7 Sensors and Actuators

A **sensor** is a device that measures a physical quantity. An **actuator** is a device that alters a physical quantity. In electronic systems, sensors often produce a voltage that is proportional to the physical quantity being measured. The voltage may then be converted to a number by an **analog-to-digital converter** (**ADC**). A sensor that is packaged with an ADC is called a **digital sensor**, whereas a sensor without an ADC is called an **analog sensor**. A digital sensor will have a limited precision, determined by the number of bits used to represent the number (this can be as few as one!). Conversely, an actuator is commonly driven by a voltage that may be converted from a number by a **digital-to-analog converter** (**DAC**). An actuator that is packaged with a DAC is called a **digital actuator**.

Today, sensors and actuators are often packaged with microprocessors and network interfaces, enabling them to appear on the Internet as services. The trend is towards a technology that deeply connects our physical world with our information world through such smart sensors and actuators. This integrated world is variously called the **Internet of Things** (**IoT**), **Industry 4.0**, the **Industrial Internet**, **Machine-to-Machine** (**M2M**), the **Internet of Everything**, the **Smarter Planet**, **TSensors** (Trillion Sensors), or **The Fog** (like The Cloud, but closer to the ground).

Some technologies for interfacing to sensors and actuators have emerged that leverage established mechanisms originally developed for ordinary Internet usage. For example, a sensor or actuator may be accessible via a web server using the so-called **Representational State Transfer** (**REST**) architectural style (Fielding and Taylor, 2002). In this style, data may be retrieved from a sensor or commands may be issued to an actuator by constructing a URL (uniform resource locator), as if you were accessing an ordinary web page from a browser, and then transmitting the URL directly to the sensor or actuator device, or to a web server that serves as an intermediary.

In this chapter, we focus not on such high-level interfaces, but rather on foundational properties of sensors and actuators as bridges between the physical and the cyber worlds. Key low-level properties include the rate at which measurements are taken or actuations are performed, the proportionality constant that relates the physical quantity to the measurement or control signal, the offset or bias, and the dynamic range. For many sensors and actuators, it is useful to model the degree to which a sensor or actuator deviates from a proportional measurement (its **nonlinearity**), and the amount of random variation introduced by the measurement process (its noise).

A key concern for sensors and actuators is that the physical world functions in a multidimensional continuum of time and space. It is an **analog** world. The world of software, however, is **digital**, and strictly quantized. Measurements of physical phenomena must be quantized in both magnitude and time before software can operate on them. And commands to the physical world that originate from software will also be intrinsically quantized. Understanding the effects of this quantization is essential.

This chapter begins in Section 7.1 with an outline of how to construct models of sensors and actuators, specifically focusing on linearity (and nonlinearity), bias, dynamic range, quantization, noise, and sampling. That section concludes with a brief introduction to signal conditioning, a signal processing technique to improve the quality of sensor data and actuator controls. Section 7.2 then discusses a number of common sensing problems, including measuring tilt and acceleration (accelerometers), measuring position and velocity (anemometers, inertial navigation, GPS, and other ranging and triangulation techniques), measuring rotation (gyroscopes), measuring sound (microphones), and measuring distance (rangefinders). The chapter concludes with Section 7.3, which shows how to apply the modeling techniques to actuators, focusing specifically on LEDs and motor controllers.

Higher-level properties that are not addressed in this chapter, but are equally important, include security (specifically access control), privacy (particularly for data flowing over the open Internet), name-space management, and commissioning. The latter is particularly big issue when the number of sensors or actuators get large. Commissioning is the process of associating a sensor or actuator device with a physical location (e.g., a temperature sensor gives the temperature of what?), enabling and configuring network interfaces, and possibly calibrating the device for its particular environment.

7.1 Models of Sensors and Actuators

Sensors and actuators connect the cyber world with the physical world. Numbers in the cyber world bear a relationship with quantities in the physical world. In this section, we provide models of that relationship. Having a good model of a sensor or actuator is essential to effectively using it.

7.1.1 Linear and Affine Models

Many sensors may be approximately modeled by an affine function. Suppose that a physical quantity $x(t)$ at time t is reported by the sensor to have value $f(x(t))$, where $f\colon \mathbb{R} \to \mathbb{R}$ is a function. The function f is **linear** if there exists a **proportionality con-**

stant $a \in \mathbb{R}$ such that for all $x(t) \in \mathbb{R}$

$$f(x(t)) = ax(t).$$

It is an **affine function** if there exists a proportionality constant $a \in \mathbb{R}$ and a **bias** $b \in \mathbb{R}$ such that

$$f(x(t)) = ax(t) + b. \tag{7.1}$$

Clearly, every linear function is an affine function (with $b = 0$), but not vice versa.

Interpreting the readings of such a sensor requires knowledge of the proportionality constant and bias. The proportionality constant represents the **sensitivity** of the sensor, since it specifies the degree to which the measurement changes when the physical quantity changes.

Actuators may also be modeled by affine functions. The affect that a command to the actuator has on the physical environment may be reasonably approximated by a relation like (7.1).

7.1.2 Range

No sensor or actuator truly realizes an affine function. In particular, the **range** of a sensor, the set of values of a physical quantity that it can measure, is always limited. Similarly for an actuator. Outside that range, an affine function model is no longer valid. For example, a thermometer designed for weather monitoring may have a range of $-20°$ to $50°$ Celsius. Physical quantities outside this range will typically **saturate**, meaning that they yield a maximum or a minimum reading outside their range. An affine function model of a sensor may be augmented to take this into account as follows,

$$f(x(t)) = \begin{cases} ax(t) + b & \text{if } L \le x(t) \le H \\ aH + b & \text{if } x(t) > H \\ aL + b & \text{if } x(t) < L, \end{cases} \tag{7.2}$$

where $L, H \in \mathbb{R}$, $L < H$, are the low and high end of the sensor range, respectively.

A relation between a physical quantity $x(t)$ and a measurement given by (7.2) is not an affine relation (it is, however, piecewise affine). In fact, this is a simple form of nonlinearity that is shared by all sensors. The sensor is reasonably modeled by an affine function within an **operating range** (L, H), but outside that operating range, its behavior is distinctly different.

7.1.3 Dynamic Range

Digital sensors are unable to distinguish between two closely-spaced values of the physical quantity. The **precision** p of a sensor is the smallest absolute difference between two values of a physical quantity whose sensor readings are distinguishable. The **dynamic range** $D \in \mathbb{R}_+$ of a digital sensor is the ratio

$$D = \frac{H - L}{p},$$

where H and L are the limits of the range in (7.2). Dynamic range is usually measured in decibels (see sidebar on page 189), as follows:

$$D_{dB} = 20 \log_{10} \left(\frac{H - L}{p} \right). \tag{7.3}$$

7.1.4 Quantization

A digital sensor represents a physical quantity using an n-bit number, where n is a small integer. There are only 2^n distinct such numbers, so such a sensor can produce only 2^n distinct measurements. The actual physical quantity may be represented by a real number $x(t) \in \mathbb{R}$, but for each such $x(t)$, the sensor must pick one of the 2^n numbers to represent it. This process is called **quantization**. For an ideal digital sensor, two physical quantities that differ by the precision p will be represented by digital quantities that differ by one bit, so precision and quantization become intertwined.

We can further augment the function f in (7.2) to include quantization, as illustrated in the following example.

Example 7.1: Consider a 3-bit digital sensor that can measure a voltage between zero and one volt. Such a sensor may be modeled by the function $f\colon \mathbb{R} \to \{0, 1, \cdots, 7\}$ shown in Figure 7.1. The horizontal axis is the input to the sensor (in volts), and the vertical axis is the output, with the value shown in binary to emphasize that this is a 3-bit digital sensor.

In the figure, the low end of the measurable range is $L = 0$, and the high end is $H = 1$. The precision is $p = 1/8$, because within the operating range, any

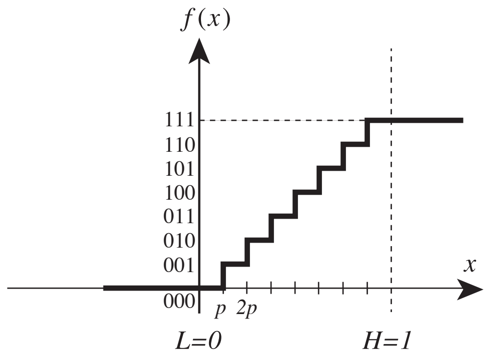

Figure 7.1: Sensor distortion function for a 3-bit digital sensor capable of measuring a range of zero to one volt, where the precision $p = 1/8$.

two inputs that differ by more than $1/8$ of a volt will yield different outputs. The dynamic range, therefore, is

$$D_{dB} = 20\log_{10}\left(\frac{H-L}{p}\right) \approx 18dB.$$

A function f like the one in Figure 7.1 that defines the output of a sensor as a function of its input is called the **sensor distortion function**. In general, an ideal n-bit digital sensor with a sensor distortion function like that shown in Figure 7.1 will have a precision given by

$$p = (H-L)/2^n$$

and a dynamic range of

$$D_{dB} = 20\log_{10}\left(\frac{H-L}{p}\right) 20\log_{10}(2^n) = 20n\log_{10}(2) \approx 6n\ dB. \tag{7.4}$$

Each additional bit yields approximately 6 decibels of dynamic range.

Example 7.2: An extreme form of quantization is performed by an **analog comparator**, which compares a signal value against a threshold and produces a binary value, zero or one. Here, the sensor function $f\colon \mathbb{R} \to \{0, 1\}$ is given by

$$f(x(t)) = \begin{cases} 0 & \text{if } x(t) \le 0 \\ 1 & \text{otherwise} \end{cases}$$

Such extreme quantization is often useful, because the resulting signal is a very simple digital signal that can be connected directly to a GPIO input pin of a microprocessor, as discussed in Chapter 10.

The analog comparator of the previous example is a one-bit ADC. The quantization error is high for such a converter, but using signal conditioning, as described below in Section 7.1.8, if the sample rate is high enough, the noise can be reduced considerably by digital low-pass filtering. Such a process is called **oversampling**; it is commonly used today because processing signals digitally is often less costly than analog processing.

Actuators are also subject to quantization error. A digital actuator takes a digital command and converts it to an analog physical action. A key part of this is the digital to analog converter (DAC). Because the command is digital, it has only a finite number of possible values. The precision with which an analog action can be taken, therefore, will depend on the number of bits of the digital signal and the range of the actuator.

As with ADCs, however, it is possible to trade off precision and speed. A **bang-bang controller**, for example, uses a one-bit digital actuation signal to drive an actuator, but updates that one-bit command very quickly. An actuator with a relatively slow response time, such as a motor, does not have much time to react to each bit, so the reaction to each bit is small. The overall reaction will be an average of the bits over time, much smoother than what you would expect from a one-bit control. This is the mirror image of oversampling.

The design of ADC and DAC hardware is itself quite an art. The effects of choices of sampling interval and number of bits are quite nuanced. Considerable expertise in signal processing is required to fully understand the implications of choices (see Lee and Varaiya (2011)). Below, we give a cursory view of this rather sophisticated topic. Section 7.1.8 discusses how to mitigate the noise in the environment and noise due to quantization, showing the intuitive result that it is beneficial to filter out frequency ranges that are not of interest. These frequency ranges are related to the sample rate. Hence, noise and sampling are the next topics.

7.1.5 Noise

By definition, **noise** is the part of a signal that we do not want. If we want to measure $x(t)$ at time t, but we actually measure $x'(t))$, then the noise is the difference,

$$n(t) = x'(t) - x(t).$$

Equivalently, the actual measurement is

$$x'(t) = x(t) + n(t), \tag{7.5}$$

a sum of what we want plus the noise.

Example 7.3: Consider using an accelerometer to measure the orientation of a slowly moving object (see Section 7.2.1 below for an explanation of why an accelerometer can measure orientation). The accelerometer is attached to the moving object and reacts to changes in orientation, which change the direction of the gravitational field with respect to the axis of the accelerometer. But it will also report acceleration due to vibration. Let $x(t)$ be the signal due to orientation and $n(t)$ be the signal due to vibration. The accelerometer measures the sum.

In the above example, noise is a side effect of the fact that the sensor is not measuring exactly what we want. We want orientation, but it is measuring acceleration. We can also model sensor imperfections and quantization as noise. In general, a sensor distortion function can be modeled as additive noise,

$$f(x(t)) = x(t) + n(t), \tag{7.6}$$

where $n(t)$ by definition is just $f(x(t)) - x(t)$.

It is useful to be able to characterize how much noise there is in a measurement. The **root mean square** (**RMS**) $N \in \mathbb{R}_+$ of the noise is equal to the square root of the average value of $n(t)^2$. Specifically,

$$N = \lim_{T \to \infty} \sqrt{\frac{1}{2T} \int_{-T}^{T} (n(\tau))^2 d\tau}. \tag{7.7}$$

This is a measure of (the square root of) **noise power**. An alternative (statistical) definition of noise power is the square root of the expected value of the square of $n(t)$. Formula (7.7) defines the noise power as an *average* over time rather than an expected value.

The **signal to noise ratio** (**SNR**, in decibels) is defined in terms of RMS noise,

$$SNR_{dB} = 20 \log_{10} \left(\frac{X}{N} \right),$$

where X is the RMS value of the input signal x (again, defined either as a time average as in (7.7) or using the expected value). In the next example, we illustrate how to calculate SNR using expected values, leveraging elementary probability theory.

Example 7.4: We can find the SNR that results from quantization by using (7.6) as a model of the quantizer. Consider Example 7.1 and Figure 7.1, which show a 3-bit digital sensor with an operating range of zero to one volt. Assume that the input voltage is equally likely to be anywhere in the range of zero to one volt. That is, $x(t)$ is a random variable with uniform distribution ranging from 0 to 1. Then the RMS value of the input x is given by the square root of the expected value of the square of $x(t)$, or

$$X = \sqrt{\int_0^1 x^2 dx} = \frac{1}{\sqrt{3}}.$$

Examining Figure 7.1, we see that if $x(t)$ is a random variable with uniform distribution ranging from 0 to 1, then the error $n(t)$ in the measurement (7.6) is equally likely to be anywhere in the range from $-1/8$ to 0. The RMS noise is therefore given by

$$N = \sqrt{\int_{-1/8}^{0} 8n^2 dn} = \sqrt{\frac{1}{3 \cdot 64}} = \frac{1}{8\sqrt{3}}.$$

The SNR is therefore

$$SNR_{dB} = 20\log_{10}\left(\frac{X}{N}\right) = 20\log_{10}(8) \approx 18dB.$$

Notice that this matches the 6 dB per bit dynamic range predicted by (7.4)!

To calculate the SNR in the previous example, we needed a statistical model of the input x (uniformly distributed from 0 to 1) and the quantization function. In practice, it is difficult to calibrate ADC hardware so that the input x makes full use of its range. That is, the input is likely to be distributed over less than the full range 0 to 1. It is also unlikely to be uniformly distributed. Hence, the actual SNR achieved in a system will likely be considerably less than the 6 dB per bit predicted by (7.4).

7.1.6 Sampling

A physical quantity $x(t)$ is a function of time t. A digital sensor will **sample** the physical quantity at particular points in time to create a discrete signal. In **uniform sampling**, there is a fixed time interval T between samples; T is called the **sampling interval**. The resulting signal may be modeled as a function $s\colon \mathbb{Z} \to \mathbb{R}$ defined as follows,

$$\forall\, n \in \mathbb{Z}, \quad s(n) = f(x(nT)), \tag{7.8}$$

where $\mathbb{Z}$ is the set of integers. That is, the physical quantity $x(t)$ is observed only at times $t = nT$, and the measurement is subjected to the sensor distortion function. The **sampling rate** is $1/T$, which has units of **samples per second**, often given as **Hertz** (written **Hz**, meaning cycles per second).

In practice, the smaller the sampling interval T, the more costly it becomes to provide more bits in an ADC. At the same cost, faster ADCs typically produce fewer bits and hence have either higher quantization error or smaller range.

Example 7.5: The ATSC digital video coding standard includes a format where the frame rate is 30 frames per second and each frame contains $1080 \times 1920 =$

Decibels

The term "**decibel**" is literally one tenth of a **bel**, which is named after Alexander Graham Bell. This unit of measure was originally developed by telephone engineers at Bell Telephone Labs to designate the ratio of the **power** of two signals.

Power is a measure of energy dissipation (work done) per unit time. It is measured in **watts** for electronic systems. One bel is defined to be a factor of 10 in power. Thus, a 1000 watt hair dryer dissipates 1 bel, or 10 dB, more power than a 100 watt light bulb. Let $p_1 = 1000$ watts be the power of the hair dryer and $p_2 = 100$ be the power of the light bulb. Then the ratio is

$$\log_{10}(p_1/p_2) = 1 \text{ bel}, \quad \text{or}$$

$$10\log_{10}(p_1/p_2) = 10 \text{ dB}.$$

Comparing against (7.3) we notice a discrepancy. There, the multiplying factor is 20, not 10. That is because the ratio in (7.3) is a ratio of amplitude (magnitude), not powers. In electronic circuits, if an amplitude represents the voltage across a resistor, then the power dissipated by the resistor is proportional to the *square* of the amplitude. Let a_1 and a_2 be two such amplitudes. Then the ratio of their powers is

$$10\log_{10}(a_1^2/a_2^2) = 20\log_{10}(a_1/a_2).$$

Hence the multiplying factor of 20 instead of 10 in (7.3). A 3 dB power ratio amounts to a factor of 2 in power. In amplitudes, this is a ratio of $\sqrt{2}$.

In audio, decibels are used to measure sound pressure. A statement like "a jet engine at 10 meters produces 120 dB of sound," by convention, compares sound pressure to a defined reference of 20 micropascals, where a pascal is a pressure of 1 newton per square meter. For most people, this is approximately the threshold of hearing at 1 kHz. Thus, a sound at 0 dB is barely audible. A sound at 10 dB has 10 times the power. A sound at 100 dB has 10^{10} times the power. The jet engine, therefore, would probably make you deaf without ear protection.

2,073,600 pixels. An ADC that is converting one color channel to a digital representation must therefore perform 2,073,600 × 30 = 62,208,000 conversions per second, which yields a sampling interval T of approximately 16 nsec. With such a short sampling interval, increasing the number of bits in the ADC becomes expensive. For video, a choice of $b = 8$ bits is generally adequate to yield good visual fidelity and can be realized at reasonable cost.

An important concern when sampling signals is that there are many distinct functions x that when sampled will yield the same signal s. This phenomenon is known as **aliasing**.

Example 7.6: Consider a sinusoidal sound signal at 1 kHz (kilohertz, or thousands of cycles per second),

$$x(t) = \cos(2000\pi t).$$

Suppose there is no sensor distortion, so the function f in (7.8) is the identity function. If we sample at 8000 samples per second (a rate commonly used in telephony), we get a sampling interval of $T = 1/8000$, which yields the samples

$$s(n) = f(x(nT)) = \cos(\pi n/4).$$

Suppose instead that we are given a sound signal at 9 kHz,

$$x'(t) = \cos(18,000\pi t).$$

Sampling at the same 8kHz rate yields

$$s'(n) = \cos(9\pi n/4) = \cos(\pi n/4 + 2\pi n) = \cos(\pi n/4) = s(n).$$

The 1 kHz and 9 kHz sound signals yield exactly the same samples, as illustrated in Figure 7.2. Hence, at this sampling rate, these two signals are aliases of one another. They cannot be distinguished.

Aliasing is a complex and subtle phenomenon (see Lee and Varaiya (2011) for details), but a useful rule of thumb for uniform sampling is provided by the **Nyquist-Shannon sampling theorem**. A full study of the subject requires the machinery of Fourier transforms, and is beyond the scope of this text. Informally, this theorem states that a set of samples at sample rate $R = 1/T$ uniquely defines a continuous-time signal that is a sum of sinusoidal components with frequencies less than $R/2$. That is, among all continuous-time signals that are sums of sinusoids with frequencies less than $R/2$, there is only one that matches any given set of samples taken at rate R. The rule of thumb, therefore, is that if you sample a signal where the most rapid expected variation occurs at frequency $R/2$, then sampling the signal at a rate at least R will result in samples that uniquely represent the signal.

Example 7.7: In traditional telephony, engineers have determined that intelligible human speech signals do not require frequencies higher than 4 kHz. Hence, removing the frequencies above 4 kHz and sampling an audio signal with human speech at 8kHz is sufficient to enable reconstruction of an intelligible audio signal from the samples. The removal of the high frequencies is accomplished by a frequency selective filter called an **anti-aliasing filter**, because it prevents frequency components above 4 kHz from masquerading as frequency components below 4 kHz.

The human ear, however, can easily discern frequencies up to about 15 kHz, or 20 kHz in young people. Digital audio signals intended for music, therefore, are sampled at frequencies above 40 kHz; 44.1 kHz is a common choice, a rate defined originally for use in compact discs (CDs).

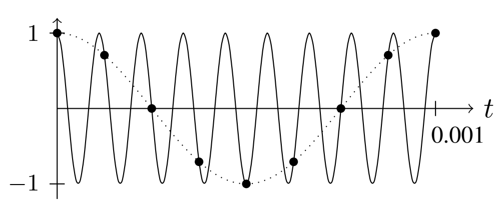

Figure 7.2: Illustration of aliasing, where samples of a 9 kHz sinusoid taken at 8,000 samples per second are the same as samples of a 1 kHz sinusoid taken at 8,000 samples per second.

Example 7.8: Air temperature in a room varies quite slowly compared to sound pressure. We might assume, for example, that the most rapid expected air temperature variations occur at rates measured in minutes, not seconds. If we want to capture variations on the scale of a minute or so, then we should take at least two samples per minute of the temperature measurement.

7.1.7 Harmonic Distortion

A form of nonlinearity that occurs even within the operating range of sensors and actuators is **harmonic distortion**. It typically occurs when the sensitivity of the sensor or actuator is not constant and depends on the magnitude of the signal. For example, a microphone may be less responsive to high sound pressure than to lower sound pressure.

Harmonic distortion is a nonlinear effect that can be modeled by powers of the physical quantity. Specifically, **second harmonic distortion** is a dependence on the square of the physical quantity. That is, given a physical quantity $x(t)$, the measurement is modeled as

$$f(x(t)) = ax(t) + b + d_2(x(t))^2, \tag{7.9}$$

where d_2 is the amount of second harmonic distortion. If d_2 is small, then the model is nearly affine. If d_2 is large, then it is far from affine. The $d_2(x(t))^2$ term is called second harmonic distortion because of the effect it has the frequency content of a signal $x(t)$ that is varying in time.

Example 7.9: Suppose that a microphone is stimulated by a purely sinusoidal input sound

$$x(t) = \cos(\omega_0 t),$$

where t is time in seconds and ω_0 is the frequency of the sinusoid in radians per second. If the frequency is within the human auditory range, then this will sound like a pure tone.

A sensor modeled by (7.9) will produce at time t the measurement

$$
\begin{aligned}
x'(t) &= ax(t) + b + d_2(x(t))^2 \\
&= a\cos(\omega_0 t) + b + d_2\cos^2(\omega_0 t) \\
&= a\cos(\omega_0 t) + b + \frac{d_2}{2} + \frac{d_2}{2}\cos(2\omega_0 t),
\end{aligned}
$$

where we have used the trigonometric identity

$$
\cos^2(\theta) = \frac{1}{2}(1 + \cos(2\theta)).
$$

To humans, the bias term $b + d_2/2$ is not audible. Hence, this signal consists of a pure tone, scaled by a, and a distortion term at twice the frequency, scaled by $d_2/2$. This distortion term is audible as harmonic distortion as long as $2\omega_0$ is in the human auditory range.

A cubic term will introduce **third harmonic distortion**, and higher powers will introduce higher harmonics.

The importance of harmonic distortion depends on the application. The human auditory system is very sensitive to harmonic distortion, but the human visual system much less so, for example.

7.1.8 Signal Conditioning[1]

Noise and harmonic distortion often have significant differences from the desired signal. We can exploit those differences to reduce or even eliminate the noise or distortion. The easiest way to do this is with **frequency selective filtering**. Such filtering relies on Fourier theory, which states that a signal is an additive composition of sinusoidal signals of different frequencies. While Fourier theory is beyond the scope of this text (see Lee and Varaiya (2003) for details), it may be useful to some readers who have some background to see how to apply that theory in the context of embedded systems. We do that in this section.

[1]This section may be skipped on a first reading. It requires a background in signals and systems at the level typically covered in a sophomore or junior engineering course.

Example 7.10: The accelerometer discussed in Example 7.3 is being used to measure the orientation of a slowly moving object. But instead it measures the sum of orientation and vibration. We may be able to reduce the effect of the vibration by **signal conditioning**. If we assume that the vibration $n(t)$ has higher frequency content than the orientation $x(t)$, then frequency-selective filtering will reduce the effects of vibration. Specifically, vibration may be mostly rapidly changing acceleration, whereas orientation changes more slowly, and filtering can remove the rapidly varying components, leaving behind only the slowly varying components.

To understand the degree to which frequency-selective filtering helps, we need to have a model of both the desired signal x and the noise n. Reasonable models are usually statistical, and analysis of the signals requires using the techniques of random processes, estimation, and machine learning. Although such analysis is beyond the scope of this text, we can gain insight that is useful in many practical circumstances through a purely deterministic analysis.

Our approach will be to condition the signal $x' = x + n$ by filtering it with an LTI system S called a **conditioning filter**. Let the output of the conditioning filter be given by

$$y = S(x') = S(x + n) = S(x) + S(n),$$

where we have used the linearity assumption on S. Let the residual error signal after filtering be defined to be

$$r = y - x = S(x) + S(n) - x. \tag{7.10}$$

This signal tells us how far off the filtered output is from the desired signal. Let R denote the RMS value of r, and X the RMS value of x. Then the SNR after filtering is

$$\mathit{SNR}_{dB} = 20 \log_{10} \left(\frac{X}{R} \right),$$

We would like to design the conditioning filter S to maximize this SNR. Since X does not depend on S, we maximize this SNR if we minimize R. That is, we choose S to minimize the RMS value of r in (7.10).

Although determination of this filter requires statistical methods beyond the scope of this text, we can draw some intuitively appealing conclusions by examining (7.10). It is easy to show that the denominator is bounded as follows,

$$R = \mathit{RMS}(r) \leq \mathit{RMS}(S(x) - x) + \mathit{RMS}(n) \tag{7.11}$$

where RMS is the function defined by (7.7). This suggests that we may be able to minimize R by making $S(x)$ close to x (i.e., make $S(x) \approx x$) while making $\mathit{RMS}(n)$ small. That is, the filter S should do minimal damage to the desired signal x while filtering out as much as possible of the noise.

As illustrated in Example 7.3, x and n often differ in frequency content. In that example, x contains only low frequencies, and n contains only higher frequencies. Therefore, the best choice for S will be a lowpass filter.

7.2 Common Sensors

In this section, we describe a few sensors and show how to obtain and use reasonable models of these sensors.

7.2.1 Measuring Tilt and Acceleration

An **accelerometer** is a sensor that measures **proper acceleration**, which is the acceleration of an object as observed by an observer in free fall. As we explain here, gravitational force is indistinguishable from acceleration, and therefore an accelerometer measures not just acceleration, but also gravitational force. This result is a precursor to Albert Einstein's Theory of General Relativity and is known as Einstein's **equivalence principle** (Einstein, 1907).

A schematic view of an accelerometer is shown in Figure 7.3. A movable mass is attached via a spring to a fixed frame. Assume that the sensor circuitry can measure the position of the movable mass relative to the fixed frame (this can be done, for example, by measuring capacitance). When the frame accelerates in the direction of the double arrow in the figure, the acceleration results in displacement of the movable mass, and hence this acceleration can be measured.

The movable mass has a neutral position, which is its position when the spring is not deformed at all. It will occupy this neutral position if the entire assembly is in free fall,

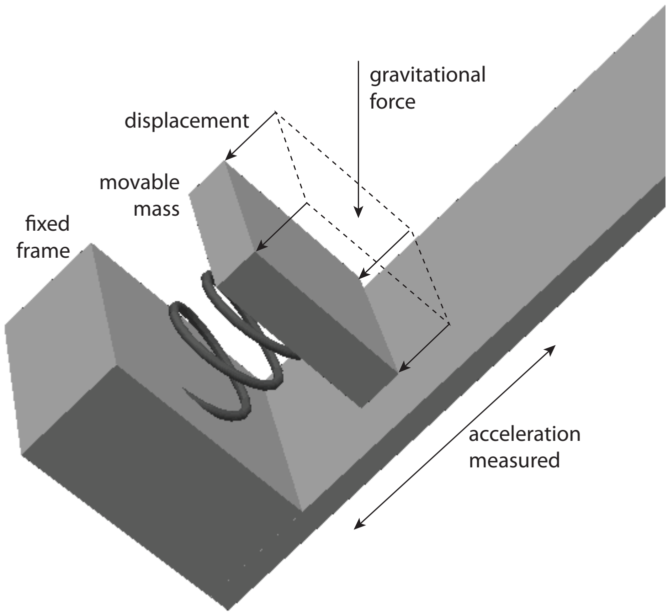

Figure 7.3: A schematic of an accelerometer as a spring-mass system.

or if the assembly is lying horizontally. If the assembly is instead aligned vertically, then gravitational force will compress the spring and displace the mass. To an observer in free fall, this looks exactly as if the assembly were accelerating upwards at the **acceleration of gravity**, which is approximately $g = 9.8$ meters/second^2.

An accelerometer, therefore, can measure the tilt (relative to gravity) of the fixed frame. Any acceleration experienced by the fixed frame will add or subtract from this measurement. It can be challenging to separate these two effects, gravity and acceleration. The combination of the two is what we call proper acceleration.

Assume x is the proper acceleration of the fixed frame of an accelerometer at a particular time. A digital accelerometer will produce a measurement $f(x)$ where

$$f\colon (L, H) \to \{0, \ldots, 2^b - 1\}$$

where $L \in \mathbb{R}$ is the minimum measurable proper acceleration and $H \in \mathbb{R}$ is the maximum, and $b \in \mathbb{N}$ is the number of bits of the ADC.

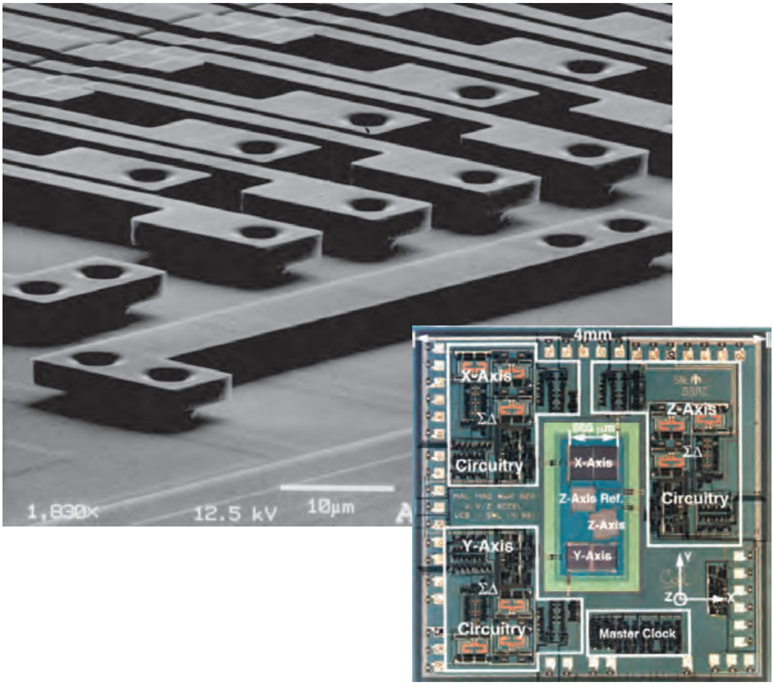

Figure 7.4: A silicon accelerometer consists of flexible silicon fingers that deform under gravitational pull or acceleration (Lemkin and Boser, 1999).

Today, accelerometers are typically implemented in silicon (see Figure 7.4), where silicon fingers deform under gravitational pull or acceleration (see for example Lemkin and Boser (1999)). Circuitry measures the deformation and provides a digital reading. Often, three accelerometers are packaged together, giving a three-axis accelerometer. This can be used to measure orientation of an object relative to gravity, plus acceleration in any direction in three-dimensional space.

7.2.2 Measuring Position and Velocity

In theory, given a measurement x of acceleration over time, it is possible to determine the velocity and location of an object. Consider an object moving in a one-dimensional space.

Let the position of the object over time be $p\colon \mathbb{R}_+ \to \mathbb{R}$, with initial position $p(0)$. Let the velocity of the object be $v\colon \mathbb{R}_+ \to \mathbb{R}$, with initial velocity $v(0)$. And let the acceleration be $x\colon \mathbb{R}_+ \to \mathbb{R}$. Then

$$p(t) = p(0) + \int_0^t v(\tau) d\tau,$$

and

$$v(t) = v(0) + \int_0^t x(\tau) d\tau.$$

Note, however, that if there is a non-zero bias in the measurement of acceleration, then $p(t)$ will have an error that grows proportionally to t^2. Such an error is called **drift**, and it makes using an accelerometer alone to determine position not very useful. However, if the position can be periodically reset to a known-good value, using for example GPS, then an accelerometer becomes useful to approximate the position between such settings.

In some circumstances, we can measure velocity of an object moving through a medium. For example, an **anemometer** (which measures air flow) can estimate the velocity of an aircraft relative to the surrounding air. But using this measurement to estimate position is again subject to drift, particularly since the movement of the surrounding air ensures bias.

Direct measurements of position are difficult. The **global positioning system** (**GPS**) is a sophisticated satellite-based navigation system using triangulation. A GPS receiver listens for signals from four or more GPS satellites that carry extremely precise clocks. The satellites transmit a signal that includes the time of transmission and the location of the satellite at the time of transmission. If the receiver were to have an equally precise clock, then upon receiving such a signal from a satellite, it would be able to calculate its distance from the satellite using the speed of light. Given three such distances, it would be able to calculate its own position. However, such precise clocks are extremely expensive. Hence, the receiver uses a fourth such distance measurement to get a system of four equations with four unknowns, the three dimensions of its location and the error in its own local clock.

The signal from GPS satellites is relatively weak and is easily blocked by buildings and other obstacles. Other mechanisms must be used, therefore, for indoor localization. One such mechanism is **WiFi fingerprinting**, where a device uses the known location of WiFi access points, the signal strength from those access points, and other local information. Another technology that is useful for indoor localization is **bluetooth**, a short-distance wireless communication standard. Bluetooth signals can be used as beacons, and signal strength can give a rough indication of distance to the beacon.

Strength of a radio signal is a notoriously poor measure of distance because it is subject to local diffraction and reflection effects on the radio signal. In an indoor environment, a radio signal is often subject to **multipath**, where it propagates along more than one path to a target, and at the target experiences either constructive or destructive interference. Such interference introduces wide variability in signal strength that can lead to misleading measures of distance. As of this writing, mechanisms for accurate indoor localization are not widely available, in notable contrast to outdoor localization, where GPS is available worldwide.

7.2.3 Measuring Rotation

A **gyroscope** is a device that measures changes in orientation (rotation). Unlike an accelerometer, it is (mostly) unaffected by a gravitational field. Traditional gyroscopes are bulky rotating mechanical devices on a double gimbal mount. Modern gyroscopes are either MEMS devices (microelectromechanical systems) using small resonating structures, or optical devices that measure the difference in distance traveled by a laser beam around a closed path in opposite directions, or (for extremely high precision) devices that leverage quantum effects.

Gyroscopes and accelerometers may be combined to improve the accuracy of **inertial navigation**, where position is estimated by **dead reckoning**. Also called **ded reckoning** (for deduced reckoning), dead reckoning starts from a known initial position and orientation, and then uses measurements of motion to estimate subsequent position and orientation. An **inertial measurement unit** (**IMU**) or **inertial navigation system** (**INS**) uses a gyroscope to measure changes in orientation and an accelerometer to measure changes in velocity. Such units are subject to drift, of course, so they are often combined with GPS units, which can periodically provided "known good" location information (though not orientation). IMUs can get quite sophisticated and expensive.

7.2.4 Measuring Sound

A **microphone** measures changes in sound pressure. A number of techniques are used, including electromagnetic induction (where the sound pressure causes a wire to move in a magnetic field), capacitance (where the distance between a plate deformed by the sound pressure and a fixed plate varies, causing a measurable change in capacitance), or the piezoelectric effect (where charge accumulates in a crystal due to mechanical stress).

Microphones for human audio are designed to give low distortion and low noise within the human hearing frequency range, about 20 to 20,000 Hz. But microphones are also used outside this range. For example, an **ultrasonic rangefinder** emits a sound outside the human hearing range and listens for an echo. It can be used to measure the distance to a sound-reflecting surface.

7.2.5 Other Sensors

There are many more types of sensors. For example, measuring temperature is central to HVAC systems, automotive engine controllers, overcurrent protectection, and many industrial chemical processes. Chemical sensors can pick out particular pollutants, measure alcohol concentration, etc. Cameras and photodiodes measure light levels and color. Clocks measure the passage of time.

A switch is a particularly simple sensor. Properly designed, it can sense pressure, tilt, or motion, for example, and it can often be directly connected to the GPIO pins of a microcontroller. One issue with switches, however, is that they may **bounce**. A mechanical switch that is based on closing an electrical contact has metal colliding with metal, and the establishment of the contact may not occur cleanly in one step. As a consequence, system designers need to be careful when reacting to the establishment of an electrical contact or they may inadvertently react several times to a single throwing of the switch.

7.3 Actuators

As with sensors, the variety of available actuators is enormous. Since we cannot provide comprehensive coverage here, we discuss two common examples, LEDs and motor control. Further details may be found in Chapter 10, which discusses particular microcontroller I/O designs.

7.3.1 Light-Emitting Diodes

Very few actuators can be driven directly from the digital I/O pins (GPIO pins) of a microcontroller. These pins can source or sink a limited amount of current, and any attempt to exceed this amount risks damaging the circuits. One exception is **light-emitting diodes** (**LEDs**), which when put in series with a resistor, can often be connected directly to a

GPIO pin. This provides a convenient way for an embedded system to provide a visual indication of some activity.

Example 7.11: Consider a microcontroller that operates at 3 volts from a coin-cell battery and specifies that its GPIO pins can sink up to 18 mA. Suppose that you wish to turn on and off an LED under software control (see Chapter 10 for how to do this). Suppose you use an LED that, when forward biased (turned on), has a voltage drop of 2 volts. Then what is the smallest resistor you can put in series with the LED to safely keep the current within the 18 mA limit? **Ohm's law** states

$$V_R = IR, \tag{7.12}$$

where V_R is the voltage across the resistor, I is the current, and R is the resistance. The resistor will have a voltage drop of $V_R = 3 - 2 = 1$ volt across it (two of the 3 supply volts drop across the LED), so the current flowing through it will be

$$I = 1/R.$$

To limit this current to 18 mA, we require a resistance

$$R \geq 1/0.018 \approx 56 \text{ ohms}.$$

If you choose a 100 ohm resistor, then the current flowing through the resistor and the LED is

$$I = V_R/100 = 10\text{mA}.$$

If the battery capacity is 200 mAh (milliamp-hours), then driving the LED for 20 hours will completely deplete the battery, not counting any power dissipation in the microcontroller or other circuits. The power dissipated in the resistor will be

$$P_R = V_R I = 10mW.$$

The power dissipated in the LED will be

$$P_L = 2I = 20mW.$$

These numbers give an indication of the heat generated by the LED circuit (which will be modest).

The calculations in the previous example are typical of what you need to do to connect any device to a micro controller.

7.3.2 Motor Control

A **motor** applies a torque (angular force) to a load proportional to the current through the motor windings. It might be tempting, therefore, to apply a voltage to the motor proportional to the desired torque. However, this is rarely a good idea. First, if the voltage is digitally controlled through a DAC, then we have to be very careful to not exceed the current limits of the DAC. Most DACs cannot deliver much power, and require a power amplifier between the DAC and the device being powered. The input to a power amplifier has high impedance, meaning that at a given voltage, it draws very little current, so it can usually be connected directly to a DAC. The output, however, may involve substantial current.

Example 7.12: An audio amplifier designed to drive 8 ohm speakers can often deliver 100 watts (peak) to the speaker. Power is equal to the product of voltage and current. Combining that with Ohm's law, we get that power is proportional to the square of current,

$$P = RI^2,$$

where R is the resistance. Hence, at 100 watts, the current through an 8 ohm speaker is

$$I = \sqrt{P/R} = \sqrt{100/8} \approx 3.5\text{amps},$$

which is a substantial current. The circuitry in the power amplifier that can deliver such a current without overheating and without introducing distortion is quite sophisticated.

Power amplifiers with good linearity (where the output voltage and current are proportional to the input voltage) can be expensive, bulky, and inefficient (the amplifier itself dissipates significant energy). Fortunately, when driving a motor, we do not usually need such a power amplifier. It is sufficient to use a switch that we can turn on and off with a digital signal from a microcontroller. Making a switch that tolerates high currents is much easier than making a power amplifier.

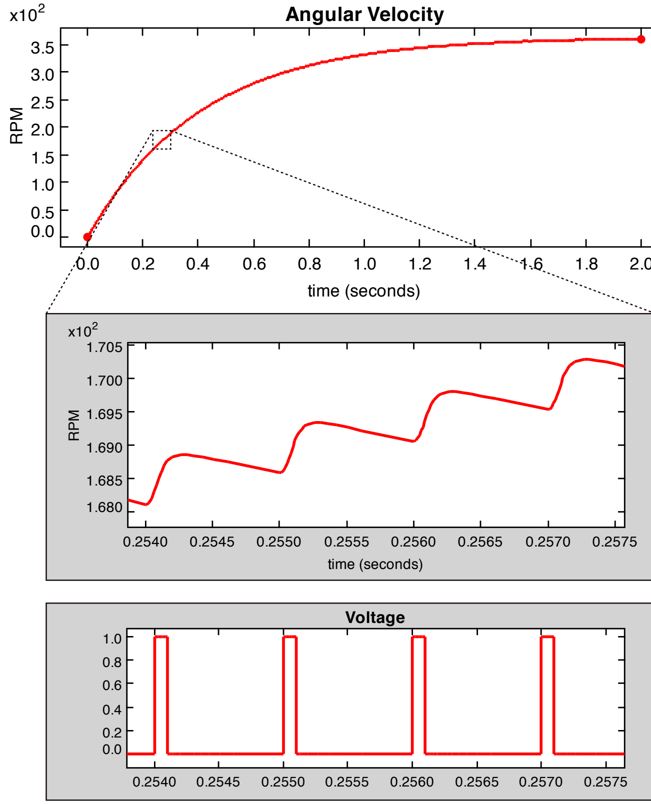

Figure 7.5: PWM control of a DC motor.

We use a technique called **pulse width modulation** (**PWM**), which can efficiently deliver large amounts of power under digital control, as long as the device to which the power is being delivered can tolerate rapidly switching on and off its power source. Devices that tolerate this include LEDs, incandescent lamps (this is how dimmers work), and DC motors. A PWM signal, as shown on the bottom of Figure 7.5, switches between a high level and a low level at a specified frequency. It holds the signal high for a fraction of the cycle period. This fraction is called the **duty cycle**, and in Figure 7.5, is 0.1, or 10%.

A **DC motor** consists of an electromagnet made by winding wires around a core placed in a magnetic field made with permanent magnets or electromagnets. When current flows through the wires, the core spins. Such a motor has both inertia and inductance that smooth its response when the current is abruptly turned on and off, so such motors tolerate PWM signals well.

Let $\omega\colon \mathbb{R} \to \mathbb{R}$ represent the angular velocity of the motor as a function of time. Assume we apply a voltage v to the motor, also a function of time. Then using basic circuit theory, we expect the voltage and current through the motor to satisfy the following equation,

$$v(t) = Ri(t) + L\frac{di(t)}{dt},$$

where R is the resistance and L the inductance of the coils in the motor. That is, the coils of the motor are modeled as series connection of a resistor and an inductor. The voltage drop across the resistor is proportional to current, and the voltage drop across the inductor is proportional to the rate of change of current.

However, motors exhibit a phenomenon that when a coil rotates in a magnetic field, it *generates* a current (and corresponding voltage). In fact, a motor can also function as an electrical generator; if instead of mechanically coupling it to a passive load, you couple it to a source of power that applies a torque to the motor, then the motor will generate electricity. Even when the motor is being used a motor rather than a generator, there will be some torque resisting the rotation, called the **back electromagnetic force**, due to this tendency to generate electricity when rotated. To account for this, the above equation becomes

$$v(t) = Ri(t) + L\frac{di(t)}{dt} + k_b\omega(t), \tag{7.13}$$

where k_b is an empirically determined **back electromagnetic force constant**, typically expressed in units of volts/RPM (volts per revolutions per minute).

Having described the electrical behavior of the motor in (7.13), we can use the techniques of Section 2.1 to describe the mechanical behavior. We can use the rotational version of Newton's second law, $F = ma$, which replaces the force F with torque, the mass m with moment of inertia and the acceleration a with angular acceleration. The torque T on the motor is proportional to the current flowing through the motor, adjusted by friction and any torque that might be applied by the mechanical load,

$$T(t) = k_T i(t) - \eta\omega(t) - \tau(t),$$

where k_T is an empirically determined **motor torque constant**, η is the kinetic friction of the motor, and τ is the torque applied by the load. By Newton's second law, this needs to be equal to the moment of inertia I times the angular acceleration, so

$$I \frac{d\omega(t)}{dt} = k_T i(t) - \eta\omega(t) - \tau(t). \tag{7.14}$$

Together, (7.14) and (7.13) describe how the motor responds to an applied voltage and mechanical torque.

Example 7.13: Consider a particular motor with the following parameters,

$$\begin{aligned} I &= 3.88 \times 10^{-7} \text{ kg}\cdot\text{meters}^2 \\ k_b &= 2.75 \times 10^{-4} \text{ volts/RPM} \\ k_T &= 5.9 \times 10^{-3} \text{ newton}\cdot\text{meters/amp} \\ R &= 1.71 \text{ ohms} \\ L &= 1.1 \times 10^{-4} \text{ henrys} \end{aligned}$$

Assume that there is no additional load on the motor, and we apply a PWM signal with frequency 1 kHz and duty cycle 0.1. Then the response of the motor is as shown in Figure 7.5, which has been calculated by numerically simulating according to equations (7.14) and (7.13). Notice that the motor settles at a bit more than 350 RPM after 2 seconds. As shown in the detailed plot, the angular velocity of the motor jitters at a rate of 1 kHz. It accelerates rapidly when the PWM signal is high, and decelerates when it is low, the latter due to friction and back electromagnetic force. If we increase the frequency of the PWM signal, then we can reduce the magnitude of this jitter.

In a typical use of a PWM controller to drive a motor, we will use the feedback control techniques of Section 2.4 to set the speed of the motor to a desired RPM. To do this, we require a measurement of the speed of the motor. We can use a sensor called a **rotary encoder**, or just **encoder**, which reports either the angular position or velocity (or both) of a rotary shaft. There are many different designs for such encoders. A very simple one provides an electrical pulse each time the shaft rotates by a certain angle, so that counting pulses per unit time will provide a measurement of the angular velocity.

7.4 Summary

The variety of sensors and actuators that are available to engineers is enormous. In this chapter, we emphasize *models* of these sensors and actuators. Such models are an essential part of the toolkit of embedded systems designers. Without such models, engineers would be stuck with guesswork and experimentation.

Exercises

1. Show that the composition $f \circ g$ of two affine functions f and g is affine.

2. The dynamic range of human hearing is approximately 100 decibels. Assume that the smallest difference in sound levels that humans can effectively discern is a sound pressure of about 20 μPa (micropascals).

   (a) Assuming a dynamic range of 100 decibels, what is the sound pressure of the loudest sound that humans can effectively discriminate?

   (b) Assume a perfect microphone with a range that matches the human hearing range. What is the minimum number of bits that an ADC should have to match the dynamic range of human hearing?

3. The following questions are about how to determine the function

$$f\colon (L, H) \to \{0, \ldots, 2^B - 1\},$$

   for an accelerometer, which given a proper acceleration x yields a digital number $f(x)$. We will assume that x has units of "g's," where 1g is the acceleration of gravity, approximately $g = 9.8\text{meters}/\text{second}^2$.

   (a) Let the bias $b \in \{0, \ldots, 2^B - 1\}$ be the output of the ADC when the accelerometer measures no proper acceleration. How can you measure b?

   (b) Let $a \in \{0, \ldots, 2^B - 1\}$ be the *difference* in output of the ADC when the accelerometer measures 0g and 1g of acceleration. This is the ADC conversion of the sensitivity of the accelerometer. How can you measure a?

   (c) Suppose you have measurements of a and b from parts (3b) and (3a). Give an affine function model for the accelerometer, assuming the proper acceleration is x in units of g's. Discuss how accurate this model is.

   (d) Given a measurement $f(x)$ (under the affine model), find x, the proper acceleration in g's.

   (e) The process of determining a and b by measurement is called **calibration** of the sensor. Discuss why it might be useful to individually calibrate each particular accelerometer, rather than assume fixed calibration parameters a and b for a collection of accelerometers.

(f) Suppose you have an ideal 8-bit digital accelerometer that produces the value $f(x) = 128$ when the proper acceleration is 0g, value $f(x) = 1$ when the proper acceleration is 3g to the right, and value $f(x) = 255$ when the proper acceleration is 3g to the left. Find the sensitivity a and bias b. What is the dynamic range (in decibels) of this accelerometer? Assume the accelerometer never yields $f(x) = 0$.

4. (this problem is due to Eric Kim)

You are a Rebel Alliance fighter pilot evading pursuit from the Galactic Empire by hovering your space ship beneath the clouds of the planet Cory. Let the positive z direction point upwards and be your ship's position relative to the ground and v be your vertical velocity. The gravitational force is strong with this planet and induces an acceleration (in a vacuum) with absolute value g. The force from air resistance is linear with respect to velocity and is equal to rv, where the drag coefficient $r \leq 0$ is a constant parameter of the model. The ship has mass M. Your engines provide a vertical force.

(a) Let $L(t)$ be the input be the vertical lift force provided from your engines. Write down the dynamics for your ship for the position $z(t)$ and velocity $v(t)$. Ignore the scenario when your ship crashes. The right hand sides should contain $v(t)$ and $L(t)$.

(b) Given your answer to the previous problem, write down the explicit solution to $z(t)$ and $v(t)$ when the air resistance force is negligible and $r = 0$. At initial time $t = 0$, you are $30m$ above the ground and have an initial velocity of $-10\frac{m}{s}$. *Hint: Write $v(t)$ first then write $z(t)$ in terms of $v(t)$.*

(c) Draw an actor model using integrators, adders, etc. for the system that generates your vertical position and velocity. Make sure to label all variables in your actor model.

(d) Your engine is slightly damaged and you can only control it by giving a pure input, switch, that when present instantaneously switches the state of the engine from on to off and vice versa. When on, the engine creates a positive lift force L and when off $L = 0$. Your instrumentation panel contains an accelerometer. Assume your spaceship is level (i.e. zero pitch angle) and the accelerometer's positive z axis points upwards. Let the input sequence of engine switch commands be

$$\text{switch}(t) = \left\{ \begin{array}{ll} \mathit{present} & \text{if } t \in \{.5, 1.5, 2.5, \ldots\} \\ \mathit{absent} & \text{otherwise} \end{array} \right\}.$$

To resolve ambiguity at switching times $t = .5, 1.5, 2.5, \ldots$, at the moment of transition the engine's force takes on the new value instantaneously. Assume that air resistance is negligible (i.e. $r = 0$), ignore a crashed state, and the engine is on at $t = 0$.

Sketch the vertical component of the accelerometer reading as a function of time $t \in \mathbb{R}$. Label important values on the axes. *Hint: Sketching the graph for force first would be helpful.*

(e) If the spaceship is flying at a constant height, what is the value read by the accelerometer?