alpha \leq X \leq \beta)$. Let $j = \min\{a_i >= \alpha\}$, and $k = \max\{i | a_i \leq \beta\}$. Then $\mathbf{P}(\alpha \leq X \leq \beta) = \sum_{i=j}^k p_i$, is a linear function of $\mathbf{p}$ and, hence is convex and concave.

(d) $\sum_{i=1}^n p_i \log p_i$. We know $\mathbf{p}\log\mathbf{p}$ is a convex of $\mathbf{p}$ on $\mathbb{R}_+$ (assuming $0\log 0 = 0$), so $\sum_i p_i \log p_i$ is convex. Note that the function is not concave. To check this we consider an example, where $n = 2$, $\mathbf{p} = [1, 0]$ and $\mathbf{p}^{'} = [0, 1]$. The function value at both points $\mathbf{p}$ and $\mathbf{p}^{'}$ is equal to 0. Now consider the convex combination $[0.5, 0.5]$ has function value $\log(1/2) < 0$. Indeed this is against concavity inequality.

(e) $\mathbf{var}X = \mathbf{E}(X - \mathbf{E}X)^2$. We have

$$\mathbf{var}\, X = \mathbf{E}X^2 - (\mathbf{E}X)^2 = \sum_{i=1}^n p_i a_i^2 - (\sum_{i=1}^n p_i a_i)^2,$$

So $\mathbf{var}\, X$ is a concave quadratic function of $\mathbf{p}$. The function is not convex. For example consider the case $n = 2$, $a_1 = 0$, $a_2 = 1$. Both X_1 with probabilities $[p_1, p_2] = [1/4, 3/4]$ and X_2 with $[p_1, p_2] = [3/4, 1/4]$ lie in probability simplex and we have $\mathbf{var}\, X_1 = \mathbf{var}\, X_2 = 3/16$. But the convex combination $X_3 = 1/2X_1 + 1/2X_2$ with probabilities $[p_1, p_2] = [1/2, 1/2]$ has a variance $\mathbf{var}\, X_3 = 1/4 > 3/16$. This contradicts convex inequality.

(f) **quartile**$(X) = \inf\{\beta \,|\, \mathbf{P}(X \leq \beta) \geq 0.5\}$. This function is piecewise constant, So it is not continuous. Therefore, **quartile**(X) is not convex nor concave. To see this, consider an example with $n = 2$ and probability simplex $p_1 + p_2 = 1$. We have

$$f(\mathbf{p}) = \begin{cases} a_2 & \text{for } p_1 \in [0, 1/2) \\ a_1 & \text{for } p_1 \in [1/2, 1] \end{cases}$$

SOLUTION 1.4

Amplitude Quantization

(a) The signal-to-noise ratio does not depend on the signal amplitude. With an A/D range of $[-A, A]$, the quantization interval $\Delta = 2A/2^B$ and the signal's rms value (again assuming it is a sinusoid) is $A/\sqrt{2}$.

(b) Solving $2^{-B} = .001$ results in $B = 10$ bits.

(c) A 16-bit A/D converter yields a SNR of $6 \times 16 + 10 \log 1.5 = 97.8$dB.

SOLUTION 1.5

Accelerometer system design and system scale estimate

(a)

$$10^{-5} = 10^{-12}/(1/\omega_0^2) \quad \text{for} \omega < \omega_o. \quad \text{Thus,} \quad \omega_0^2 = 10^7 \quad \text{and} \quad \omega_0 = 3.16 \times 10^3 \quad \text{rad/sec} = 504 \quad \text{Hz.}$$

(b)

$$\text{TNEA} \equiv \sqrt{\frac{4\text{k}_\text{b}\text{T}\omega_\text{O}}{\text{MQ}}} = \sqrt{\frac{4 \times 1.38 \times 10^{-23} \times 300 \times 3160}{\text{MQ}}} = 7.23 \times 10^{-9}\sqrt{\frac{1}{\text{MQ}}}$$

For TNEA = 10^{-5}, $\sqrt{\frac{1}{MQ}} = 1.38 \times 10^3; MQ = 5.2 \times 10^{-7}$ Thus, for $Q = 1$, $M = 0.52$ μkg$= 0.52$mg

for $Q = 100$, $M = 5.2 \times 10^{-9}$ kg$= 5.2\mu$g

for $Q = 10,000$, $M = 5.2 \times 10^{-11}$ kg$= 0.052\mu$g

(c) For $Q = 1$

$0.52\text{mg}/2.33 \quad \text{gm/cm}^3 = 2.23 \times 10^{-4} \quad \text{cm}^2$

At $t = 1\mu = 10^{-4} \quad \text{cm}$; A $= 2.23\text{cm}^2$.

SOLUTION 1.6

Signal dependent temperature coefficients

(a) $V_\text{offset} = 10\mu\text{V} \times 10^{-4}(300\text{K} - \text{T})$

$V_\text{out} = 0 + V_\text{offset} = 10\mu\text{V} \times 10^{-4}(300\text{K} - \text{T})$

Temperature coefficient is 10^{-4}/K

(b) $V_{\text{offset}} = 10\mu\text{V}(300\text{K} - \text{T})$

$\alpha = 1\text{V}/\mu(1 + 10^{-2}T)$

$K = 0.01\mu/\text{N}/\text{m}^2(1 + 10^{-4}\text{T})$

$G = 10(1 - 10^{-3}T)$

$V_{\text{out}} = P_{\text{in}}\alpha KG + GV_{\text{offset}}$

Now,

250K: -6.2×10^{-5}

350K: -8.3×10^{-5}

(c) $V_{\text{offset}} = 10\mu\text{V}(300\text{K} - \text{T})$

$\alpha = 1\text{V}/\mu(1 + 10^{-2}T)$

$K = 0.01\mu/\text{N}/\text{m}^2(1 + 10^{-4}\text{T})$

$G = 10(1 - 10^{-3}T)$

$V_{\text{out}} = P_{\text{in}}\alpha KG + GV_{\text{offset}}$

Now,

250K: -5.2×10^{-3}

350K: -7.3×10^{-3}

```
event void button_handler( button_state_t state ) {

    if ( state == BUTTON_PRESSED && sendflag == TRUE ) {
                    sendflag = FALSE;
                    // send the first packet with broadcast
                    msg->senderId = TOS_NODE_ID;
                    msg->counter = counter;
                    broadcast_message(&msg);//forward message
                    }//end_if
                }
```

(a) Broadcast

```
event message_t* Receive.receive(message_t* msg) {

          newmsg->senderId = TOS_NODE_ID;
          newmsg->counter = msg->counter+1;
          newmsg->receiverId = msg->senderId;
          unicast_message(&newmsg);
    }
```

(b) Unicast

Figure 2.3.1: (a) Pseudo code for initiating "ping pong" message. As it is shown, the first node uses broadcast to initiate the transmission. (b) Code for receiving message "handler". After receiving a message "exchange" the node addresses and resend the message by using unicast.

2 Programming Wireless Sensor Networks

SOLUTION 2.1

Hello world

Before starting the solutions, let us review some basic concepts for TinyOS programming.

SOLUTION 2.2

Counter

SOLUTION 2.3

Ping Pong

SOLUTION 2.4

Dissemination Protocol

```
every 10 seconds{
cmd = random_cmmand(); //get a random command
lastSeqNum++; //increase seq num
interprete_command(cmd); //handle command
broadcast_command(cmd); //forward to surrounding nodes
}
```

(a) Sink NodeId=1

```
broadcast_msg_handler(packet* msg){
if( msg->seqNum > lastSeqNum){
lastSeqNum = msg->seqNum; //store seq num
interprete_command(msg->cmd); //handle command
broadcast_message(msg); //forward msg
}
}
```

(b) Receiver NodeId !=1

Figure 2.4.1: (a) Pseudo code for dissemination protocol. Sink node every 10 seconds picks a random command and broadcasts it to surrounding nodes. (b) Node i, after receiving **new** message interprets the command and rebroadcasts it.

3 Wireless Channel

SOLUTION 3.1

The noisy sensor

(a) Let r_i be the distance from the source to a node, and let S be the signal power. Then for node 1 to be involved in a decision rather than some other node one must have

$$\frac{S}{10r_1^2} > \frac{S}{r_i^2} \quad \text{or} \quad \frac{r_i}{r_1} > \sqrt{10}$$

For reasons of symmetry one need only consider the all-positive quadrant with node 1 as the origin. For a source at position $(x, 0)$ equal SNR is obtained with respect to node 3

$$\frac{d-x}{x} = \sqrt{10} \quad \text{or} \quad x = d/4.16\,.$$

For node 2 the result is

$$\frac{d+x}{x} = \sqrt{10} \quad \text{or} \quad x = d/2.16\,.$$

Similarly for node 4 one solves

$$\frac{2d-x}{x} = \sqrt{10} \quad \text{or} \quad x = d/2.08\,.$$

Node 5 turns out to have the second tightest constraint on the x-axis:

$$\frac{d^2+x^2}{x^2} = 10 \quad \text{or} \quad x = d/3\,.$$

Thus, node 1 is only among the two best nodes if $x < d/3$. The situation is symmetric with respect to the y-axis, with nodes 2 and 6 being better for $y > d/3$. Now consider the position (x, x); nodes 2 and 5 will have the same SNR as node 1 if

$$\frac{(d+x)^2+x^2}{2x^2} = 10 \text{ or } d^2 + 2xd - 18x^2 = 0;$$

solving the quadratic equation, $x = d/3.35$. The node diagonally opposite produces $x = d/4.16$ while nodes 6 and 3 produce a tighter result with $x = d/5.35$.

(b) For a source at position $(0.25d, 0)$ the respective SNRs for nodes 5 and 1 are $S/1.0625d^2\sigma_n^2$ and $S/0.625d^2\sigma_n^2$. Thus the ratio in signal normalized noise variances is 1.7, or put another way, the standard deviation is 1.3 times as large for node 5 as node 1. There are a variety of ways to approximate the likelihood that a given measurement will be lower at node 5 than node 1. One is to simulate; another is to divide both normalized noise distributions into bins of equal size, with events in the same bin assigned equal likelihood of being larger or smaller. The likelihoods of being in particular bins are easily obtained from the Q-function table. Normalize the noise variance for node 1 to be so that the standard deviation for node 5 is 1.3. Then, the probability that the absolute value of the noise at node 1 is less than the noise at node 5 is approximately given by the sum of P(1 in bin i)(1/2 P(5 in bin i)+P(5 larger than bin i)). Thus, with bin sizes of $\sigma_1/2$ (taking both positive and negative values) we obtain (with the last one being the rest of the distribution):

Bin	P(1 in bin)	P(5 in bin)	P(5 bigger)
1	.383	.300	.700
2	.300	.248	.452
3	.184	.204	.248
4	.088	.124	.124
5	.017	.070	.054
6	.012	.054	...

Hence P(1 less than 5) $= .383(.15 + .7) + .3(.124 + .452) + .184(.102 + .248) + .088(.062 + .124) + .017(.035 + .054) + .012(.027) = 0.62$.

SOLUTION 3.2

Power optimization

(a) If we consider $k \geq 0$ and the intermediate nodes are equidistant, then the sum of required energy between source and destination considering k intermediate node is as follows

$$E = (k+1)\left(\frac{d}{k+1}\right)^{\alpha} = d^{\alpha}(k+1)^{1-\alpha}$$

since $1 - \alpha < 0$ if k increases E decreases and in limit point, when k goes to infinity E goes to zero. So there is no optimal number of nodes to minimize the energy consumption (infinity number of nodes here makes the energy to be zero).

(b) Based on part (a), if we consider a number of nodes that tends to ∞ as optimal solution, the energy consumption goes to zero.

(c) A new model of energy consumption with the constant value is more realistic:$E(A, B) = d(A, B)^{\alpha} + C$. If we put this new formula in the limit computed in part (a), the minimum required energy for transmission would be a value greater than zero and it is more reasonable, because in real world it is impossible to send data without any energy consumption.

(d) We have

$$E = (k+1)\left(\frac{d}{k+1}\right)^{\alpha} + (k+1)C\,.$$

By taking the derivative

$$\frac{dE}{dk} = \left(\frac{d}{k+1}\right)^{\alpha} - \left(\frac{d}{k+1}\right)^{\alpha}\alpha + C = \left(\frac{d}{k+1}\right)^{\alpha}(1-\alpha) + C$$

and putting it to zero, we have:

$$k = d\left(\frac{\alpha-1}{C}\right)^{\frac{1}{\alpha}} - 1$$

which is the optimal number of intermediate nodes that minimizes the overall energy consumption.

(e) If we put the computed value of k in previous case into the energy consumption equation (of previous section), the following closed form can be achieved:

$$E(S,T) = (k+1)\left(\frac{d}{k+1}\right)^{\alpha} + (k+1)C$$

$$k = d\left(\frac{\alpha-1}{C}\right)^{\frac{1}{\alpha}} - 1$$

so

$$E(S,T) = \alpha d\left(\frac{C}{\alpha-1}\right)^{\frac{\alpha-1}{\alpha}} .$$

SOLUTION 3.3

Deriving the Density of a Function of a Random Variable: Rayleigh fading

We use the method from [4] on the calculation of pdf for functions of one random variable. Assume y be a function of x with known distribution and pdf $f_x(x)$. To find $f_y(y)$ for a specific y, we solve $y = g(x)$. Denoting its real roots by x_n,

$$y = g(x_1) = \cdots = g(x_n) = \cdots .$$

From [4], it can be shown that

$$f_y(y) = \frac{f_x(x_1)}{|g'(x_1)|} + \cdots + \frac{f_x(x_n)}{|g'(x_n)|} + \cdots ,$$

where $g'(x)$ is the derivative of $g(x)$.

For $\mathbf{y} = \sqrt{\mathbf{x}}$ and $g'(x) = 1/(2\sqrt{x})$, the equation $y = \sqrt{x}$ has a single solution $x = y^2$ for $y > 0$ and no solution for $y < 0$. Hence

$$f_y(y) = 2y f_x(y^2)U(y). \tag{3.3}$$

Suppose that $\mathbf{x}$ has a chi-square density as

$$f(x) = \frac{1}{2^{n/2}\Gamma(n/2)} x^{n/2-1} e^{-x/2} U(x),$$

and $y = \sqrt{x}$. In this case, (3.3) yields

$$f_y(y) = \frac{2}{2^{n/2}\Gamma(n/2)} y^{n-1} e^{-y^2/2} U(y).$$

This function is called the chi density with n degree of freedom. In special case, for $n = 2$, we obtain Rayleigh density $f_y(y) = ye^{-y^2/2}U(y)$.

SOLUTION 3.4

Deriving the Density of a Function of a Random Variable: Step windowing

Let $\mathbf{y} = \mathbf{x}U(\mathbf{x})$ and $g'(x) = U(x)$. Clearly, $f_y(y) = 0$ and $F_y(y) = 0$ for $y < 0$. if $y > 0$, then the equation $y = xU(x)$ has a single solution $x_1 = y$. Hence

$$f_y(y) = f_x(y) \quad F_y(y) = F_x(y) \quad y > 0.$$

Thus $F_y(y)$ is discontinuous at $y = 0$ with discontinuity $F_y(0^+) - F_y(0^-) = F_x(0)$. Hence,

$$f_y(y) = f_x(y)U(y) + F_x(0)\delta(y)$$

SOLUTION 3.5

Deriving the Density of a Function of a Random Variable: Shadow fading

We have $\mathbf{y} = \exp(\mathbf{x})$ and $g'(x) = \exp(x)$. If $y > 0$, then the equation $y = e^x$ has the single solution $x = \ln y$. Therefore

$$f_y(y) = \frac{1}{y} f_x(\ln y) \quad y > 0.$$

If $y < 0$, then $f_y(y) = 0$. Now if $\mathbf{x} \sim \mathcal{N}(\mu, \sigma^2)$, then

$$f_y(y) = \frac{1}{\sigma y \sqrt{2\pi}} e^{-(\ln y - \mu)^2/2\sigma^2}.$$

This density is called log-normal and it is the standard model for the shadow fading, namely the slow variation of the wireless channel.

SOLUTION 3.6

Mean and Variance of Log-normal Distribution

For $\mathbf{x} \sim \mathcal{N}(\mu, \sigma)$, the expected value of $\mathbf{y} = \exp(\mathbf{x})$, which has a log-normal distribution, is

$$\mathbf{E}\{\,y\} = \int_{-\infty}^{+\infty} y f(y) dy = \int_{-\infty}^{+\infty} \frac{y}{\sigma y \sqrt{2\pi}} e^{-(\ln y - \mu)^2/2\sigma^2} dy = e^{\mu + \sigma^2/2}$$

The variance of $\mathbf{y} = \exp(\mathbf{x})$ is

$$\mathbf{E}\{\,y^2\} - \mathbf{E}^2\{\,y\} = \int_{-\infty}^{+\infty} (y-\mu)^2 f(y) dy = \int_{-\infty}^{+\infty} \frac{(y-\mu)^2}{\sigma y \sqrt{2\pi}} e^{-(\ln y - \mu)^2/2\sigma^2} dy = (e^{\sigma^2} - 1)e^{2\mu + \sigma^2}$$

SOLUTION 3.7

Gillbert-Elliot model for the wireless channels

(a) Steady state probabilities are derived via the steady-state equations

$$\pi_G + \pi_B = 1,$$
$$\pi_G = (1-p)\pi_G + r\pi_B$$

which yields

$$\pi_G = \frac{r}{p+r}$$

and

$$\pi_B = 1 - \frac{r}{p+r} = \frac{p}{p+r}.$$

(b) steady state error $p_E = (1-k)\pi_G + (1-h)\pi_B$.

(c) we determine $r = 1/\text{AEL}$ and $p_E = \text{APD}$. From section (a) and (b) we have $p = p_E r/(h - p_E)$. Having r and p the stationary probabilities π_G and π_B are in order.

SOLUTION 3.8

Gillbert-Elliot Channel

(a) The average length of an error burst is the same as the average time of staying in the bad state. The probability of an error burst of length t is the same as the probability of staying in bad state for an interval of length t, which is equal to

$$\Pr\{\text{error burst of length}\, t\} = (1-r)^{t-1} r.$$

Average error burst length is

$$\text{AEL} = \sum_{t \geq 1} t \Pr\{\text{error burst of length}\, t\} = \sum_t t(1-r)^{t-1} r = \frac{1}{r} = 10.$$

(b) Similar to the last part, the average length of an error-free sequence of bits is the average time of staying in good state, which is

$$\text{AEFL} = \sum_{t \geq 1} t \Pr\{\text{error-free sequence of length}\, t\} = \sum_t t(1-p)^{t-1} p = \frac{1}{p} = 10^5.$$

(c) Looking at the system as a Markov chain, the stationary probability of being in the bad state is

$$\pi_B = \frac{p}{p+r} = \frac{10^{-5}}{10^{-1} + 10^{-5}} = \frac{1}{10^4 + 1} \approx 10^{-4}.$$

moreover, message loss rate is given based on the stationary probability of error $p_E = (1-h)\pi_B + (1-k)\pi_G$. Since $k \approx 1$ and $h \approx 0$, then $p_E \approx \pi_B \approx 10^{-4}$.

4 Physical Layer

SOLUTION 4.1

Gray Code. Using the mirror imaging several times, one example (of many) of the Gray code is as 4.1.1

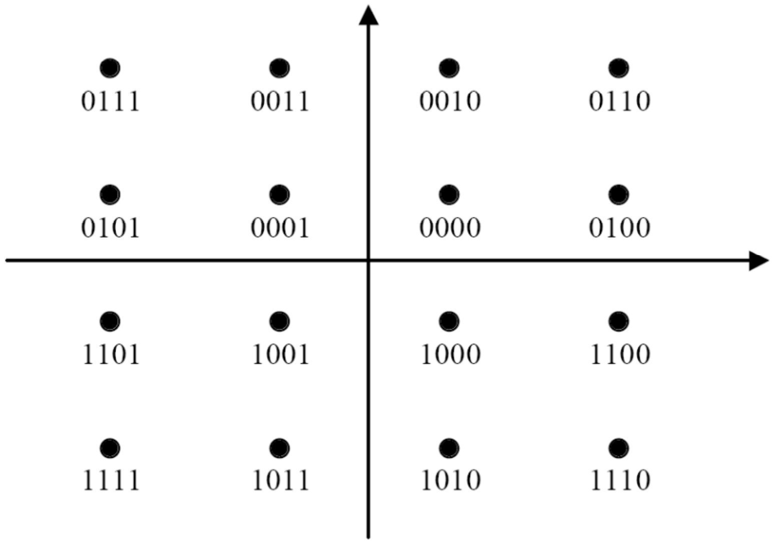


Figure 4.1.1: Gray-coded 16-QAM.

SOLUTION 4.2

Network reconfiguration

(a) The alternate route will be selected if the expected number of transmissions into and out of the malfunctioning node is 2 or greater. Let the packet dropping probability be p. The last successful transmission will have probability $(1-p)$. Then the expected number of transmissions is

$$1(1-p)+2p(1-p)+3p^2(1-p)+\cdots=2$$

Solving above equality we have

$$\sum_{i=0}^{\infty}(i+1)p^i(1-p)=\frac{1}{1-p}=2\,,$$

and hence, $p=0.5$.

(b) The probability of requiring less than or equal to 3 transmissions is

$$(1-p)+p(1-p)+p^2(1-p)=0.99$$

This is a cubic in p and can be solved in any number of ways. A probability of 0.2 is close. Thus the delay requirement leads more quickly to choice of alternative paths in this example.

SOLUTION 4.3

Bit error probability for BPSK over AWGN channels

The general form for BPSK follows the equation:

$$s_n(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f t + \pi(1-n)), \quad n = 0, 1.$$

This yields two phases 0 and π. Specifically, binary data is often conveyed via the following signals:

$$s_0(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f t + \pi) = -\sqrt{\frac{2E_b}{T_b}}\cos(2\pi f t)$$
$$s_1(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f t)$$

Hence, the signal-space can be represented by

$$\phi(t) = \sqrt{\frac{2}{T_b}}\cos(2\pi f t)$$

where 1 is represented by $\sqrt{E_b}\phi(t)$ and 0 is represented by $-\sqrt{E_b}\phi(t)$. Now we comment on the channel model. The transmitted signal that gets corrupted by noise n typically refereed as added white Gausssian noise. It is called white since the spectrum of the noise is flat for all frequencies. Moreover, the values of the noise n follows a zero mean gaussian probability distribution function with variance $\sigma^2 = N_0/2$. So for above model, the received signal take the form

$$y(t) = s_0(t) + n$$
$$y(t) = s_1(t) + n$$

The conditional probability distribution function (PDF) of y for the two cases are:

$$f(y|s_0) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y+\sqrt{E_b})^2}{N_0}}$$
$$f(y|s_1) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y-\sqrt{E_b})^2}{N_0}}$$

Assuming that s_0 and s_1 are equally probable, the threshold 0 forms the optimal decision boundary. Therefore, if the received signal y is greater than 0, then the receiver assumes s_1 was transmitted and vise versa. With this threshold the probability of error given s_1 is transmitted is

$$p(e|s_1) = \frac{1}{\sqrt{\pi N_0}}\int_{-\infty}^{0} e^{-\frac{(y-\sqrt{E_b})^2}{N_0}} dy = \frac{1}{\sqrt{\pi}}\int_{\sqrt{\frac{E_b}{N_0}}}^{\infty} e^{-z^2} dz = \mathbf{Q}\left(\sqrt{\frac{2E_b}{N_0}}\right) = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right),$$

where $\text{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-x^2} dx$ is the complementary error function. Similarly, the probability of error given s_0 is transmitted is

$$p(e|s_0) = \frac{1}{\sqrt{\pi N_0}}\int_{0}^{\infty} e^{-\frac{(y+\sqrt{E_b})^2}{N_0}} dy = \frac{1}{\sqrt{\pi}}\int_{\sqrt{\frac{E_b}{N_0}}}^{\infty} e^{-z^2} dz = \mathbf{Q}\left(\sqrt{\frac{2E_b}{N_0}}\right) = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right).$$

Hence, the total probability of error is

$$P_b = p(s_1)p(e|s_1) + p(s_0)p(e|s_0) = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right).$$

Note that the probabilities $p(s_0)$ and $p(s_1)$ are equally likely.

SOLUTION 4.4

Bit error probability for QPSK over AWGN channels

The general form for QPSK follows the equation:

$$s_n(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f t + \pi(2n-1)/4), \quad n = 1, \cdots 4.$$

In this case, the signal space can be constructed using only two basis functions:

$$\phi_1(t) = \sqrt{\frac{2}{T_b}} \cos(2\pi f t)$$
$$\phi_2(t) = \sqrt{\frac{2}{T_b}} \sin(2\pi f t)$$

The first basis function is used as the in-phase component of the signal and the second as the quadrature component of the signal. Hence, the signal constellation consists of the signal-space 4 points

$$\left(\pm\sqrt{E_b/2}, \pm\sqrt{E_b/2}\right)$$

The factors of $1/2$ indicate that the total power is split equally between the two carriers. Comparing these basis functions with that for BPSK shows clearly how QPSK can be viewed as two independent BPSK signals. Although QPSK can be viewed as a quaternary modulation, it is easier to see it as two independently modulated quadrature carriers. With this interpretation, the even (or odd) bits are used to modulate the in-phase component of the carrier, while the odd (or even) bits are used to modulate the quadrature-phase component of the carrier. BPSK is used on both carriers and they can be independently demodulated. As a result, the probability of bit-error for QPSK is the same as for BPSK:

$$P_b = \mathbf{Q}\left(\sqrt{\frac{2E_b}{N_0}}\right) = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right).$$

However, in order to achieve the same bit-error probability as BPSK, QPSK uses twice the power (since two bits are transmitted simultaneously). The symbol error rate is given by:

$$P_s = 1 - (1 - P_b)^2 = 2\mathbf{Q}\left(\sqrt{\frac{2E_b}{N_0}}\right) - \mathbf{Q}^2\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

If the signal-to-noise ratio is high (as is necessary for practical QPSK systems) the probability of symbol error may be approximated:

$$P_s \sim 2\mathbf{Q}\left(\sqrt{\frac{2E_b}{N_0}}\right).$$

SOLUTION 4.5

Error probability for 4-PAM over AWGN channels

Signal sets are designed to maximize the minimum distance between the points in signal space, subject to peak and average power constraints. This is because the low error rates is desired. The symbol error probability is closely approximated by

$$P(e) \sim N_{d_{\text{min}}} \mathbf{Q}\left(\frac{d_{\text{min}}}{\sqrt{2N_0}}\right),$$

where $N_{d_{\text{min}}}$ is the average number of nearest neighbors at the minimum distance. Pulse amplitude modulation (PAM) is effected by multiplying a rectangular pulse of duration T by one of M equally spaced voltage levels symmetric about the origin. It can be easily shown that this symmetry minimizes the peak and average power, without affecting the error probability. In the particular case of 4-PAM, the signals are given by

$$S_{00} = \sqrt{E/5}\phi_1(t), s_{01} = 3\sqrt{E/5}\phi_1(t), \qquad s_{10} = -\sqrt{E/5}\phi_1(t), s_{11} = -3\sqrt{E/5}\phi_1(t),$$

where $\phi_1(t) = \sqrt{1/T}$, $0 \leq t \leq 1T$. Clearly, the average energy is $E = [(2)(E/5) + (2)(9E/5)]/4$, and the squared minimum distance is $4E/5$. Here s_{00} and s_{10} have two neighbors at the minimum distance, while s_{11} and s_{01} have only one. Thus$N_{d_{\min}} = 1.5$. Also, the most likely error is to cross only into the neighboring region, for which the bit labels differ in only one position. Thus a symbol error results in only one of the two bits being in error. Consequently, the symbol error rate is

$$P(e) \sim 1.5\mathbf{Q}\left(\sqrt{\frac{2E}{5N_0}}\right) = 1.5\mathbf{Q}\left(\sqrt{\frac{4E_b}{5N_0}}\right),$$

and bit error rate

$$P_b(e) \sim 0.75\mathbf{Q}\left(\sqrt{\frac{4E_b}{5N_0}}\right),$$

where,

$$E_b = \text{energy per bit} = \frac{E}{\log_2 M} = \frac{E}{2}.$$

SOLUTION 4.6

Average error probability for Rayleigh fading

Let $P(\gamma)$ be the probability of error for a digital modulation as a function of E_b/N_0, γ, in the Gaussian channel. Let the channel amplitude be denoted by the random variable α, and let the average SNR normalized per bit be denoted by $\gamma^\star = \mathbf{E}[\alpha^{\mathbf{2}}]\mathbf{E_b}/\mathbf{N_0}$. Then to obtain $P(e)$ for a Rayleigh fading channel $P(\gamma)$ must be integrated over the probability that a given γ is encountered:

$$P(e) = \int_0^\infty P(\gamma)p(\gamma)d\gamma,$$

For Rayleigh fading,

$$p(\gamma) = \frac{1}{\gamma^\star}e^{-\gamma/\gamma^\star}.$$

In the case of coherent BPSK, the integration can actually be computed yielding

$$P(e) = \frac{1}{2}\left[1 - \sqrt{\frac{\gamma^\star}{1+\gamma^\star}}\right].$$

At high SNR like OQPSK systems, the approximation $(1+x)^{1/2} \sim 1 + x/2$ can be used, giving

$$P(e) \sim \frac{1}{4\gamma^\star}$$

compared with $P(e) = \mathbf{Q}(\sqrt{2\gamma^\star})$ for the Gaussian channels.

SOLUTION 4.7

Detection in a Rayleigh fading channel

We have

$$\begin{aligned} P(e) &= \mathbf{E}[\mathbf{Q}(\sqrt{\mathbf{2|h|^2}\text{SNR}})], \\ &= \int_0^\infty e^{-x} \int_{\sqrt{2x\text{SNR}}}^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2}\, dt dx = \frac{1}{\sqrt{2\pi}} \int_0^\infty \int_0^{t^2/(2\text{SNR})} e^{-t^2/2} e^{-x} dx dt \\ &= \frac{1}{\sqrt{2\pi}} \int_0^\infty e^{-t^2/2} \left(1 - e^{-t^2/(2\text{SNR})}\right) dt = \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \int_0^\infty e^{-t^2(1+1/\text{SNR})/2} dt \\ &= \frac{1}{2}\left(1 - \sqrt{\frac{\text{SNR}}{1+\text{SNR}}}\right), \end{aligned}$$

In the first step we take into account that $|h|$ is a Rayleigh random variable; i.e., it has the density $\frac{x}{\sigma^2} e^{\frac{-x^2}{2\sigma^2}}$ and hence its squared magnitude $|h|^2$ is exponentially distributed with density $\frac{1}{\sigma^2} e^{\frac{-x}{\sigma^2}}$, $x \geq 0$. Remember that according to the question assumptions $\sigma = 1$. Moreover, the third step follows from changing the order of integration.

Now, for large SNR, Taylor series expansion yields

$$\sqrt{\frac{\text{SNR}}{1+\text{SNR}}} = 1 - \frac{1}{2\text{SNR}} + \mathbf{O}\left(\frac{\mathbf{1}}{\text{SNR}^{\mathbf{2}}}\right) \sim \mathbf{1} - \frac{\mathbf{1}}{\mathbf{2}\text{SNR}}$$

which implies

$$P(e) \sim \frac{1}{4\text{SNR}}.$$

SOLUTION 4.8

Average error probability for Log-normal fading

First, we present a short summary of the Stirling approximation. A natural way of approximating a function is using the Taylor expansion. Specially, for a function $f(\theta)$ of a random variable θ having mean μ and variance σ^2, using the Taylor expansion about the mean we have

$$f(\theta) = f(\mu) + (\theta - \mu) f'(\mu) + \frac{1}{2}(\theta - \mu)^2 f''(\mu) + \cdots.$$

By taking expectation

$$\mathbf{E}\{f(\theta)\} \sim f(\mu) + \frac{1}{2} f'(\mu)\sigma^2.$$

However, in Stirling approximation one can start with these differences

$$f(\theta) = f(\mu) + (\theta - \mu)\frac{f(\mu+h) - f(\mu-h)}{2h} + \frac{1}{2}(\theta-\mu)^2 \frac{f(\mu+h) - 2f(\mu) + f(\mu-h)}{h^2} + \cdots,$$

then, taking the expectation we have

$$\mathbf{E}\, f(\theta) \sim f(\mu) + \frac{1}{2}\frac{f(\mu+h) - 2f(\mu) + f(\mu-h)}{h^2}\sigma^2.$$

It has been shown that $h = \sqrt{3}$ yields a good result. So we obtain $\mathbf{Q}(\gamma)$. Given a log-normal random variable z with mean μ_z and variance σ_z^2, we calculate the average probability of error as the average of $\mathbf{Q}(\gamma)$. Namely,

$$\mathbf{E}\{\mathbf{Q}(z)\} \sim \frac{2}{3}\mathbf{Q}(\mu_z) + \frac{1}{6}\mathbf{Q}(\mu_z + \sqrt{3}\sigma_z) + \frac{1}{6}\mathbf{Q}(\mu_z - \sqrt{3}\sigma_z).$$

SOLUTION 4.9

Probability of error at the message level.

(a) Given the bit error probability $P(e)$, the probability of successfully receiving a message is

$$p = (1 - P(e))^f.$$

For Rayleigh fading, from Exercise 4.7 we have

$$P(e) \approx \frac{1}{4\text{SNR}}.$$

(b) To have a message reception probability of at least $p > 0.35$, it is required that

$$\left(1 - \frac{1}{4\text{SNR}}\right)^f > 0.9^{10}$$

By substituting $f = 10$, and

$$\text{SNR} = \frac{\alpha E_b}{N_0 d^2} = \frac{10}{d^2}$$

in inequality above we obtain

$$\left(1 - \frac{d^2}{40}\right)^{10} > (0.9)^{10} \Longrightarrow d \leq 2.$$

SOLUTION 4.10

The rate $2/3$ parity check code. A sequence of three **PSK** symbols can be represented by vertices of a cube, with each coordinate axis corresponding to signal space for individual **PSK** symbols. This is illustrated in Figure 4.10.1, using the convention that a 1 indicates a positive coordinate, while a 0 represents a negative coordinate. Clearly, the squared minimum distance is now $8E$, compared with $4E$ for uncoded **PSK**. Hence E_b/N_0 performance has improved by $2 \times 2/3 = 4/3$ or 1.25d**B**.

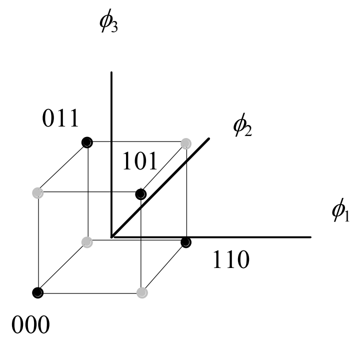


Figure 4.10.1: $2/3$ parity check code.

SOLUTION 4.11

Hard vs. soft decisions.

(a) The Hamming distance is 2. The code can detect one error and cannot correct any errors.

(b) Since it cannot tolerate any errors, the error probability is

$$p_e = 1 - p^3$$

where

$$p = 1 - Q\left(\frac{\sqrt{E}/2}{\sqrt{N_0/2}}\right) = 1 - Q\left(\frac{\sqrt{2E_b/3}}{\sqrt{2N_0}}\right) = 1 - Q\left(\sqrt{\frac{E_b}{3N_0}}\right)$$

(c) If the $Q(\cdot)$ in previous part is very small, the error probability can be approximated as

$$p_e = 3Q\left(\sqrt{\frac{E_b}{3N_0}}\right)$$

The error probability for soft-decisions is

$$p_e = Q\left(\sqrt{\frac{8E_b}{3N_0}}\right)$$

which is lower than the error probability with hard-decisions.

5 Medium Access Control

SOLUTION 5.1

Slotted Aloha.

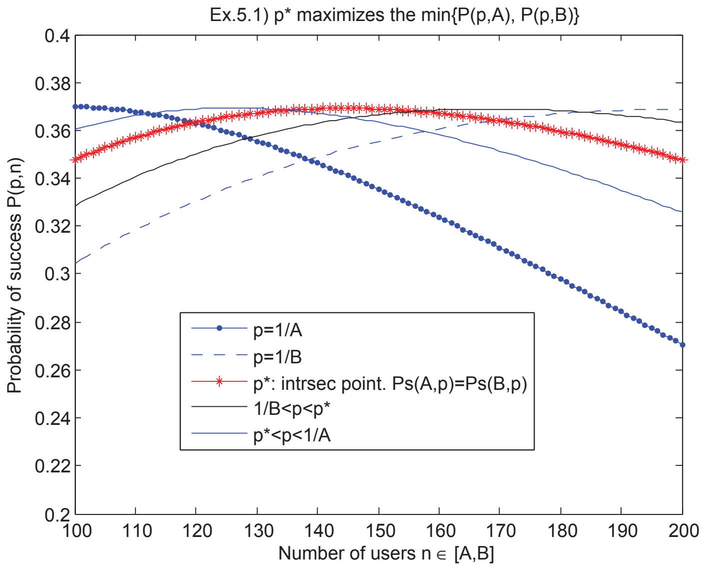


Figure 5.1.1: maximum worst-case success probability in Slotted Aloha is achieved for the point $p^\star$ fulfilling $P(p^\star, A) = P(p^\star, B)$.

(a) We define the function $P : \mathbb{R}^2 \to \mathbb{R}$ as

$$P(p, n) := \Pr(\text{success}) = n \cdot p(1-p)^{n-1} .$$

For a fixed p, $P(p, n)$ is monotone increasing for $n \leq -1/\ln(1-p)$ and monotone decreasing for $n \geq -1/\ln(1-p)$ and therefore $P(p, n)$ is minimized either at $n = A$ or at $n = B$ for $n \in [A, B]$. Therefore, we have to find

$$p^\star = \arg\max_p(\min\{P(p, A), P(p, B)\}) .$$

For a fixed n, $P(p, n)$ is monotone increasing for $p \leq 1/n$ and monotone decreasing for $p \geq 1/n$ (for $p \in [0, 1]$). Furthermore, $P(1/A, A) \geq P(1/A, B)$ and $P(1/B, B) \geq P(1/B, A)$ for $B \geq A + 1$ and therefore the intersection between $P(p, A)$ and $P(p, B)$ is between the maxima of $P(p, A)$ and $P(p, B)$, respectively. Thus $p^\star$ is found where $P(p^\star, A) = P(p^\star, B)$. Therefore,

$$A \cdot p^\star \cdot (1-p^\star)^{A-1} = B \cdot p^\star \cdot (1-p^\star)^{B-1}$$
$$\frac{A}{B} = (1-p^\star)^{B-1-(A-1)} = (1-p^\star)^{B-A}$$
$$p^\star = 1 - \sqrt[B-A]{\frac{A}{B}}$$

Figure 5.1.1 plots $P(p, n)$ versus number of nodes $n \in [A, B]$, where $A = 100$ and $B = 200$ and the optimal $p^\star$ that maximizes $\min\{P(p, A), P(p, B)\}$.

Note:another explanation of the maxima for the worst case is as following. The worst case for success probability $P(p, n)$ *w.r.t.* n *happens when either* $n = A$ *and* $n = B$*. On the other hand, for such cases*

$p^\star = 1/n$ is the maximizer. Now assume if we pick $p = 1/A$ but it happen that $n = B$ is the worst case then if we decrease $p^\star$ toward $1/B$ will increase $P(p, n = B)$. Similarly, if we choose $p^\star = 1/B$ and it happen that worst case success probability is for $P(p, n = A)$, then increasing $p^\star$ toward $1/A$ is maximizing $P(p, n = A)$. Consequently, To be sure that we have picked the best $p^\star$ for the worst case of n we find it when $P(p^\star, A) = P(p^\star, B)$. Moreover, One can check that $p^\star = 1 - \sqrt[B-A]{A/B}$ is in $[1/B, 1/A]$.

(b) For $A = 100$ and $B = 200$, we get

$$p^\star = 0.006908 = \frac{1}{144.8}$$

.

SOLUTION 5.2

ARQ

(a) That the packet is correctly received after n transmissions is to say that there are $n - 1$ corrupted transmissions and 1 correct transmission. The probability is thus given by

$$P_n = (1 - P_e)P_e^{n-1}\,.$$

The average number of transmissions required for a correct reception is

$$\begin{aligned}N &= \sum_{n=0}^{\infty} nP_n = \sum_{n=1}^{\infty} n(1 - P_e)P_e^{n-1} = (1 - P_e)\frac{d}{dP_e}\sum_{n=1}^{\infty} P_e^n \\ &= (1 - P_e)\frac{d}{dP_e}\Big(\frac{P_e}{1 - P_e}\Big) = \frac{1}{1 - P_e}\end{aligned}$$

(b) Since there are a total number of N data packet transmission and $N - 1$ acknowledgement packet transmissions, the average delay experienced by a packet is

$$D = \Big(\frac{L}{R} + t_d\Big)N + \Big(\frac{L_{\text{ARQ}}}{R} + t_d\Big)(N - 1)\,,$$

where N is the average number of transmissions from (a).

SOLUTION 5.3

Analysis of CSMA based MAC in WSNs

(a) Consider a contention round with length M and total number of sensors N. Let x_n denotes the selected slot of node n. $p_s(m)$ is the probability of having a successful transmission at slot m which happens when a node selects slot m and rest of the nodes select greater slots. P_s is the probability of success over the entire contention round and is obtained by the summation over $p_s(m)$

$$P_s = \sum_{m=1}^{M} p_s(m) = \sum_{m=1}^{M}\sum_{n=1}^{N} \text{Prob}\{x_n = m,\ x_j > m\ \forall j \neq n\}$$

$$= \sum_{m=1}^{M} \binom{N}{1}\frac{1}{M}\Big(1 - \frac{m}{M}\Big)^{N-1}$$

(b) Denote $p_c(m)$ as the probability of collision at slot m then

$$\begin{aligned}p_c(m) &= \text{Prob}\{x_n \geq m,\ \forall n\} \cdot \Big[1 - \text{Prob}\{x_n = m,\ x_j > m \forall j \neq n | x_n \geq m\ \forall n\} \\ &- \text{Prob}\{x_n \geq m+1 | x_n \geq m,\ \forall n\}\Big] \\ &= \frac{1}{M^N}\Big[(M-m+1)^N - (N+M-m)\cdot(M-m)^{N-1}\Big]\end{aligned}$$

Which is essentially one minus the probability of having successful or idle slots. Also the probability of having collision after contention round can be formulated in a similar way as success case. i.e.,

$$P_c = \sum_{m=1}^{M} p_c(m) = 1 - P_s.$$

SOLUTION 5.4

MAC optimization for a distributed estimation application

1. for this tradeoff of transmission rate and success probability we can minimize the state vector transmission interval normalized by state vector success probability. Hence we have

$$\text{minimize} \quad f(z) = \frac{T_u}{P_s(z)} = \frac{1}{zP_s(z)}$$

   where $P_s(z)$ is the probability of success given that a node is sending in current slot. $P_s(z) = (1-\tau)^{N-1}$, where $\tau = z/S$.

2. The problem is convex for $\tau \in [0,1]$. This is because computing in 0 and 1 the function tends to infinity, it is continuous and down-ward oriented in between, and in that interval there is only one critical point: By taking first derivative of $f(z)$, we have

$$f'(z) = \left(\frac{\frac{1}{z}}{(1-\frac{z}{S})^{N-1}}\right)' = \left(z(1-\frac{1}{S})^{N-1}\right)^{-2}\left((1-\frac{z}{S})^{N-2}(1-\frac{z}{S}-(N-1)\frac{z}{S})\right) = 0$$

   which gives $z^\star = S/N$. This is also the point that solved the minimization problem.

SOLUTION 5.5

Broadcast. Student A is right. An exemplary algorithm: Source originating the broadcast: Transform the message m as follows: Replace a 1 with 10 and append 11 at the end and at the front of a message, i.e. message $m = 10110$ becomes message $m' = 11 \quad 10010100\ 11$. Transmit ”Hello" in round i if bit i of m' is 1. If a node is not the source it waits until it detects twice a non-free channel for two consecutive rounds. It decodes a non-free channel as 1 and a free channel as 0. It can easily reconstruct the message m by ignoring 11 at the beginning and end and replacing 10 with 1 for the bits received in between the first received 11 and the second 11. As soon as a node decoded the entire message m, it starts to transmit the same m' in the same way as the source.

SOLUTION 5.6

M/M/1 queue

(a) The average number of customers in the queue is

$$N = \sum_{n=0}^{\infty} np(n) = \sum_{n=0}^{\infty} n(1-\rho)\rho^n = (1-\rho)\rho \frac{d}{d\rho} \sum_{n=1}^{\infty} \rho^n$$

,

$$(1-\rho)\rho \frac{d}{d\rho}\left(\frac{\rho}{1-\rho}\right) = \frac{\rho}{1-\rho}\,.$$

(b) The plot is shown in Figure 5.6.1. The average number of customers in the queue will become unbounded when $\rho \geq 1$.

(c) From Little's theorem, the average delay experienced by customers is

$$D = \frac{N}{\lambda} = \frac{1}{\mu - \lambda}\,.$$

The average service time experience by customers is $1/\mu$. Therefore, the average waiting time is

$$w = D - \frac{1}{\mu} = \frac{\lambda}{\mu - \lambda}\,.$$

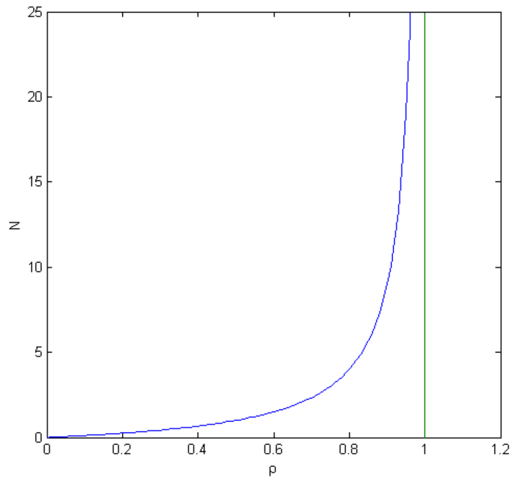


Figure 5.6.1: Average number of customers in the queue as a function of ρ

6 Routing

SOLUTION 6.1

Shortest path routing: Bellman-Ford algorithm.

(a) One can see that

$$D_i^0 = \begin{cases} 0 & i = 0 \\ \infty & \text{otherwise} \end{cases}$$

and d_{ij} is described by the following matrix, in which the element at i-th row and j-th column is d_{ij}:

$$[d_{ij}] = \begin{bmatrix} 0 & 1 & 5 & \infty & \infty & \infty \\ 1 & 0 & 1 & \infty & 10 & \infty \\ 5 & 1 & 0 & 4 & \infty & \infty \\ \infty & \infty & 4 & 0 & 1 & 3 \\ \infty & 10 & \infty & 3 & 0 & 1 \\ \infty & \infty & \infty & 3 & 1 & 0 \end{bmatrix}.$$

(b) From what we have in (a), we can determine

$$D_i^1 = \begin{cases} 1 & i = 2 \\ 5 & i = 3 \\ \infty & i = 4, 5, \text{and } 6\,. \end{cases}.$$

(c) Continue the iterations, for $h = 2$ going forward:

h	D_1^h	D_2^h	D_3^h	D_4^h	D_5^h	D_6^h
1	0	1	5	∞	∞	∞
2	0	1	2	9	11	∞
3	0	1	2	6	10	12
4	0	1	2	6	7	9
5	0	1	2	6	7	8
6	0	1	2	6	7	8

The minimum distance from node i to node 1 is determined when $h = 6$, since there is no change in the table from the previous iteration.

SOLUTION 6.2

Dijkestra. The iteration is illustrated in the following steps:

Step 1:

$$D_j = \begin{cases} 1 & j = 2 \\ 5 & j = 3 \\ \infty & j = 4, 5, \text{and } 6, \end{cases} \qquad i = 2, P = \{1, 2\}\,.$$

Step 2:

$$D_j = \begin{cases} 2 & j = 3 \\ 11 & j = 5 \\ \infty & j = 4, \text{and } 6, \end{cases} \qquad i = 3, P = \{1, 2, 3\}\,.$$

Step 3:

$$D_j = \begin{cases} 6 & j = 4 \\ 11 & j = 5 \quad i = 4, P = \{1,2,3,4\}. \\ \infty & j = 6, \end{cases}$$

Step 4:

$$D_j = \begin{cases} 7 & j = 5 \\ 9 & j = 6 \end{cases} \quad i = 5, P = \{1,2,3,4,5\}.$$

Step 5:

$$D_j = 8 \quad j = 6$$

$$i = 6, P = \{1,2,3,4,5,6\}.$$

Solution 6.3

Shortest path routing in WSNs

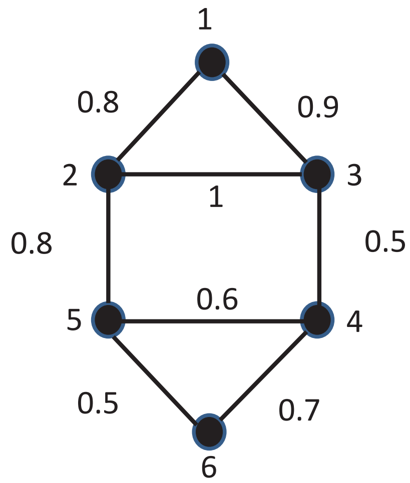


Figure 6.3.1: A sample topology of the WSN. Node 1 is the sink and link qualities(PRR) are depicted on each arcs

(a) Denote ETX$[x_i]$ as the expected number of transmissions required for node x_i to send a packet to the sink. Also, denote $\mathcal{N}_i$ and $\mathcal{P}_i$ as the neighbors set and parent of node i, respectively. Then given PRR(i, j) as the packet reception rate from i to j, One can formulate ETX$[x_i]$ as

$$\text{ETX}[x_i] = \min_{j \in \mathcal{N}_i} \left\{ \text{ETX}[x_j] + \frac{1}{\text{PRR}(i,j)} \right\}$$

and $\mathcal{P}_i = \{x_j\}$ where x_j is the neighbor that minimizes the ETX$[x_i]$. ETX$[x_1] = 0$ where x_1 is the sink. Starting from the sink, nodes put their ETX equal by infinity. Then sink propagates its ETX value to one hop neighbors, they update their ETX and broadcast their values. Whenever a node receives a ETX message from a neighbor, it checks if the value differs from the previous reported value. If so they update their ETX and parent node (if it happens) and broadcast their ETX. It can be proved that this algorithm constructs a MST and converges in a couple of iterations (as long as PRR values remain unchanged).

(b) According to Figure 6.3.1 nodes update their ETX as following

$$\text{ETX}[1] = 0$$

node 3:

$$\text{ETX}[3] = \min\left\{\frac{1}{0.9}, 1 + \text{ETX}[2]\right\} = \min\{1.1, \infty\} = 1.1,\ \mathcal{P}_3 = \{1\}.$$

node 2:

$$\text{ETX}[2] = \min\left\{\frac{1}{0.8}, 1 + \text{ETX}[3]\right\} = \min\{1.25, 2.1\} = 1.25,\ \mathcal{P}_2 = \{1\}.$$

Note that here we assumed node 2 receives the ETX[3] before computing its value.

node 4:

$$\text{ETX}[4] = \min\left\{\frac{1}{0.5} + \text{ETX}[3],\ \frac{1}{0.6} + \text{ETX}[5],\ \frac{1}{0.7} + \text{ETX}[6]\right\} = \min\{3.1,\ \infty,\ \infty\} = 3.1,\ \mathcal{P}_4 = \{3\}.$$

node 5:

$$\text{ETX}[5] = \min\left\{\frac{1}{0.8} + \text{ETX}[2],\ \frac{1}{0.6} + \text{ETX}[4],\ \frac{1}{0.5} + \text{ETX}[6]\right\} = \min\{2.5,\ 4.77,\ \infty\} = 2.5,\ \mathcal{P}_5 = \{2\}.$$

node 6:

$$\text{ETX}[6] = \min\left\{\frac{1}{0.7} + \text{ETX}[4],\ \frac{1}{0.5} + \text{ETX}[5]\right\} = \min\{4.53,\ 4.5\} = 3.5,\ \mathcal{P}_6 = \{5\}.$$

in next iteration all the ETX values will remain unchanged and the algorithm converges. The set of $\mathcal{P}$'s builds the topology.

SOLUTION 6.4

Anycast routing in WSNs

(a) We calculate the new ETX metric for s and a given ordered set of parents $\mathcal{P}_s = \{1 \dots n\}$. The probability of success in this structure is given by

$$P_{\text{success}} = 1 - \prod_{i \in \mathcal{P}_s} (1 - p_i).$$

Hence, the average number of TX to reach one hop neighborhood is given by $\text{ETX}_1[s] = 1/P_{\text{success}}$. The second part corresponds to the average cost to reach the destination. This average is given by

$$\text{ETX}_2[s] = \sum_{i \in \mathcal{P}_s} \Pr(\text{i is forwader}|\text{success at first hop})\text{ETX}[i]$$

where

$$\Pr(\text{i is forwader}|\text{success at first hop}) = \frac{p_i \prod_{j=1}^{i-1}(1 - p_j)}{P_{\text{success}}},$$

which is the probability that node i receives successfully from s and all the parents prior than i, i.e., $1, \dots i-1$ fail. Remember we condition this probability that at least one of the parents receives successfully. In total, $\text{ETX}[s]$ for a given ordered set of parents $\mathcal{P}_s = \{1 \dots n\}$ is as following

$$\text{ETX}[s] = \text{ETX}_1[s] + \text{ETX}_2[s].$$

(b) ETX with one parent (traditional ETX) for the example is $\text{ETX}[s] = 1/0.5 + 2 = 4$ and $\text{P}_s = \{1\}$. For the case of multi-parents, $\text{ETX}[s]_{\{1,2,3\}} = 3.66$.

(c) To find optimal parent set, one needs to calculate ETX for all subsets of neighbors (here 7 subsets) as the parent set. But there is a useful result: the optimal parent set is an ordered set of parents where the neighbors are sorted based on the increasing order of their ETX. So for this example this set can be either $\{1\}$ or $\{1, 2\}$, or {1,2,3}. If we calculate the ETX based on section (a) with different parent sets as input we will get $\text{ETX}[s]_{\{1\}}$ = 4, $\text{ETX}[s]_{\{1,2\}} = 3.5$, $\text{ETX}[s]_{\{1,2,3\}} = 3.66$. So the optimal ETX[$s$]=3.5 with parent set $\mathcal{P}_s^\star = \{1, 2\}$.

SOLUTION 6.6

Spanning Tree

(a) There are 3 spanning trees with node 1 as the root.

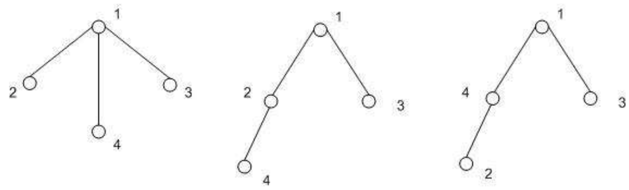


Figure 6.6.1: N=4, A=3.

(b) There are 5 spanning trees with node 1 as the root.

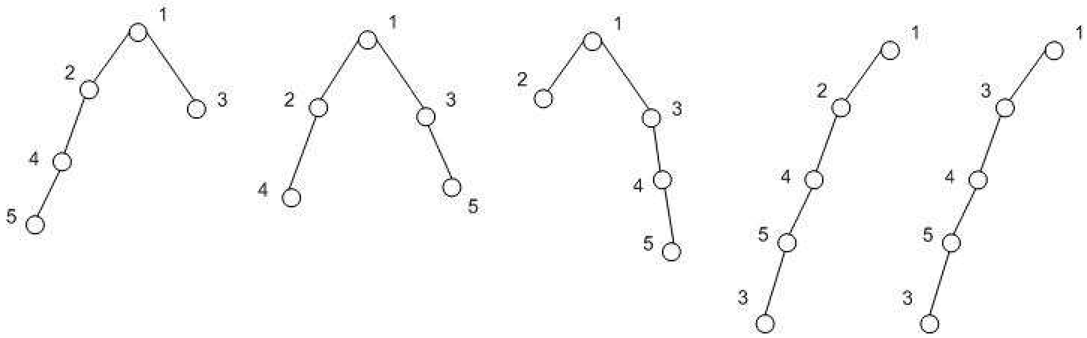


Figure 6.6.2: N=5, A=4. The following relation holds: $N = A + 1$.

SOLUTION 6.6

Directed diffusion First compute the probability of choosing particular paths for data exiting a node.

The probability of transmission taking place on each branch is listed in the Figure 6.6.2. Therefore the likelihood of transmission flowing through B and C are 241/1344 and 1103/1344 respectively. Notice that there is bi-directional flow of packets along one link, and here one must invoke the rule against back-tracking to compute the output flows for the two nodes it connects.

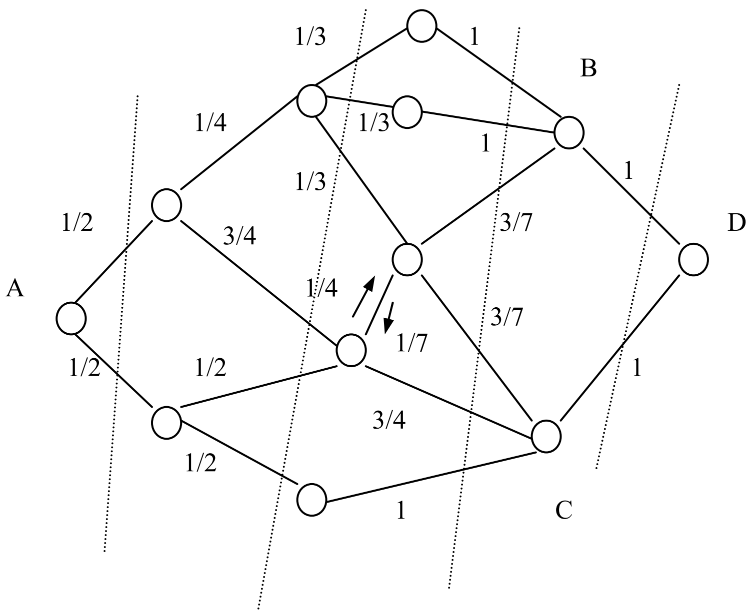


Figure 6.6.1: The probability of choosing a particular path for data exiting a node

7 Detection

SOLUTION 7.1

Binary choice in Gaussian noise. The decision threshold γ is chosen under the MAP criterion so that $\Pr(z|H_1) = \Pr(z|H_0)$, i.e., equal error probabilities result. Due to the equal prior probabilities, this is also the ML threshold. The threshold is obviously $k/2$ due to the symmetry of Gaussian distribution, with the error probability being the area of the tail past the threshold. In this case it is

$$\int_{k/2}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} dx = Q\left(\frac{k}{2\sigma}\right) .$$

Notice that what determines the error probability is the ratio of the voltage difference between the two hypothesis to the expected noise amplitude, and not the absolute voltage levels.

SOLUTION 7.2

Binary hypothesis test and SNR. The power of the hypothesized noise as a function of the threshold γ is

$$E[n^2(\gamma)] = (-k)^2 \Pr(D_1|H_0) + (k)^2 \Pr(D_0|H_1) = \frac{k^2}{2}\left[\int_{\gamma}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} dx + \int_{-\infty}^{\gamma} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-k)^2}{2\sigma^2}} dx\right] .$$

The SNR maximum occurs when we let the $E[n^2(\gamma)]$ minimum, while $dE[n^2(\gamma)]/d\gamma = 0$. Thus,

$$dE[n^2(\gamma)]/d\gamma = \frac{k^2}{2}\left(-\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-\gamma^2}{2\sigma^2}} + \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\gamma-k)^2}{2\sigma^2}}\right) := 0 ,$$

$$\gamma = \frac{k}{2} .$$

Furthermore, we should check whether $d^2E[n^2(\gamma)]/d\gamma^2 > 0$, when $\gamma = k/2$, to guarantee it is the minimum. The derivative $d^2E[n^2(\gamma)]/d\gamma^2$ can be obtained by

$$d^2E[n^2(\gamma)]/d\gamma^2 = \frac{k^2}{2}\left[\frac{1}{\sigma\sqrt{2\pi}}\left(\frac{k}{\sigma^2} e^{-\frac{-(\gamma-k)^2}{2\sigma^2}} - \frac{\gamma}{\sigma^2} e^{-\frac{-(\gamma-k)^2}{2\sigma^2}} + \frac{\gamma}{\sigma^2} e^{-\frac{-\gamma^2}{2\sigma^2}}\right)\right] .$$

Thus, when $\gamma = k/2$, $d^2E[n^2(\gamma)]/d\gamma^2 > 0$.

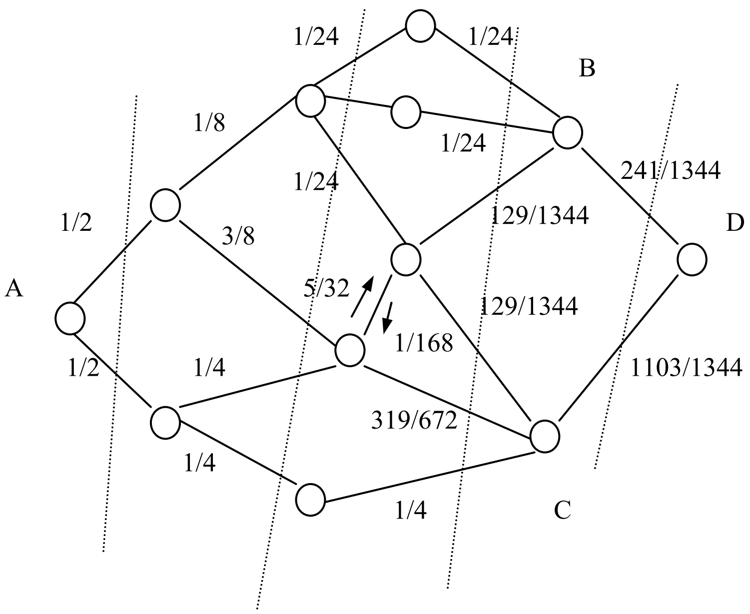


Figure 6.6.2: Probability of transmission for each branch

SOLUTION 7.3

MAP and the LRT The MAP decision rule is based on if $\Pr(s_1|z) > \Pr(s_0|z)$. This criterion can be rewritten as

$$\frac{\Pr(s_1, z)}{\Pr(z)} > \frac{\Pr(s_0, z)}{\Pr(z)} .$$

Since $\Pr(z)$ shows up on both sides, it can be discarded. It can expressed as

$$\Pr(z|s_1)\Pr(s_1) > \Pr(z|s_0)\Pr(s_0) .$$

This is equivalent to the likelihood ratio test

$$\frac{\Pr(z|s_1)}{\Pr(z|s_0)} > \frac{\Pr(s_0)}{\Pr(s_1)} .$$

SOLUTION 7.4

Binary decisions with unequal a priori probabilities The likelihood ratio test is

$$\frac{\exp\left(\frac{-(z-k)^2}{2\sigma^2}\right)}{\exp\left(\frac{-z^2}{2\sigma^2}\right)} > \frac{1}{2}$$

$$\exp\left(\frac{-1}{2\sigma^2}(-2kz + k^2)\right) > 2$$

$$z > \frac{2\sigma^2 \ln 2 - k^2}{-2k}$$

The threshold is thus

$$\gamma = \frac{2\sigma^2 \ln 2 - k^2}{-2k} .$$

SOLUTION 7.5

Since the signal and noise are independent, the results can be obtained as

$$A = \frac{1}{N}\sum_{k=0}^{N-1} s_k^2\,,$$

while $\bar{e}$ is

$$\bar{e} = \frac{1}{N}\sum_{k=0}^{N-1} s_k e_k\,,$$

in which $e_k \sim \mathcal{N}(0, \sigma^2)$. Thus $\bar{e} \sim \mathcal{N}(0, \sigma^2/N)$.

SOLUTION 7.6

The probability can be obtained as $\Pr(|s_k| = t | y_k \notin [-3\sigma, 3\sigma])$, which can be obtained as

$$\Pr(|s_k| = t | y_k \notin [-3\sigma, 3\sigma]) = \frac{\Pr(|s_k| = t, y_k \notin [-3\sigma, 3\sigma])}{\Pr(y_k \notin [-3\sigma, 3\sigma])}\,,$$

where

$$\Pr(|s_k| = t, y_k \notin [-3\sigma, 3\sigma]) = p_t Q(0)\,,$$

and

$$\begin{aligned}\Pr(y_k \notin [-3\sigma, 3\sigma]) =& \Pr(|s_k| = t, y_k \notin [-3\sigma, 3\sigma]) + \Pr(|s_k| = 0, y_k \notin [-3\sigma, 3\sigma]) \\ =& p_t Q(0) + 2p_0 Q(3)\,,\end{aligned}$$

in which $Q(\cdot)$ is Q function of the normal distribution.

SOLUTION 7.7

Optimal Data Fusion in Multiple Sensor Detection Systems

1. We have

$$\begin{aligned}\Pr(H_1|\mathbf{u}) = \frac{\Pr(H_1, \mathbf{u})}{\Pr(\mathbf{u})} &= \frac{P_1}{\Pr(\mathbf{u})}\prod_{S_+}\Pr(u_i = +1|H_1)\prod_{S_-}\Pr(u_i = -1|H_1) \\ &= \frac{P_1}{\Pr(\mathbf{u})}\prod_{S_+}(1 - P_{M_i})\prod_{S_-} P_{M_i}\,.\end{aligned}$$

   In a similar manner,

$$\Pr(H_0|\mathbf{u}) = \frac{P_0}{\Pr(\mathbf{u})}\prod_{S_+}(1 - P_{F_i})\prod_{S_-} P_{F_i}\,.$$

   Thus, we have that

$$\log\frac{\Pr(H_1|\mathbf{u})}{\Pr(H_0|\mathbf{u})} = \log\frac{P_1}{P_0} + \sum_{S_+}\log\frac{1 - P_{M_i}}{P_{F_i}} + \sum_{S_-}\log\frac{P_{M_i}}{1 - P_{F_i}}\,.$$

2. In this case, using Bayes rule to express the conditional probabilities, we can obtain the log-likelihood ratio test as

$$\log \frac{\Pr(H_1|\mathbf{u})}{\Pr(H_0|\mathbf{u})} \underset{H_0}{\overset{H_1}{\gtrless}} 0\,.$$

Therefore using the result obtained in previous subproblem, the data fusion rule is expressed as

$$f(u_1, \ldots, u_n) = \begin{cases} 1 & \text{if} \quad \mathrm{a_0 + \sum_{i=1}^{n} a_i u_i > 0} \\ -1 & \text{otherwise} \end{cases}$$

where the optimum weights are given by

$$a_0 = \log \frac{P_1}{P_0}$$
$$a_i = \begin{cases} \log \frac{1-P_{M_i}}{P_{F_i}} & \mathrm{if u_i = +1} \\ \log \frac{1-P_{F_i}}{P_{M_i}} & \mathrm{if u_i = -1} \end{cases}$$

Solution 7.8

Counting Rule At the fusion center level, the probability of false alarm P_F is

$$P_F = \sum_{N=T}^{\infty} \Pr(N) \Pr(\Lambda \geq T | N, H_0)\,,$$

where

$$\Pr(\Lambda \geq T | N, H_0) = \sum_{i=T}^{N} \binom{i}{N} P_f^i (1 - P_f)^{N-i}\,.$$

When N is large enough, $\Pr(\Lambda \geq T | N, H_0)$ can be obtained by using Laplace-De Moivre approximation:

$$\Pr(\Lambda \geq T | N, H_0) \approx Q\left(\frac{T - NP_f}{\sqrt{NP_f(1 - P_f)}} \right)\,.$$

Similarly, the probability of detection P_D is

$$P_D = \sum_{N=T}^{\infty} \Pr(N) \Pr(\Lambda \geq T | N, H_1)\,,$$

where

$$\Pr(\Lambda \geq T | N, H_1) = \sum_{i=T}^{N} \binom{i}{N} P_d^i (1 - P_d)^{N-i}$$
$$P_d = 1 - P_f - P_m\,.$$

Solution 7.9

Matched filter and SNR Assume the received signal $r(t)$ consists of the signal $s(t)$ and AWGN $n(t)$ which has zero-mean and power spectral density $1/2N_0$ W/Hz. Suppose the signal is passed through a filter with impulse response $h(t), 0 \le t \le T$, and its output is sampled at time $t = T$. The filter response to the signal and noise components is

$$y(t) = \int_0^t r(\tau)\, h(t-\tau) d\tau = \int_0^t s(\tau)\, h(t-\tau) d\tau + \int_0^t n(\tau)\, h(t-\tau) d\tau\,.$$

At the sampling instant $t = T$, the signal and noise components are

$$y(T) = \int_0^T s(\tau) h(t-\tau) d\tau + \int_0^T n(\tau) h(t-\tau) d\tau = y_s(T) + y_n(T))\,,$$

where $y_s(T)$ represents the signal component and $y_n(t)$ the noise component. The problem is to select the filter impulse response that maximizes the output signal-to-noise ratio defined as

$$SNR_0 = \frac{y_s^2(T)}{E[y_n^2(T)]}\,.$$

The denominator is simply the variance of the noise term at the output of the filter.

$$E[y_n^2(T)] = \int_0^T \int_0^T E[n(\tau)n(t)] h(T-\tau) h(T-t) dt d\tau = \tfrac{1}{2} N_0 \int_0^T h^2(T-t) dt\,.$$

By substituting for $y_s(T)$ and $E[y_n^2(T)]$ into SNR_0 , we obtain

$$SNR_0 = \frac{\left[\int_0^T s(\tau) h(T-\tau) d\tau\right]^2}{\frac{1}{2} N_0 \int_0^T h^2(T-t) dt} = \frac{\left[\int_0^T h(\tau) s(T-\tau) d\tau\right]^2}{\frac{1}{2} N_0 \int_0^T h^2(T-t) dt}\,.$$

Since the denominator of the SNR depends on the energy in $h(t)$, the maximum output SNR over $h(t)$ is obtained by maximizing the numerator subject to the constraint that the denominator is held constant. The maximization of the numerator is most easily performed by use of the Cauchy-Schwarz inequality

$$\left[\int_{-\infty}^{\infty} g_1(t) g_2(t) dt\right]^2 \le \int_{-\infty}^{\infty} g_1^2(t) dt \int_{-\infty}^{\infty} g_2^2(t) dt\,,$$

with equality when $g_1(t) = C g_2(t)$ for any arbitrary constant C. If we set $g_1(t) = h(t)$ and $g_2(t) = s(T-t)$, it is clear that the SNR is maximized when $h(t) = Cs(T-t)$. The scale factor C^2 drops out of the expression for the SNR. The maximum output SNR obtained with the matched filter is

$$SNR_0 = \tfrac{2}{N_0} \int_0^T s^2(t) dt = 2E/N_0\,.$$

SOLUTION 7.10

Binary hypothesis testing and mutual information The mutual information between the source signals X and the decision set Y is

$$I(X;Y) = H(X) - H(X|Y)\,,$$

where $H(X) = \log 2$. We denote $P_1 = \Pr(D_1|H_1)$ and $P_0 = \Pr(D_0|H_0)$. Then $H(X|Y)$ can be expressed as

$$-\frac{1}{2}\left[P_0 \log \frac{P_0}{P_0 + 1 - P_1} + (1-P_1) \log \frac{1-P_1}{P_0 + 1 - P_1} + P_1 \log \frac{P_1}{P_1 + 1 - P_0} + (1-P_0) \log \frac{1-P_0}{P_1 + 1 - P_0}\right]\,.$$

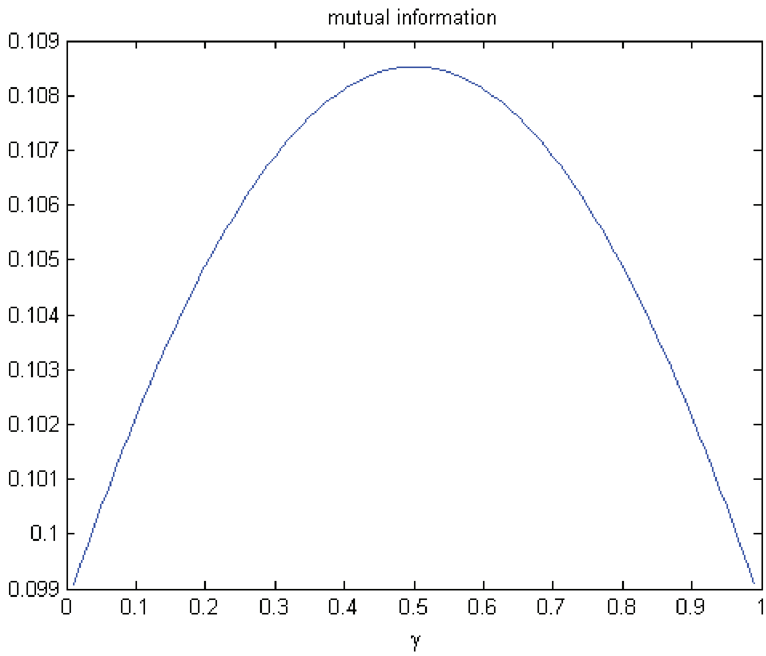


Figure 7.10.1: mutual information for different γ.

The values of P_1 and P_2 are evaluated as

$$P_1 = \int_{\gamma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}} dx \, .$$

$$P_0 = \int_{-\infty}^{\gamma} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \, .$$

The values of mutual information are plotted in Figure 7.10.1. It is obvious the maximal mutual information occurs when the threshold is equal to 0.5.

8 Estimation

SOLUTION 8.1

If one observes Y as a vector of random variables which is a function of random variable X, then a proposition shows that the best linear estimator of X to minimize the MMSE is the conditional expectation of $\mathbf{E}(X|Y = y)$. In general, we can only use this proposition. But if there are other assumptions about X, Y such as (X, Y has Gaussian distribution or X has zero mean and etc.) we can go further and prove other propositions to achieve the best estimator of X.

SOLUTION 8.2

MMSE estimator

consider the estimator of sensor fusion of the form

$$p^{-1}\hat{x} = \sum_{i=1}^{n} p_i^{-1}\hat{x}_i \,,$$

$$p_i^{-1}\hat{x}_i = H_i R_{v_i}^{-1} y_i \,.$$

Note that v_i's matrices are uncorrelated and consequently they are block diagonal. Also, consider the assumption for which each H_i is a row of the H matrix, it is evident that above iteration on each step i at the whole performs a matrix multiply between the matrices H and R_v^{-1} and vector y which its component are each sensor's measurement ($\sum_{i=1}^{n} p_i^{-1}\hat{x}_i = H R_v^{-1} y$).

For the estimate of the error covariance, noting that R_X is diagonal and consisting of different values of R_{Xii} as the covariance of node i, we can obtain the same result as MMSE estimator (the estimator calculates the same value with regard to the assumptions). From a computational point of view, in both cases the output is the same, but the computation effort is not comparable. In the first case, the computation of the matrix operation takes $O(n^2)$ operation while the order of the second one is $O(n)$, where n is the number of nodes in the network. For the first case, computing the multiplication between the matrices $H_{n\times n}$, $R_v^{-1}(n \times n)$ and $y_{n\times 1}$ needs $n^2 + n$ multiplications as well as $n^2 - 1$ additions. But for the second case, by considering the diagonal matrices, many of these operations can be omitted. Considering summation over the sensor values at the sink, we have $2n$ multiplications and n additions. Here much savings can be done especially when the number of nodes (n) is large. From an implementation point of view, this scheme is useful because it pushes the complexity of computation to the sink (central node) towards the sensors by putting some computation effort to each sensor node. It is more considerable when we note that in order to implement this estimator at a sink node, each node must send its local values to the sink directly or via some relay nodes, which has a large communication cost (overhead) for the network.

SOLUTION 8.3

Mean square (MS) estimation Let

$$\varepsilon = \mathbf{E}\left[(x-\hat{x})^2\right] = \int_{-\infty}^{\infty} (x-\hat{x})^2 f_X(x)dx$$

be the MSE. To find the estimator that minimize the MSE, we take the first and the second derivatives

$$\frac{\partial\varepsilon}{\partial\hat{x}} = -2\int_{-\infty}^{\infty} (x-\hat{x}) f_X(x)dx = 0 \,,$$

$$\frac{\partial\varepsilon}{\partial\hat{x}} = \int_{-\infty}^{\infty} f_X(x)dx = 2 > 0 \,.$$

But

$$\int_{-\infty}^{\infty} x f_X(x)dx = \mu_x = \quad \text{mean} \quad \text{of} \quad x\,,$$

$$\int_{-\infty}^{\infty} \hat{x} f_X(x)dx = \hat{x}\,.$$

Thus, the optimal estimate is

$$\hat{x} = \mu_x\,.$$

SOLUTION 8.4

Distributed MMSE estimator Considering $y = H\theta + v$, $H \in \mathbb{R}^{\mathrm{nm}\times\mathrm{m}}$ is a known matrix and $v \in \mathbb{R}^{\mathrm{m}\times 1}$ is a noise vector with zero mean with independent components having PDF $N(0, R_v)$, under these conditions the joint PDF of v due to independence of its components can be written as:

$$p(v) = \prod_{i=1}^{n} (2\pi)^{-\frac{m}{2}} \det(R_{v_i})^{-\frac{1}{2}} \exp\left(-\frac{1}{2} v_i^T R_{v_i}^{-1} v_i\right)\,.$$

Take the natural log and replace v with $y - H\theta$. Now we can write above equation in matrix form as follows:

$$\ln p(y|\theta) = \sum_{i=1}^{n} \left(-\frac{m}{2}\ln 2\pi - \frac{1}{2}\ln\det(R_{v_i})\right) - \frac{1}{2}(y - H\theta)^T R_v^{-1}(y - H\theta)\,.$$

The MLE of θ is found by minimizing

$$j(\theta) = \frac{1}{2}(y - H\theta)^T R_v^{-1}(y - H\theta)\,,$$

since this is a quadratic function of the elements of θ and R_v^{-1} is a positive definite matrix, differentiation will produce the global minimum. Now we have:

$$\frac{\partial j(\theta)}{\partial \theta} = -H^T R_v^{-1}(y - H\theta)\,.$$

By setting this equation equal by zero, we can find $\hat{\theta}$ as follows:

$$H^T R_v^{-1}(y - H\hat{\theta}) = 0\,,$$

$$H^T R_v^{-1} y - H^T R_v^{-1} H\hat{\theta} = 0\,,$$

$$\hat{\theta} = (H^T R_v^{-1} H)^{-1} H^T R_v^{-1} y\,.$$

However, all the v_i's are uncorrelated with each other. Hence R_v is a block diagonal matrix with blocks R_{v_i}. Thus, above equation can be decomposed as

$$\hat{\theta} = (H^T R_v^{-1} H)^{-1} H^T R_v^{-1} y = \left(\sum_{i=1}^{n} H_i^T R_{v_i}^{-1} H_i\right)^{-1} \sum_{i=1}^{n} H_i^T R_{v_i}^{-1} y_i\,.$$

On the other hand, to compute $R_{\hat{\theta}}$ we have

$$R_{\hat{\theta}} = \mathrm{var}\left((\mathrm{H}^{\mathrm{T}}\mathrm{R}_{\mathrm{v}}^{-1}\mathrm{H})^{-1}\mathrm{H}^{\mathrm{T}}\mathrm{R}_{\mathrm{v}}^{-1}\mathrm{y}\right)\,.$$

For a constant vector a and random variable X we have $\mathrm{var(aX) = avar(X)a^T}$. Here, the random variable is Y which has the distribution of $N(H\theta, R_v)$ so we can calculate error covariance as

$$R_{\hat{\theta}} = \left((H^T R_v^{-1} H)^{-1} H^T R_v^{-1}\right) R_v \left((H^T R_v^{-1} H)^{-1} H^T R_v^{-1}\right)^T \ ,$$

$$R_{\hat{\theta}} = \left((H^T R_v^{-1} H)^{-1} H^T R_v^{-1}\right) R_v R_v^{-1} H (H^T R_v^{-1} H)^{-1} \,,$$

$$R_{\hat{\theta}} = (H^T R_v^{-1} H)^{-1} \,.$$

SOLUTION 8.5

Cramér-Rao bound The probability of one observation is

$$p_X(x|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\theta)^2}{2\sigma^2}\right)$$

Take the natural log and the second derivative, we get

$$\ln p_X(x|\theta) = -\tfrac{1}{2}\ln(2\pi) - \ln\sigma - \frac{(x-\theta)^2}{2\sigma^2} \,,$$

$$\frac{\partial^2 \ln p_X(x|\theta)}{\partial\theta^2} = -\frac{1}{\sigma^2} \,.$$

With n observations, we have

$$\frac{\partial^2 \ln p(x_1, x_2, ..., x_n|\theta)}{\partial\theta^2} = \frac{\partial^2 \sum_{i=1}^n \ln p_X(x_i|\theta)}{\partial\theta^2} = -\frac{n}{\sigma^2}$$

Therefore, the bound is $\left\{-E[-n/\sigma^2]\right\}^{-1}$, which is σ^2/n . On the other hand, the variance of

$$\frac{x_1 + x_2 + ... + x_n}{n}$$

is σ^2/n, which is identical to the Cramér-Rao bound bound. In other words, it is an efficient estimator.

SOLUTION 8.6

ML estimates of mean and variance of Gaussian random variables The probability is

$$f(x_1, x_2, ..., x_n; \theta_1, \theta_2) = \frac{1}{\theta_2^n (2\pi)^{n/2}} e^{-\frac{1}{2\theta_2^2}\sum\limits_{i=1}^n (x_i-\theta_1)^2} \,.$$

First, we take the natural log of the probability

$$\ln f(x_1, x_2, ..., x_n; \theta_1, \theta_2) = -\frac{n}{2}\ln 2\pi - n\ln\theta_2 - \frac{1}{2\theta_2^2}\sum_{i=1}^n (x_i - \theta_1)^2 \,.$$

To find the ML estimate of θ_1 , we take the first and second derivatives of the function with respect to θ_1

$$\frac{\partial \ln f(x_1,x_2,...,x_n;\theta_1,\theta_2)}{\partial\theta_1} = \frac{1}{\theta_2^2}\sum_{i=1}^n (x_i - \theta_1) = 0$$

$$\frac{\partial^2 \ln f(x_1,x_2,...,x_n;\theta_1,\theta_2)}{\partial{\theta_1}^2} = -\frac{n}{\theta_2^2} < 0 \,.$$

Thus, the ML estimate of θ_1 is

$$\hat{\theta}_1 = \frac{\sum_{i=1}^{n} x_i}{n} = \hat{\mu}\,.$$

Similarly, we take the derivatives with respect to θ_2

$$\begin{aligned}
\tfrac{\partial \ln f(x_1,x_2,\ldots,x_n;\theta_1,\theta_2)}{\partial \theta_2} &= -\tfrac{n}{\theta_2} + \tfrac{\sum_{i=1}^{n}(x_i-\theta_1)^2}{\theta_2^3} = 0 \\
\tfrac{\partial^2 \ln f(x_1,x_2,\ldots,x_n;\theta_1,\theta_2)}{\partial {\theta_2}^2} &= \tfrac{n}{\theta_2^2} - \tfrac{3\sum_{i=1}^{n}(x_i-\theta_1)^2}{\theta_2^4}\,.
\end{aligned}$$

The ML estimate of θ_2 is

$$\hat{\theta}_2 = \left[\frac{\sum_{i=1}^{n}(x_i-\hat{\mu})^2}{n}\right]^{1/2} = \hat{\sigma}\,.$$

We also have to make sure the second derivative is less than 0. When the first derivatives w.r.t θ_1 and θ_2 are zero. The second derivative w.r.t. θ_2 is

$$\frac{\partial^2 \ln f(x_1,x_2,\ldots,x_n;\theta_1,\theta_2)}{\partial {\theta_2}^2} = \frac{n}{\hat{\sigma}^2} - \frac{3n\hat{\sigma}^2}{\hat{\sigma}^4} = -\frac{2n}{\hat{\sigma}^2} < 0\,.$$

SOLUTION 8.7

Distributed detection/estimation

(a) From the problem, we can get

$$\begin{aligned}
\Pr(m_k = 1) &= \int_{-\theta}^{1} p(v)dv = \frac{1}{2}(1+\theta) \\
\Pr(m_k = 0) &= \int_{-1}^{-\theta} p(v)dv = \frac{1}{2}(1-\theta)\,.
\end{aligned}$$

Then

$$\begin{aligned}
\mathbf{E}(m_k) &= \frac{1}{2}(1+\theta) \\
\mathbf{E}(m_k - \mathbf{E}(m_k))^2 &= \frac{1}{4}(1-\theta^2)
\end{aligned}$$

(b) From the previous results, we can obtain

$$\mathbf{E}(m_k - \mathbf{E}(m_k))^2 = \frac{1}{4}(1-\theta^2) \le \frac{1}{4}$$

(c) Based on the results from (a) and the given fusion function, we can have

$$\mathbf{E}(\hat{\theta}) = \frac{2}{N}\sum_{k=1}^{N}\mathbf{E}(m_k) - 1 = \frac{2}{N}\sum_{k=1}^{N}\frac{1}{2}(1+\theta) - 1 = \theta\,,$$

whereas

$$\mathbf{E}(\hat{\theta}-\theta)^2 = \mathbf{E}\left(\left(\frac{2}{N}\sum_{k=1}^{N} m_k - 1\right) - \theta\right)^2 = \frac{4}{N^2}\mathbf{E}\left(\sum_{k=1}^{N} m_k - \sum_{k=1}^{N}\frac{1}{2}(1+\theta)\right)^2$$

$$= \frac{4}{N^2}\mathbf{E}\left(\sum_{k=1}^{N}(m_k - \mathbf{E}(m_k))\right)^2 .$$

Since m_k's are independent, we calculate the previous equation as

$$\frac{4}{N^2}\mathbf{E}\left(\sum_{k=1}^{N}(m_k - \mathbf{E}(m_k))\right)^2 = \frac{4}{N^2}\sum_{k=1}^{N}\mathbf{E}(m_k - \mathbf{E}(m_k))^2 = \frac{1}{N}(1-\theta)^2 \leq \frac{1}{N}$$

(d) Based on the results above, we need more than $1/\epsilon$ nodes to satisfy the variance bound.

SOLUTION 8.8

Distributed detection, MAC, and routing

(a) Let x be the measured signal, which is given by the source's transmitted signal plus the measurement noise. Denote the false alarm as $\Pr(x<0, D=H_0)$, and miss detection as $\Pr(x>0, D=H_1)$, where x is the signal and D is the decision made by per every node. For the false alarm, we have

$$\Pr(x<0, D=H_0) = \Pr(x<0)\Pr(D=H_0|x<0) = \frac{1}{2}\int_{-\infty}^{0}\int_{-x}^{\infty}\frac{1}{\sigma_n}e^{-\frac{y^2}{2(\sigma_n^2+\sigma_S^2)}}dydx\,.$$

Similarly, we have

$$\Pr(x>0, D=H_1) = \Pr(x>0)\Pr(D=H_1|x>0) = \frac{1}{2}\int_{0}^{\infty}\int_{-\infty}^{-x}\frac{1}{\sigma_n}e^{-\frac{y^2}{2(\sigma_n^2+\sigma_S^2)}}dydx\,.$$

(b) Let r_i be the distance from the source to a node, and let S be the signal power. Then for node 1 to be involved in a decision rather than some other node one must have

$$\frac{S}{10r_1^2} > \frac{S}{r_i^2} \Rightarrow \frac{r_i}{r_1} > \sqrt{10}$$

For reasons of symmetry, we need only consider one of the nodes 2-6 with node 1 as the origin. Without loss of generality, let the source at position (x,y) and consider the equal SNR respecting to node 3. Then we have

$$\frac{r_3^2}{r_1^2} = \frac{(x-d)^2+y^2}{x^2+y^2} = 10 \Rightarrow \frac{(x+\frac{1}{9}d)^2}{(\frac{\sqrt{10}}{9}d)^2} + \frac{y^2}{(\frac{\sqrt{10}}{9}d)^2} = 1\,,$$

which is an ellipse with center in $(-d/9, 0)$ having x- and y-axis radius $(d\sqrt{10}/9, d\sqrt{10}/9)$ shown as the curve of E1 in Fig. 8.8.1. The regions over which node 1 is better than node 2,3 and 5.6 are shown in Fig. 8.8.1 as E1,E2 and E3,E4 respectively. Thus when the source in the shadowed region in Fig. 8.8.1, node 1 is among the two best nodes.

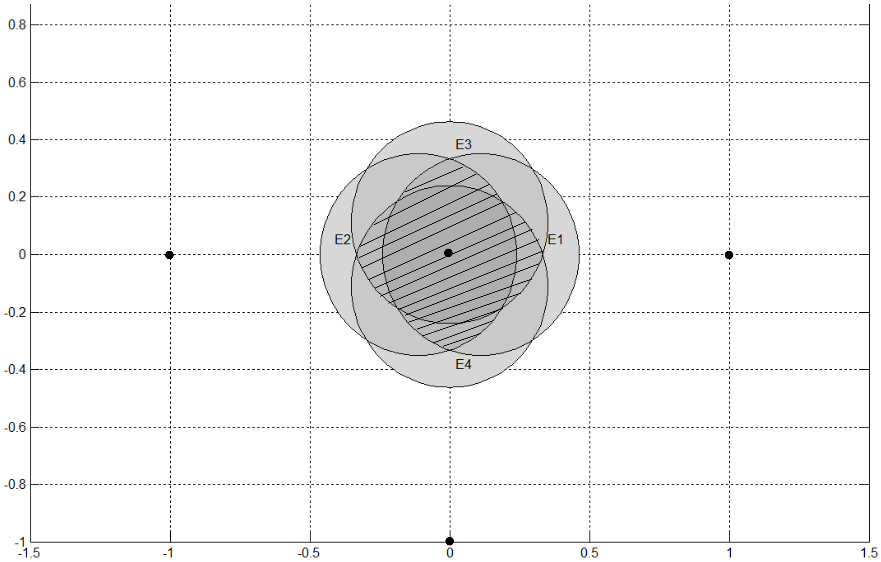


Figure 8.8.1: The region over which node 1 is better than others.

(c) The alternate route will be selected if the expected number of transmissions into and out of the malfunctioning node is 2 or greater. Let the packet dropping probability be p. The last successful transmission will have probability $(1-p)$. Then the expected number of transmissions is

$$1(1-p) + 2p(1-p) + 3p^2(1-p) + \cdots = 2$$

Solving above equality we have $\sum_{i=0}^{\infty}(i+1)p^i(1-p) = 1/(1-p) = 2$ and hence, $p = 0.5$.

(d) The probability of requiring less than or equal to 3 transmissions is

$$(1-p) + p(1-p) + p^2(1-p) = 0.99$$

This is a cubic in p and can be solved in any number of ways. A probability of 0.2 is close. Thus the delay requirement leads more quickly to choice of alternative paths in this example.

SOLUTION 8.9

Unknown mean in Gaussian noise The sample average as a mean estimator is unbiased with variance,

$$\begin{aligned}\hat{\theta} &= \frac{1}{N}\sum_{k=1}^{N} y_k \,,\\ E[\hat{\theta}] &= \theta \,,\\ \text{Var}(\hat{\theta}) &= \frac{\sigma^2}{N} \,.\end{aligned}$$

Furthermore, the second derivative of the log likelihood function does not include any stochastic terms, so the CRLB follows as

$$\begin{aligned}p(y_{1:N}|\theta) &= \frac{1}{(2\pi\sigma^2)^{N/2}} e^{\frac{-1}{2\sigma^2}\sum(y_k-\theta)^2} \,,\\ \frac{d^2 \log p(y_{1:N}|\theta)}{d\theta^2} &= \frac{N}{\sigma^2} \,,\\ \text{Var}(\hat{\theta}) &\geq \frac{\sigma^2}{N} \,.\end{aligned}$$

That is, the sample average is the minimum variance estimator and thus a consistent estimator. Further, the variance is equal to the CRLB, so it is also efficient.

SOLUTION 8.10

Moments method
We have

$$\begin{aligned}
\mu_1 =& E(y) = 0\,,\\
\mu_2 =& E(y^2) = \alpha\sigma_1^2 + (1-\alpha)\sigma_2^2\,,\\
\hat{\mu}_2 =& \frac{1}{N}\sum_{k=1}^{N} y_k^2\,,\\
\hat{\alpha} =& \frac{\hat{\mu}_2 - \sigma_2^2}{\sigma_1^2 - \sigma_2^2}.
\end{aligned}$$

In unknown variances case, we need more equations. Since all odd moments are zero at least the following even moments are needed:

$$\begin{aligned}
\mu_4 =& E(y^4) = 3\alpha\sigma_1^4 + 3(1-\alpha)\sigma_2^4 + \alpha(1-\alpha)\sigma_1^2\sigma_2^2\,,\\
\mu_6 =& E(y^6) = 15\alpha\sigma_1^6 + 15(1-\alpha)\sigma_2^6 + 3\alpha(1-\alpha)\sigma_1^4\sigma_2^2 + 3\alpha(1-\alpha)\sigma_1^2\sigma_2^2\,.
\end{aligned}$$

9 Positioning and Localization

SOLUTION 9.1

Timing Offset and GPS.

(a) Denote the unknown position as (x, y). (x, y) may be found by solving the following nonlinear system of equations

$$\begin{aligned}(x-x_1)^2+(y-y_1)^2 &= r_1^2 = 9\\ (x-x_2)^2+(y-y_2)^2 &= r_2^2 = 16\\ (x-x_3)^2+(y-y_3)^2 &= r_3^2 = 25\end{aligned}$$

The solution is

$$x = \frac{\begin{bmatrix} x_1^2-r_1^2 & x_2^2-r_2^2 & x_3^2-r_3^2 \end{bmatrix} \cdot \begin{bmatrix} (y_3-y_2)\\ (y_1-y_3)\\ (y_2-y_1)\end{bmatrix} + (y_3-y_2)(y_1-y_3)(y_2-y_1)}{x_1(y_3-y_2)+x_2(y_1-y_3)+x_3(y_2-y_1)} = 0;$$

$$y = \frac{\begin{bmatrix} y_1^2-r_1^2 & y_2^2-r_2^2 & y_3^2-r_3^2 \end{bmatrix} \cdot \begin{bmatrix} (x_3-x_2)\\ (x_1-x_3)\\ (x_2-x_1)\end{bmatrix} + (x_3-x_2)(x_1-x_3)(x_2-x_1)}{y_1(x_3-x_2)+y_2(x_1-x_3)+y_3(x_2-x_1)} = 0.$$

(b) Notice that while points A and B's measured ranges intersect at point D, point C's measured range cannot go through that point. This discrepancy indicates that there is a measurement error. Since any measurement error or offset has in this case been assumed to affect all measurements, we should look for a single correction factor that would allow all the measurements to intersect at one point. In our example, one discovers that by subtracting 0.5 meter from each measurement the ranges would all intersect at one point. After finding that correction factor, the receiver can then apply the correction to all measurements.

SOLUTION 9.2

Linearizing GPS Equations. Expanding (9.2), we have

$$\begin{cases} (0-x_1)^2+(0-y_1)^2+(0-z_1)^2+2\begin{bmatrix}(0-x_1) & (0-y_1) & (0-z_1)\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix} = d_1^2\\ (0-x_2)^2+(0-y_2)^2+(0-z_2)^2+2\begin{bmatrix}(0-x_2) & (0-y_2) & (0-z_2)\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix} = d_2^2\\ (0-x_3)^2+(0-y_3)^2+(0-z_3)^2+2\begin{bmatrix}(0-x_3) & (0-y_3) & (0-z_3)\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix} = d_3^2 \end{cases}$$

or

$$\begin{cases} x_1^2+y_1^2+z_1^2-d_1^2 = 2\begin{bmatrix}x_1 & y_1 & z_1\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix}\\ x_2^2+y_2^2+z_2^2-d_2^2 = 2\begin{bmatrix}x_2 & y_2 & z_2\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix}\\ x_3^2+y_3^2+z_3^2-d_3^2 = 2\begin{bmatrix}x_3 & y_3 & z_3\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix} \end{cases}$$

which can be reorganized into

$$2\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & y_3 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_1^2+y_1^2+z_1^2-d_1^2 \\ x_2^2+y_2^2+z_2^2-d_2^2 \\ x_3^2+y_3^2+z_3^2-d_3^2 \end{bmatrix}.$$

SOLUTION 9.3

Averaging to reduce error in TOA

(a) The relationship between time of arrival t and range r is $r = ct$. Therefore, the mean of the range is $\bar{r} = c \cdot \bar{t}$, and the variance of the range is $\delta_r = c^2 \cdot \delta_t$.

(b) Again, $r(n) = ct(n)$, $n = 1, 2, 3 \ldots$. Therefore, the estimated range based on the multiple measurements of time of arrival is

$$\hat{r} = \lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} r(n) = \lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} c\cdot t(n) = \lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} c\cdot(\bar{t}+m(n))$$
$$= \lim_{N\to\infty} c\cdot\bar{t} + c\cdot\frac{1}{N}\sum_{n=1}^{N} m(n) = c\cdot\bar{t}.$$

SOLUTION 9.4

Weighted centroid computation The weighted centroid of the three known locations is

$$\bar{r} = \frac{1}{1.2+1.5+1.7}(1.2a+1.5b+1.7c) = (0.22{,}0.32).$$

SOLUTION 9.5

Collaborative multilateration Clearly $u_0 = (0,1)$ and $v_0 = (0,-1)$. The squared distances from U to nodes A, C, and V are respectively 0.5, 2.5, and 2, while the squared distances from V to B, D, and U are 2.5, 0.5, and 2. Then for the first calculation we have $r_A = r_B = 1$ and $r_v = 2$ so that

$$A = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, z = \begin{bmatrix} 1-\sqrt{0.5} \\ 1-\sqrt{2.5} \\ 2-\sqrt{2} \end{bmatrix}.$$

resulting in the system $A^T A d_u = A^T z$:

$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \delta_{xu} \\ \delta_{yu} \end{bmatrix} = \begin{bmatrix} 0.8740 \\ -0.5859 \end{bmatrix},$$

which gives $u_1 = (0+0.437, 1-0.5859) = (0.437, 0.414)$. In the next set, $r_C = r_D = 1$, $r_U = 1.48$, so that

$$A = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0.295 & 0.995 \end{bmatrix}, z = \begin{bmatrix} 1-\sqrt{2.5} \\ 1-\sqrt{0.5} \\ 1.48-\sqrt{2} \end{bmatrix}.$$

The result is $v_1 = (0-0.437, -1+0.196) = (-0.464, -0.804)$. Iterations can continue now for u_2 using A, B, and n_1 and so forth. Successive iterations produce results closer to the true ones. In general, for collaborative multilateration to converge a variety of constraints on the topology must be satisfied, and the order of the computations is important. However, if there is a relatively high density of nodes with known position these constraints are almost always satisfied without explicit checking being required; bad positions can be discarded through recognition that the values are diverging in some neighborhood.

SOLUTION 9.6

Linearization of angle of arrival (AOA) location determination From $r_0 = [\ 0 \quad 0\ ]^T$, $r_1 = [\ -3 \quad 4\ ]^T, r_2 = [\ 4 \quad 3\ ]^T$, we have

$$\begin{aligned}
&\sin\theta_{01} = \tfrac{4}{5}, \cos\theta_{01} = -\tfrac{3}{5}, d_{01} = 5,\\
&\sin\theta_{02} = \tfrac{3}{5}, \cos\theta_{02} = \tfrac{4}{5}, d_{02} = 5,\\
&f_1(r_0) = \arctan(\tfrac{-4}{3}) = -0.9273,\\
&f_2(r_0) = \arctan(\tfrac{3}{4}) = 0.6435,
\end{aligned}$$

and

$$G = \begin{bmatrix} (y_0 - y_1)/d_{01}^2 & -(x_0 - x_1)/d_{01}^2 \\ (y_0 - y_2)/d_{02}^2 & -(x_0 - x_2)/d_{02}^2 \end{bmatrix} = \begin{bmatrix} -\frac{4}{25} & -\frac{3}{25} \\ -\frac{3}{25} & \frac{4}{25} \end{bmatrix}.$$

According to (9.6), the estimate of the unknown position is

$$\begin{aligned}
&\hat{r} = r_0 + G^{-1}N^{-1} \begin{bmatrix} \theta_1 - f_1(r_0) \\ \theta_2 - f_2(r_0) \end{bmatrix}\\
&= \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{4}{25} & -\frac{3}{25} \\ -\frac{3}{25} & \frac{4}{25} \end{bmatrix}^{-1} \begin{bmatrix} 0.7854 + 0.9273 \\ 2.3562 - 0.6435 \end{bmatrix}\\
&= \begin{bmatrix} 2.5641 \\ 10.8537 \end{bmatrix}.
\end{aligned}$$

It can be seen that here the linearization error is large. In practice, a series of iterations would need to be performed.

10 Time Synchronization

SOLUTION 10.1

TOA with low-cost clocks Denote the satellite positions by $(X_i, Y_i, Z_i), i = 1, 2, 3, 4$. The user's unknown position is (U_x, U_y, U_z). With four range measurements, the nonlinear system of equations for positioning is

$$\begin{cases} (X_1 - U_x)^2 + (Y_1 - U_y)^2 + (Z_1 - U_z)^2 = c^2(t_1 - \Delta t)^2 \\ (X_2 - U_x)^2 + (Y_2 - U_y)^2 + (Z_2 - U_z)^2 = c^2(t_2 - \Delta t)^2 \\ (X_3 - U_x)^2 + (Y_3 - U_y)^2 + (Z_3 - U_z)^2 = c^2(t_3 - \Delta t)^2 \\ (X_4 - U_x)^2 + (Y_4 - U_y)^2 + (Z_4 - U_z)^2 = c^2(t_4 - \Delta t)^2 \end{cases}$$

where c is the speed of light, are respectively the true time of arrival from the four satellites. $t_i, i = 1, 2, 3, 4$, is the unknown clock drift in the receiver. Since we have four equations, the four unknown parameters can be found by solving the above system of equations. In practice, this is done with a linearized form of the problem.

SOLUTION 10.2

Time difference of arrival (TDOA) in a two-dimensional space

(a) according to lecture notes, $w = A^+ b$ and we have

$$\begin{aligned} s_1^2 &= x_t^2 + y_t^2, \\ r_2^2 &= x_2^2 + y_2^2 = 2, \\ r_3^2 &= x_3^2 + y_3^2 = 1, \\ r_4^2 &= x_4^2 + y_4^2 = 10, \\ r_5^2 &= x_5^2 + y_5^2 = 17, \end{aligned}$$

and

$$A = \begin{bmatrix} x_2 & y_2 & -t_{12} & t_{12}^2/2 \\ x_3 & y_3 & -t_{13} & t_{13}^2/2 \\ x_4 & y_4 & -t_{14} & t_{14}^2/2 \\ x_5 & y_5 & -t_{15} & t_{15}^2/2 \end{bmatrix} = \begin{bmatrix} -1 & -1 & 1.4 & 1.4^2/2 \\ 0 & 1 & -0.4 & 0.4^2/2 \\ 3 & 1 & 1.6 & 1.6^2/2 \\ 1 & 4 & 2.6 & 2.6^2/2 \end{bmatrix},$$

$$w = \begin{bmatrix} x_t \\ y_t \\ vs_1 \\ v^2 \end{bmatrix}, b = 1/2 \begin{bmatrix} r_2^2 \\ r_3^2 \\ r_4^2 \\ r_5^2 \end{bmatrix} = 1/2 \begin{bmatrix} 2 \\ 1 \\ 10 \\ 17 \end{bmatrix}.$$

The least squares solution is then

$$w = (\mathrm{A}^T \mathrm{A})^{-1} A^T b = \begin{bmatrix} 0.8750 \\ -0.1250 \\ -0.9375 \\ 3.1250 \end{bmatrix}.$$

Therefore, the unknown position is $\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} 0.8750 \\ -0.1250 \end{bmatrix}$.

(b) According to lecture notes,

$$A = \begin{bmatrix} x_2 & y_2 & -t_{12} \\ x_3 & y_3 & -t_{13} \\ x_4 & y_4 & -t_{14} \\ x_5 & y_5 & -t_{15} \end{bmatrix} = \begin{bmatrix} -1 & -1 & 1.4 \\ 0 & 1 & -0.4 \\ 3 & 1 & 1.6 \\ 1 & 4 & 2.6 \end{bmatrix},$$

$$w = \begin{bmatrix} x_t \\ y_t \\ vs_1 \end{bmatrix}, \; b = 1/2 \begin{bmatrix} r_2^2 \\ r_3^2 \\ r_4^2 \\ r_5^2 \end{bmatrix} = 1/2 \begin{bmatrix} 2 \\ 1 \\ 10 \\ 17 \end{bmatrix},$$

$$d = -1/2 \begin{bmatrix} t_{12}^2 \\ t_{13}^2 \\ t_{14}^2 \\ t_{15}^2 \end{bmatrix} = -1/2 \begin{bmatrix} 1.4^2 \\ 0.4^2 \\ 1.6^2 \\ 2.6^2 \end{bmatrix},$$

$$p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = (\mathrm{A}^T\mathrm{A})^{-1} A^T b,$$

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = (\mathrm{A}^T\mathrm{A})^{-1} A^T d,$$

$$w = \begin{bmatrix} x_t \\ y_t \\ vs_1 \end{bmatrix} = p + v^2 q = \begin{bmatrix} 0.8905 \\ -0.1645 \\ -1.0328 \end{bmatrix}$$

SOLUTION 10.3

TDOA in a three-dimensional space

(a) according to lecture notes, $w = A^+ b$ and we have

$$\begin{aligned} s_1^2 &= x_t^2 + y_t^2 + z_t^2, \\ r_2^2 &= x_2^2 + y_2^2 + z_2^2 = 9, \\ r_3^2 &= x_3^2 + y_3^2 + z_3^2 = 36, \\ r_4^2 &= x_4^2 + y_4^2 + z_4^2 = 25, \\ r_5^2 &= x_5^2 + y_5^2 + z_5^2 = 25, \\ r_6^2 &= x_6^2 + y_6^2 + z_6^2 = 64. \end{aligned}$$

and

$$A = \begin{bmatrix} x_2 & y_2 & z_2 & -t_{12} & t_{12}^2/2 \\ x_3 & y_3 & z_3 & -t_{13} & t_{13}^2/2 \\ x_4 & y_4 & z_4 & -t_{14} & t_{14}^2/2 \\ x_5 & y_5 & z_5 & -t_{15} & t_{15}^2/2 \\ x_6 & y_6 & z_6 & -t_{16} & t_{16}^2/2 \end{bmatrix} = \begin{bmatrix} 0 & 3 & 0 & 0 & 0 \\ 6 & 0 & 0 & -1 & 1/2 \\ -\;3 & 4 & 0 & -0.7 & 0.7^2/2 \\ -4 & -3 & 0 & -0.7 & 0.7^2/2 \\ 0 & 0 & -8 & -1.7 & 1.7^2/2 \end{bmatrix},$$

$$w = \begin{bmatrix} x_t \\ y_t \\ z_t \\ vs_1 \\ v^2 \end{bmatrix}, \; b = 1/2 \begin{bmatrix} r_2^2 \\ r_3^2 \\ r_4^2 \\ r_5^2 \\ r_6^2 \end{bmatrix} = 1/2 \begin{bmatrix} 9 \\ 36 \\ 25 \\ 25 \\ 64 \end{bmatrix}.$$

The least squares solution is then

$$w = (\mathrm{A}^T\mathrm{A})^{-1} A^T b = \begin{bmatrix} -1.5000 \\ 1.5000 \\ 7.3333 \\ 10.6190 \\ 75.2381 \end{bmatrix}.$$

Therefore, the unknown position is

$$\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} -1.5000 \\ 1.5000 \\ 7.3333 \end{bmatrix}.$$

(b) According to lecture notes,

$$A = \begin{bmatrix} x_2 & y_2 & z_2 & -t_{12} \\ x_3 & y_3 & z_3 & -t_{13} \\ x_4 & y_4 & z_4 & -t_{14} \\ x_5 & y_5 & z_5 & -t_{15} \\ x_6 & y_6 & z_6 & -t_{16} \end{bmatrix} = \begin{bmatrix} 0 & 3 & 0 & 0 \\ 6 & 0 & 0 & -1 \\ 3 & 4 & 0 & -0.7 \\ -4 & -3 & 0 & -0.7 \\ 0 & 0 & 8 & -1.7 \end{bmatrix},$$

$$w = \begin{bmatrix} x_t \\ y_t \\ z_t \\ vs_1 \end{bmatrix}, \ b = 1/2 \begin{bmatrix} r_2^2 \\ r_3^2 \\ r_4^2 \\ r_5^2 \\ r_6^2 \end{bmatrix} = 1/2 \begin{bmatrix} 9 \\ 36 \\ 25 \\ 25 \\ 64 \end{bmatrix},$$

$$d = -1/2 \begin{bmatrix} t_{12}^2 \\ t_{13}^2 \\ t_{14}^2 \\ t_{15}^2 \\ t_{16}^2 \end{bmatrix} = -1/2 \begin{bmatrix} 0 \\ 1 \\ 0.7^2 \\ 0.7^2 \\ 1.7^2 \end{bmatrix},$$

$$p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = (\mathrm{A}^T\mathrm{A})^{-1}A^T b,$$

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = (\mathrm{A}^T\mathrm{A})^{-1}A^T d,$$

$$w = \begin{bmatrix} x_t \\ y_t \\ z_t \\ vs_1 \end{bmatrix} = p + v^2 q = \begin{bmatrix} -1.5000 \\ 1.5000 \\ 7.3333 \\ 10.6190 \end{bmatrix}.$$

Therefore, the unknown position is

$$\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} -1.5000 \\ 1.5000 \\ 7.3333 \end{bmatrix}.$$

SOLUTION 10.4

Ex.9.3 in [3]

Comparing the two clocks, the clock offset is the difference in time between the two clocks. In this example, the current clock offset is 100. The clock rate indicates the frequency at which a clock progresses, i.e., node A's clock has a clock rate of 1.01 and node B's clock has a clock rate of 0.99. The clock skew indicates the difference in the frequencies of the two clocks, which is 0.02. Clock A is fast since its clock readings progress faster than real time. Similarly, clock B is slow since its clock readings progress slower than real time.

SOLUTION 10.5

Ex.9.4 in [3]

Each clock can deviate from real time by 100 μs per second in the worst case, *i.e.* it takes up to 10000 s to reach an offset of 1 s. However, since both clocks have a drift rate of 100 ppm, the relative offset between them can be twice as large as the offset between a single clock and the external reference clock. Therefore, the necessary re-synchronization interval is 5000 s.

SOLUTION 10.6

Ex.9.6 in [3]

Since the error can go either way, i.e., a clock can be faster or slower than the external reference time by the amount of the error, the maximum precision is then the sum of the two largest errors, i.e. $3 + 4 = 7$.

SOLUTION 10.7

Ex.9.7 in [3]

1. If $t_1 = 3150$, $t_2 = 3120$, and $t_3 = 3250$, then the offset can be calculated as

$$\text{offset} = \frac{(\mathrm{t}_2 - \mathrm{t}_1) - (\mathrm{t}_3 - \mathrm{t}_2)}{2} = -80\,,$$

   That is, the two clocks differ by 80 time units.

2. Node A's clock is going too fast compared to node B's clock.

3. One approach is to simply reset the clock by 80 time units. However, this can lead to problems since the clock repeats the last 80 time units, potentially triggering events in the node that have already been triggered previously. Therefore, node A should slow down the progress of its clock until clock B had an opportunity to catch up the 80 time units it lags behind node A's clock.

SOLUTION 10.8

Ex.9.8 in [3]

As indicate in Figure 10.8.1, the times for the synchronization messages to travel between nodes can differ, e.g., based on the distances between senders and receivers. Besides propagation delays, synchronization messages also experience send, access, and receive delays that can differ from node to node, affecting the measured offsets.

11 Networked Control Systems

SOLUTION 11.1

Matrix Exponential The exponential of A, denote by e^A or $\exp(A)$, is the $n \times n$ matrix given by the power series

$$e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k .$$

The above series always converges, so the exponential of A is well defined. Note that if A is a 1×1 matrix, the matrix exponential of A is a 1×1 matrix consisting of the ordinary exponential of the signal element of A. Thus we have that

$$e^A = I + A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} ,$$

since $A * A = 0$.

We can find e^A via Laplace transform as well. As we know that the solution to the system linear differential equations given by

$$\frac{d}{dt} y(t) = Ay(t) , \quad y(0) = y_0 ,$$

is

$$y(t) = e^{At} y_0 .$$

Using the Laplace transform, letting $Y(s) = \mathcal{L}\{y\}$, and applying to the differential equation we get

$$sY(s) - y_0 = AY(s) \Rightarrow (sI - A)Y(s) = y_0 ,$$

where I is the identity matrix. Therefore,

$$y(t) = \mathcal{L}^{-1}\{(sI - A)^{-1}\} y_0 .$$

Thus, it can be concluded that

$$e^{At} = \mathcal{L}^{-1}\{(sI - A)^{-1}\} ,$$

from this we can find e^A by setting $t = 1$. Thus we can have

$$e^{At} = \mathcal{L}^{-1}\{(sI - A)^{-1}\} = \mathcal{L}^{-1} \left\{ \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2} \\ 0 & \frac{1}{s} \end{bmatrix} \right\} = \begin{bmatrix} u(t) & tu(t) \\ 0 & u(t) \end{bmatrix} .$$

We obtain the same result as before if we insert $t = 1$ into previous equation.

SOLUTION 11.2

Stability The eigenvalue equations for a matrix Φ is

$$\Phi v - \lambda v = 0 ,$$

which is equivalent to

$$(\Phi - \lambda I)v = 0 ,$$

where I is the $n \times n$ identity matrix. It is a fundamental result of linear algebra that an equation $Mv = 0$ has a non-zero solution v if and only if the determinant $\det(M)$ of the matrix M is zero. It follows that the eigenvalues of Φ are precisely the real numbers λ that satisfy the equation

$$\det(\Phi - \lambda I) = 0 .$$

The left-hand side of this equation can be seen to be a polynomial function of variable λ. The degree of this polynomial is n, the order of the matrix. Its coefficients depend on the entries of Φ, except that its term of degree n is always $(-1)^n\lambda^n$. For example, let Φ be the matrix

$$\Phi = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 4 \\ 0 & 4 & 9 \end{bmatrix} .$$

The characteristic polynomial of Φ is

$$\det(\Phi - \lambda I) = \det\left(\begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 4 \\ 0 & 4 & 9 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) = \det \begin{bmatrix} 2-\lambda & 0 & 0 \\ 0 & 3-\lambda & 4 \\ 0 & 0 & 9-\lambda \end{bmatrix} ,$$

which is

$$(2-\lambda)[(3-\lambda)(9-\lambda)-16] = 22 - 35\lambda + 14\lambda^2 - \lambda^3 .$$

The roots of this polynomial are 2, 1, and 11. Indeed these are the only three eigenvalues of Φ, corresponding to the eigenvectors $[1,0,0]'$, $[0,2,-1]'$, and $[0,1,2]'$.

Given the matrix $\Phi = \text{diag}([-1.01, 1, -0.99])$, we plot following image, in which we can distinguish stable, asymptotical stable and instable state.

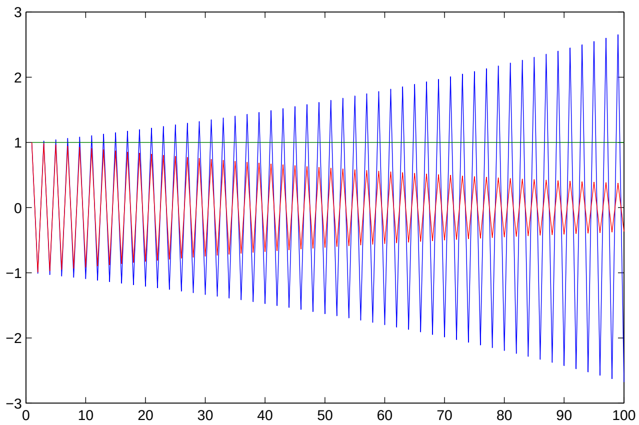


Figure 11.2.1: The stability, asymptotical stability and instability.

SOLUTION 11.3

Modeling

The dynamic for the state vector using Cartesian velocity, $(x, y, v_x, v_y, \omega)^T$, is given by:

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{v_x} &= -\omega v_y \\ \dot{v_y} &= \omega v_x \\ \dot{\omega} &= 0 . \end{aligned}$$

Secondly with state vector with polar velocity, $(x, y, v, h, \omega)^T$, the state dynamics become

$$\begin{aligned} \dot{x} &= v\cos(h) \\ \dot{y} &= v\sin(h) \\ \dot{v} &= 0 \\ \dot{h} &= \omega \\ \dot{\omega} &= 0\,. \end{aligned}$$

SOLUTION 11.4

Linearized Discretization

Consider the tracking example with $(x, y, v, h, \omega)^T$. The analytical solution is

$$\begin{aligned} x(t+T) &= x(t) + \frac{2v(t)}{\omega(t)}\sin\left(\frac{\omega(t)T}{2}\right)\cos\left(h(t) + \frac{\omega(t)T}{2}\right) \\ y(t+T) &= y(t) + \frac{2v(t)}{\omega(t)}\sin\left(\frac{\omega(t)T}{2}\right)\sin\left(h(t) + \frac{\omega(t)T}{2}\right) \\ v(t+T) &= v(t) \\ h(t+T) &= h(t) + \omega(t)T \\ \omega(t+T) &= \omega(t)\,. \end{aligned}$$

The alternative state coordinates $(x, y, v_x, v_y, \omega)^T$ give

$$\begin{aligned} x(t+T) &= x(t) + \frac{v_x(t)}{\omega(t)}\sin(\omega(t)T) - \frac{v_y(t)}{\omega(t)}(1-\cos(\omega(t)T)) \\ y(t+T) &= y(t) + \frac{v_x(t)}{\omega(t)}(1-\cos(\omega(t)T)) + \frac{v_y(t)}{\omega(t)}\sin(\omega(t)T) \\ v_x(t+T) &= v_x(t)\cos(\omega(t)T) - v_y(t)\sin(\omega(t)T) \\ v_y(t+T) &= v_x(t)\sin(\omega(t)T) + v_y(t)\cos(\omega(t)T) \\ \omega(t+T) &= \omega(t)\,. \end{aligned}$$

SOLUTION 11.5

Modeling of the Temperature Control

1. Let $X(t) = [T_i(t), T_r(t)]^T$ and $y(t) = T_i(t)$. The continuous time state space model can be obtained as

$$\begin{aligned} \dot{X}(t) &= \begin{bmatrix} -\alpha_1 - \alpha_2 & \alpha_1 \\ 0 & -\alpha_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha_3 \end{bmatrix} u \\ y(t) &= [1, 0]X(t)\,. \end{aligned}$$

2. In the discrete time domain, we need to find matrices Φ and Γ.

$$\begin{aligned} \Phi =& e^{Ah} \\ \Gamma =& \int_0^h e^{As} ds B\,. \end{aligned}$$

In this case, by using the result from previous subproblem, we can obtain that

$$
\begin{aligned}
\Phi =& \mathcal{L}^{-1}\{(sI-A)^{-1}\} = \begin{bmatrix} e^{-\alpha_1 h-\alpha_2 h} & \alpha_1(\alpha_1+\alpha_2-\alpha_3)(e^{-\alpha_3 h}-e^{-\alpha_1 h-\alpha_2 h}) \\ 0 & e^{-\alpha_3 h} \end{bmatrix} \\
\Gamma =& \int_0^h e^{As} ds B = \begin{bmatrix} \frac{\alpha_1\alpha 3\left(e^{-\alpha_1 h-\alpha_2 h}-1\right)}{(\alpha_1+\alpha_2)(\alpha_1+\alpha_2-\alpha_3)} - \frac{\alpha_1\left(e^{-\alpha_3 h}\right)}{\alpha_1+\alpha_2-\alpha_3} \\ 1-e^{-\alpha_3 h} \end{bmatrix}
\end{aligned}
$$

SOLUTION 11.6

PID Controller From the step response plot, we can find that $T_G \approx 4$. Then the parameters of PID can be obtained directly as $K_p = 6$, $K_i = 3$ and $K_d = 3$.

SOLUTION 11.7

Stability of Networked Control Systems with Network-induced Delay The are two sources of delay in the network, the sensor to controller τ_{sc} and controller to actuator τ_{ca}. The the control low is fixed. Therefore, these delays can be lumped together for analysis purposed: $\tau = \tau_{sc} + \tau_{ca}$.

Since $\tau < h$, at most two controllers samples need be applied during the k-th sampling period: $u((k-1)h)$ and $u(kh)$. The dynamical system can be rewritten as

$$
\begin{aligned}
&\dot{x}(t) = Ax(t) + Bu(t), \qquad t \in [kh+\tau, (k+1)h+\tau) \\
&y(t) = Cx(t)\,, \\
&u(t^+) = -Kx(t-\tau), \qquad t \in \{kh+\tau, \quad k = 0, 1, 2, \ldots\}
\end{aligned}
$$

where $u(t^+)$ is a piecewise continuous and changes values only at $kh+\tau$. By sampling the system with period h, we obtain

$$
\begin{aligned}
&x((k+1h)) = \Phi x(kh) + \Gamma_0(\tau)u(kh) + \Gamma_1(\tau)u((k-1)h) \\
&y(hk) = Cx(kh)\,,
\end{aligned}
$$

where

$$
\begin{aligned}
&\Phi = e^{Ah}\,, \\
&\Gamma_0(\tau)u(kh) = \int_0^{h-\tau} e^{As}Bds\,, \\
&\Gamma_1(\tau)u((k-1)h) = \int_{h-\tau}^{h} e^{As}Bds\,.
\end{aligned}
$$

Let $z(kh) = [x^T(kh), u^T((k-1)h)]^T$ be the augmented state vector, then the augmented closed loop system is

$$
z((k+1)h) = \tilde{\Phi} z(kh)\,,
$$

where

$$
\tilde{\Phi} = \begin{bmatrix} \Phi - \Gamma_0(\tau)K & \Gamma_1(\tau) \\ -K & 0 \end{bmatrix}.
$$

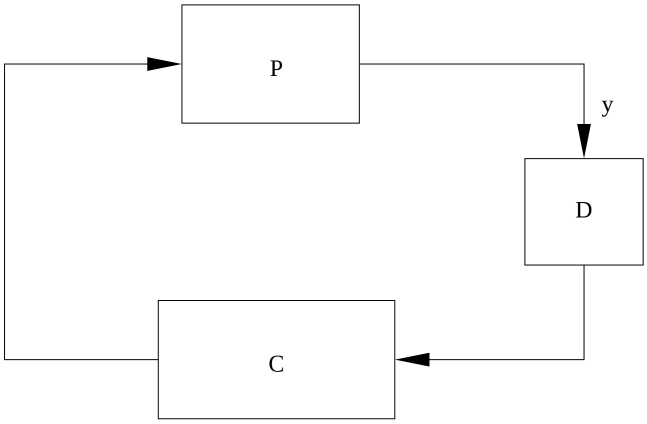


Figure 11.8.1: Closed loop system for Problem 11.3.

Given that $A = 0$ and $B = I$, we have

$$\tilde{\Phi} = \begin{bmatrix} 1 - hK + \tau K & \tau \\ -K & 0 \end{bmatrix}.$$

The characteristic polynomial of this matrix is

$$\lambda^2 - (1 - Kh + \tau K)\lambda + \tau K\,.$$

By recalling that $h = 1/K$, we define $y = \tau/h$, so it follows that the characteristic polynomial is

$$\lambda^2 - y\lambda + y\,.$$

The solutions λ_1 and λ_2 to such an equation are

$$\lambda_1 = \frac{y}{2} + j\frac{1}{\sqrt{4y - y^2}}\,,$$
$$\lambda_2 = \frac{y}{2} - j\frac{1}{\sqrt{4y - y^2}}\,.$$

Since $|\lambda| < 1$, there is no other constraint for λ

SOLUTION 11.8

The control problem over wireless sensor network can be represented as shown Figure 11.8.1. The delay Δ is such that

$$\Delta(y(kh)) = y(kh - d(k)) \qquad d(k) \in \{0, \ldots, N\}$$

The closed loop system is stable if

$$\left|\frac{P(e^{i\omega})C(e^{i\omega})}{1 + P(e^{i\omega})C(e^{i\omega})}\right| < \frac{1}{N|e^{i\omega} - 1|} \qquad \omega \in [0, 2\pi]$$

where N is the number of samples that the control signal is delayed. Notice that the previous result is valid if the closed loop transfer function is stable. In this case the closed loop transfer function is stable with poles

$$z_1 = 0.861, \quad z_2 = 0.447.$$

If we plot the bode diagram of the closed loop system without delay versus the function $1/(N|e^{i\omega} - 1|)$ for different values of N we obtain the results shown in Figure 11.8.2. It can be seen that the closed loop system is stable if $N \leq 3$. Thus the maximum delay is 3 samples. Notice that the result is only a sufficient condition. This means that it might be possible that the system is stable for larger delays than 3 samples.

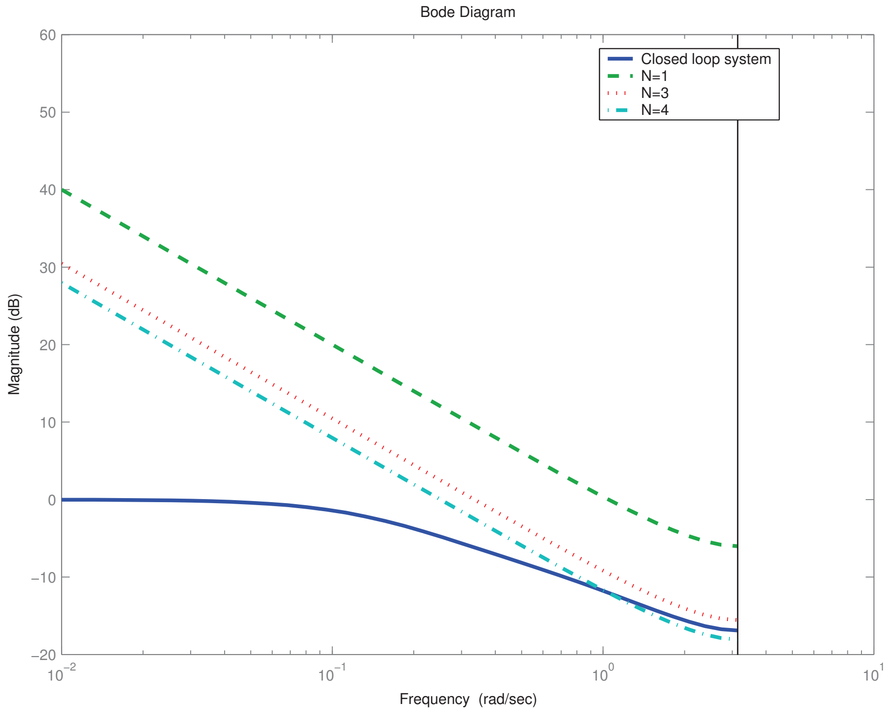


Figure 11.8.2: Bode diagram of the closed loop system of Problem 11.2 and the function $1/(N|e^{i\omega} - 1|)$ for different values of N.

SOLUTION 11.9

Stability of Networked Control Systems with Packet Losses

We use the following result to study the stability of the system:

Theorem 11.1. *Consider the system given in Figure 11.9.1. Suppose that the closed-loop system without packet losses is stable. Then*

- *if the open-loop system is marginally stable, then the system is exponentially stable for all* $0 < r \leq 1$.
- *if the open-loop system is unstable, then the system is exponentially stable for all*

$$\frac{1}{1 - \gamma_1/\gamma_2} < r \leq 1\,,$$

where $\gamma_1 = \log[\lambda^2_{\max}(\Phi - \Gamma K)]$, $\gamma_2 = \log[\lambda^2_{\max}(\Phi)]$

By sampling the system with period $h = 0.3$ we obtain:

$$\Phi = \begin{bmatrix} 1.3499 & 0.3045 \\ 0 & 0.7408 \end{bmatrix}.$$

and

$$\Gamma K = \begin{bmatrix} 0.0907 & 0.0408 \\ 0.5184 & 0.2333 \end{bmatrix}.$$

It follows that the closed loop system is stable (the matrix $\Phi - \Gamma K$ is stable), but the open loop system is not stable (Φ has the maximum eigenvalue leger than 1). The second statement of the Theorem applies. We have $\gamma_1 = -0.1011$, and $\gamma_2 = 0.6001$. It follows that $r \geq 0.85$, namely that the system can tolerate a packet loss of up about 15%.

SOLUTION 11.10

Networked Control System

(a) Since $\tau < h$, at most two controllers samples need be applied during the k-th sampling period: $u((k-1)h)$ and $u(kh)$. The dynamical system can be rewritten as

$$\begin{aligned}
&\dot{x}(t) = Ax(t) + Bu(t), \qquad t \in [kh+\tau, (k+1)h+\tau) \\
&y(t) = Cx(t)\,, \\
&u(t^+) = -Kx(t-\tau), \qquad t \in \{kh+\tau, \quad k = 0,1,2,\ldots\}
\end{aligned}$$

where $u(t^+)$ is a piecewise continuous and changes values only at $kh+\tau$. By sampling the system with period h, we obtain

$$\begin{aligned}
&x((k+1)h) = \Phi x(kh) + \Gamma_0(\tau)u(kh) + \Gamma_1(\tau)u((k-1)h) \\
&y(hk) = Cx(kh)\,,
\end{aligned}$$

where

$$\begin{aligned}
\Phi &= e^{Ah} = e^{ah}\,, \\
\Gamma_0(\tau) &= \int_0^{h-\tau} e^{As}Bds = \frac{1}{a}\left(e^{a(h-\tau)} - 1\right)\,, \\
\Gamma_1(\tau) &= \int_{h-\tau}^{h} e^{As}Bds = \frac{1}{a}\left(e^{ah} - e^{a(h-\tau)}\right)\,.
\end{aligned}$$

given that $A = a, B = 1, C = 1$.

(b) Let $z(kh) = [x^T(kh), u^T((k-1)h)]^T$ be the augmented state vector, then the augmented closed loop system is

$$z((k+1)h) = \tilde{\Phi}z(kh)\,,$$

where

$$\tilde{\Phi} = \begin{bmatrix} \Phi - \Gamma_0(\tau)K & \Gamma_1(\tau) \\ -K & 0 \end{bmatrix}\,.$$

Using the results obtained in (a), we can obtain

$$\tilde{\Phi} = \begin{bmatrix} e^{ah} - \frac{1}{a}\left(e^{a(h-\tau)} - 1\right)K & \frac{1}{a}\left(e^{ah} - e^{a(h-\tau)}\right) \\ -K & 0 \end{bmatrix}\,.$$

(c) The characteristic polynomial of this matrix is

$$\lambda^2 - \left(e^{ah} - \frac{1}{a}\left(e^{a(h-\tau)} - 1\right)\right)K + \frac{K}{a}\left(e^{ah} - e^{a(h-\tau)}\right)\,.$$

Thus when the $\max|\lambda| > 1$, the closed loop system becomes unstable.

(d) We use the following result to study the stability of the system:

Theorem 11.2. *Consider the system given in Fig. 2. Suppose that the closed-loop system without packet losses is stable. Then*

- *if the open-loop system is marginally stable, then the system is exponentially stable for all* $0 < r \leq 1$.
- *if the open-loop system is unstable, then the system is exponentially stable for all*

$$\frac{1}{1-\gamma_1/\gamma_2} < r \leq 1 \,,$$

where $\gamma_1 = \log[\lambda^2_{\max}(\Phi - \Gamma K)]$, $\gamma_2 = \log[\lambda^2_{\max}(\Phi)]$

Here we have

$$\Phi = e^{Ah} = e^{ah} \,,$$
$$\Gamma = \int_0^h e^{As} B ds = \frac{1}{a}\left(e^{ah} - 1\right) \,.$$

Thus, the stability of this system depends on the values of K, h, a. When the conditions are not satisfied, we may choose different K for controller or different sampling time h for the system to make the system stable.

SOLUTION 11.11

Energy-Efficient Control of NCS over IEEE 802.15.4 Networks

Practical-stability of each loop is ensured if the minimum inter-sampling time of the self-triggered sampler is greater than the the minimum beacon interval fixed to $15.36 \times 2 = 30.72$ ms. In other words, practical-stability of each loop is ensured if

(i)

$$\min_k \quad t_{i,k+1} - T_{i,k} \geq \mathrm{BI}_{\min} \,, \forall i = 1,2,3 \,.$$

The minimum inter-sampling time of the given self-triggered sampler, is attained by considering the peak value of the associated process output. By recalling the definition of $\mathcal{L}_\infty$-norm of a signal $s : \mathbb{R} \to \mathbb{R}^n$defined as $\|s(t)\|_{\mathcal{L}_\infty} = \sup_{t \geq t_0} \|s(t)\|$ (note that the $\mathcal{L}_\infty$-norm indicates the peak value of a signal), the minimum inter-sampling time guaranteed is given by

(ii).

$$\min_k \quad t_{i,k+1} - T_{i,k} = \frac{1}{|a_i|} \ln\left(1 + \frac{|a_i|\delta_i}{|a_i - b_i k_i| |x_i|_{\mathcal{L}_\infty}}\right) \,.$$

Under the defined sampling rule, the closed-loop system can be rewritten as $\dot{x}_i = (a_i - b_i k_i)x_i - b_i k_i e_i$, for all $t \in [T_{i,k}, T_{i,k+1})$. Because $a_i - b_i k_i < 0$ for all i, and because $|e_i| \leq \delta_i$ for all $t \geq t_0$, the output of each process is upper-bounded (recall BIBO stability) with

(iii).

$$|x_i(t)| \leq |x_{i,0}| e^{(a_i - b_i k_i)(t - t_0)} + \frac{|b_i k_i|}{|(a_i - b_i k_i)|}\delta_i \,,$$

for all $t \geq t_0$, and then

(iv).

$$|x_i|_{\mathcal{L}_\infty} \leq |x_{i,0}| + \frac{|b_i k_i|}{|a_i - b_i k_i|}\delta_i \,.$$

Thus, practical-stability of each loop is ensured if it holds

(v).

$$\frac{1}{|a_i|}\ln\left(1+\frac{|a_i|\delta_i}{|a_i-b_ik_i||x_i|_{\mathcal{L}_\infty}}\right)\geq \mathrm{BI}_{\min}\,,$$

where $|x_i|_{\mathcal{L}_\infty}$ can be estimated by previous equation. In this case an ultimate bound (or practical-stability region) is given by

$$\varepsilon_i=\frac{|b_ik_i|}{|a_i-b_ik_i|}\delta_i\,.$$

(a) The closed loop dynamics of the system #1 becomes, for $t\in[T_{1,k},T_{1,k+1})$, $\dot{x}_1=(2-k_1)x_1-k_1e_1$, from which one derives $k_1>2$. By observing that $a_1-b_1k_1<0, a_1>0, x_{1,0}>0$, we get

$$k_1\leq\frac{1}{b_1(x_{1,0}+\delta_1)}\left(\frac{a_1\delta_1}{(e^{a_1\mathrm{BI}_{\min}}-1)x_{1,0}}+a_1x_{1,0}\right)=4.687\,,$$

where the upper-bound (**iv**) is used. Then, for all $2<k_1\leq 4.68$ system #1 is practically-stable. The region of practical-stability is $0.88\leq\varepsilon_1<+\infty$, depending on the choice of the control. For instance, the practical-stability region size decreases as the control effort increases, but that way the inter-sampling times shrinks, leading to a larger energy expenditure of the network.

(b) By following the same argument as in the previous point, we get the condition

$$\delta_2\geq\frac{(e^{a_2\mathrm{BI}_{\min}}-1)(b_2k_2-a_2)x_{2,0}}{a_2-(e^{a_2\mathrm{BI}_{\min}}-1)b_2k_2}\simeq 2.955\,,$$

and then practical-stability is ensured by taking, for example, $\delta_2\geq 3$. In this case, the region of practical-stability is $\varepsilon_2\geq 12$, where the value 12 is obtained for $\delta_2=3$, and it increases as δ_2 does.

Notice that, in general, by increasing δ_i, we are enlarging the practical-stability region size, but we are also enlarging the inter-sampling times. Hence, it is clear the tradeoff between the closed-loop performance (practical-stability region size) and the energy efficiency of the network (inter-sampling times), tweaked by δ_i. Further notice that even for arbitrary large values of δ_i, condition (**v**) may not be fulfilled, and then a system may not be stabilizable over the specified IEEE 802.15.4 even if we are willing to accept large ultimate bound regions. This can be observed by looking at the argument of the logarithm in the inequality (**i**) that is bounded with respect to the variable δ_i when the upper-bound (**iv**) is used. Hence, even for $\delta_i\to+\infty$ condition(**v**) may not be fulfilled.

(c) By following the same argument as in the previous points, we get the condition

$$x_{3,0}\leq\frac{(a_3-(e^{a_3\mathrm{BI}_{\min}}-1)b_3k_3)\delta_3}{(e^{a_3\mathrm{BI}_{\min}}-1)(b_3k_3-a_3)}\simeq 14.281\,.$$

Hence, the system is practical-stabilizable over the specified IEEE 802.15.4 network if $|x_{3,0}|\leq 14.28$. The ultimate bound region size is $\varepsilon_3=1.5$. Notice how the ultimate bound region size does not dependent on the initial condition, but practical-stabilizability depends on that. This is due because the inter-sampling times depend on the distance of the current process output from the equilibrium point (that in this case is the origin). If the initial condition is far from the equilibrium point, the self-triggered sampler may give inter-sampling times that are shorter than the minimum beacon interval guaranteed by the protocol, and then practical-stability may not be achieved.

12 Scheduling

SOLUTION 12.1

Scheduling in Smart Grids We propose the problem

$$\max_{i,j,c} \quad \sum_{c=1}^{N}\sum_{i=1}^{N_{\mathrm{TTI}}}\sum_{j=1}^{N_{\mathrm{RB}}} R_{i,j}^{(c)} \tag{12.1a}$$

$$\text{s.t.} \quad \sum_{c} x_{i,j}^{(c)} \leq 1 \quad x_{i,j}^{(c)} \in \{0,1\} \qquad \forall i,j \tag{12.1b}$$

$$\sum_{i} x_{i,j}^{(c)} \leq N_{\mathrm{TTI}} \qquad \forall j,c \tag{12.1c}$$

$$\sum_{j} x_{i,j}^{(c)} \leq N_{\mathrm{RB}} \qquad \forall i,c \tag{12.1d}$$

$$\sum_{i}\sum_{j} x_{i,j}^{(c)} \leq L^{(c)} \qquad \forall c\,, \tag{12.1e}$$

where $R_{i,j}^{(c)} = \lambda_{i,j}^{(c)} x_{i,j}^{(c)}$ is the utility weight function for the c-th user with some utility parameters $\lambda_{i,j}^{(c)}$, $x_{i,j}^{(c)}$ is 1 if (i,j) resource block is assigned to c-th user, and zero otherwise. Eq. (12.1b) indicates that each resource block can be allocated to one user at most, and Eq. (12.1c) and (12.1d) give the greatest values N_{TTI}, N_{RB} in time and frequency domain respectively. Eq. (12.1e) indicates that resource blocks allocated to UE are limited by each UE transmission demand $L^{(c)}$. We make the natural assumption that the weight $\lambda_{i,j}^{(c)}$ depends on UE's information only. In other words, UEs only report the average channel condition for all available channels in every TTI. It is possible to show that problem (12.1) has multiple optimal solutions.

We propose to obtain it as

$$W_{\mathrm{P}}^{(c)} = \left[\alpha_1 \dot{r}^{(c)} + \alpha_2 \dot{l}^{(c)} + \alpha_3 \dot{q}^{(c)}\right]^{-1} ,$$

where $\alpha_i \in [0,1]$ and $\sum \alpha_i = 1$. In this case, we set α_1, α_2 and α_3 to 0.3, 0.5 and 0.2 respectively.

13 Security

SOLUTION 13.1

The Caesar code We write the possible solutions down, advancing one place in the alphabet for each row:

NBCMCMYUMS
OCDNDNZVNT
PDEOEOAWOU
QEFPFPBXPV
RFGQGQCYQW
SGHRHRDZRX
THISISEASY Clearly we can stop once recognizable English text appears. Note that the dependencies of natural languages are such that a very small number of letters are needed to determine the unique message.

SOLUTION 13.2

Unicity distance The unicity distance is $\log_{10} 26!/0.7$ or about 30 characters. Clearly substitution ciphers are not secure in any way. In fact one can usually do much better since a quite high degree of certainty is obtained already using the statistics for 5 letters of text. Furthermore, it is impossible to get enough text for statistics of dependencies beyond 9 already this is more text than in the Library of Congress. One applies instead rules of syntax and grammars to assist in the attack.

SOLUTION 13.3

Euclid's Algorithm

$$10480 = 2(3920) + 2640$$

$$3920 = 1(2640) + 1280$$

$$2640 = 2(1280) + 80$$

$$1280 = 16(80) + 0$$

Thus $\gcd(10480, 3920) = 80$.

SOLUTION 13.4

Factoring products of primes The number of primes in the two cases are 4×1017 and 4×1036 respectively. This would be a lot of divisions. Quadratic sieve filtering requires in the two cases $O(1521)$ and $O(15, 487)$ iterations respectively, each one being much more than a division. Nevertheless it is clear that an efficient factoring algorithm is the way to go here.

SOLUTION 13.5

Hash functions and sensing

(a) An obvious attack is to attempt to guess key k: try all possible sequences of the key length. Another is to try all possible signatures of length n in an attempt to forge a signature. Thus both key length and signature length have to be large enough to make this combinatorially difficult.

(b) A hash function performs a type of lossy compression on a sequence; one cannot reconstruct the original sequence from the signature. Similarly, all sensor measurements only partially represent a physical phenomenon, with greater processing resulting generally in less knowledge about the original phenomenon. However, hash functions are deterministic while sensing has randomness associated with it in the form of noise. Moreover, lossy compression codes are not designed to provide a unique and seemingly random signature but rather to produce a result that answers a query to the best fidelity possible, given constraints on the length of the report.

SOLUTION 13.6

Fighting infection

(a) For low infection probabilities, the requirement boils down to $100p < .01$, or $p < 10^{-4}$. While there will be some sweeps in which there will be a higher levels of unreliable data, on average the requirement is met.

(b) Consider the probability that a given is not corrupted by the end is $(1-p)^{100} = 1-q$, where q is the final infection probability. The chances that two of the three voters are corrupted is $3q^2(1-q)+q^3 = 0.01$. If q is reasonably small, then $3q^2$ is approximately 0.01 and so $q = 5.8 \times 10^{-2}$. Thus $(1-p) = (1-5.8 \times 10^{-2})^{1/100}$, or $p = 6 \times 10^{-4}$. Thus 2 of 3 voting enables either a six times higher infection rate or a slower audit cycle.

SOLUTION 13.7

Information theft

(a) Clearly 100 units are at risk at tier 1 while at tier 2 there are 300 units of raw data available prior to application of the compression algorithm. At tier 1 there are only 30 units available.

(b) In this case there are 20 units at risk in tier 1, 60 at tier 2, and 10 at tier 1. Note that while in the end the same amount of information is presented to the end user, less information irrelevant to the query is at risk within the network.

(c) Here tier 1 has 100 units at risk, tier 2 has 300 units at risk, and tier 1 has 600 units at risk since all the raw data has been forwarded. This is obviously the most dangerous strategy.

SOLUTION 13.8

Physical security

Clearly many answers are plausible; the assumptions made on cost and value of information obtained will matter. However, generally redundancy is a good idea at all levels. Camouflage and embedding are appropriate for multiple levels also, although unnecessary when the devices are in a secured room. Tamper detection devices can be quite simple (was the package opened?) and can be applied at multiple levels also. Hardened casings might be needed in any case for environmental robustness at gateway levels and up, while secured rooms/facilities would be reserved for the highest network layers in general such as a certificate authority.

SOLUTION 13.9

People and data reliability

Personal trust among people is based on past actions, and in particular is gained when actions are taken that cost something (e.g., time) without apparent personal benefit. Technical trust is gained based on the track record of actual accomplishments, and the publicity that surrounds them. Trust is also reinforced by the comments of peers, with the comments of trusted friends or technical authorities assigned more weight than those of others. Thus both direct observations and the opinions of a social network establish reputation.

BIBLIOGRAPHY

[1] S. Boyd and L. Vandenberghe., “Convex Optimization", Cambridge Univ Press, 2004.

[2] G. J. Pottie and W. J. Kaiser., “Principles of Embedded Networked Systems Design”, Cambridge Univ Press, 2009.

[3] W. Dargie and C. Poellabauer., “Fundamentals of Wireless Sensor Networks: Theory and Practice”, Wiley, 2010.

[4] A. Papoulis., “ Probability, Random Variables and Stochastic Processes”, Mcgraw-Hill College, 1991.

[5] U. Tiberi, C. Fischione, K.H. Johansson and M.D. Di Benedetto., “Energy-efficient sampling of NCS over IEEE 802.15.4 wireless networks”. Submitted for journal publication, 2011.

[6] F. Gustafsson., “Statistical Sensor Fusion”, Studentlitteratur, 2nd edition, 2012.

[7] Z. Chair and P.K. Varshney., “Optimal data fusion in multiple sensor detection systems,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 22, no. 1, pp. 98-101, 1986.

[8] R. Niu and P.K. Varshney., “Distributed detection and fusion in a large wireless sensor network of random size,” *EURASIP Journal on Wireless Communications and Networking*, vol. 4, pp. 462-472, 2005.