MICROELECTROMECHANICAL SYSTEMS AND SENSORS

Course reader

Chapters 1–4 · draft for review

Bachelor programme · Electrical Engineering · 7th semester

Institute of Energy and Electrical Engineering

Draft, August 2026

How to use this reader. It does not replace a textbook. It carries the material that no textbook covers — the datasheet workflow, register-level interfacing, the specific parts on your bench — and points you at Morris & Langari, *Measurement and Instrumentation* (3rd ed., Elsevier 2020) for the underlying theory. Each chapter ends with exercises and full answers; work them before the corresponding laboratory.

Numerical values in worked examples have been checked by computation. Component figures labelled “Part A”, “Part B” and “Part C” are representative of real commercial devices but are not attributed to a named product; always verify against the datasheet revision in your hand.

1 · The measurement chain

What you should be able to do after this chapter

1. **Draw** the path from a physical quantity to a stored number, name each stage, and say what information can be lost at each one.
2. **Classify** an unfamiliar device as a sensor, a transducer or an actuator, and name the physical quantity it responds to.
3. **Predict**, from scaling arguments, why microscopic mechanical devices are fast and sensitive — and name at least one property that becomes *worse* as a device shrinks.
4. **Convert** a vague request into a measurement specification: quantity, range, resolution, accuracy, bandwidth, environment and output.

Notice that all four are things you *do*. None of them is "know about".

1.1 A motor that died under observation

A plant operated a 1500 rpm induction motor driving a pump. Bearing failure on that machine was expensive, so the maintenance department instrumented it: an accelerometer bolted to the bearing housing, a microcontroller, a daily log, and a trend line on a screen that an engineer actually looked at every week.

For six weeks the screen showed a vibration component near 20 Hz, slowly rising in amplitude. Everybody who saw it read it the same way: a gradual increase in load. Unremarkable.

In the seventh week the bearing seized and the motor was destroyed.

The post-mortem found no fault. The accelerometer met every number in its datasheet. The circuit board was sound. The firmware contained no bug. The logger had recorded every sample it was asked to record. The engineer had looked at the screen.

Every component met its specification, nobody was negligent, and the motor was scrap.

Hold that. We will return to it at the end of the chapter, once you have the equipment to explain it. The explanation is not exotic — it is arithmetic — and it is the reason this course exists.

1.2 Three words, used precisely

From here on these three words mean different things, and datasheets will not respect the distinction for you.

A **sensor** converts a physical quantity into a signal you can read. A MEMS microphone converts sound pressure into a voltage.

A **transducer** converts energy from one form into another. This is the general term; a sensor is the special case that points *inward*, from the world into your electronics. A loudspeaker, a piezoelectric disc and a strain gauge are all transducers.

An **actuator** converts a signal into a physical action on the world. A MEMS mirror converts a voltage into an angular deflection of a reflective surface.

Two remarks worth carrying forward.

First, **every actuator is also a measurement problem.** You only know that an actuator acted if you sense the result. A motor commanded to turn is not a motor that turned.

Second, some sensors are **passive**: they produce nothing at all on their own. A strain gauge changes its electrical resistance when it is stretched, but a resistance is not a signal. Until you pass a known current through it, or place it in a bridge with a known excitation voltage, it says nothing. Chapter 12 is largely about supplying that excitation correctly.

> **The quantity you are measuring has a name: the *measurand*.** Get into the habit of naming it explicitly and in SI units, because a surprising number of measurement failures begin with a team that never agreed on what it was measuring. "Vibration" is not a measurand. "Housing acceleration in m/s², RMS, in the band 1–4 kHz" is.

1.3 A measurement is a chain, not a component

Students arriving at this course usually picture a measurement as a component: you buy the sensor, you read the number. That picture is the single most expensive misconception in the field, and replacing it is the work of this chapter.

A measurement is a **path** from a physical quantity to a stored, trustworthy number. Every stage on that path transforms the information, and every stage can damage it. Figure 1.1 is the path. It is also, more or less, the syllabus of this course: every one of the sixteen lectures lives inside one of these seven boxes.

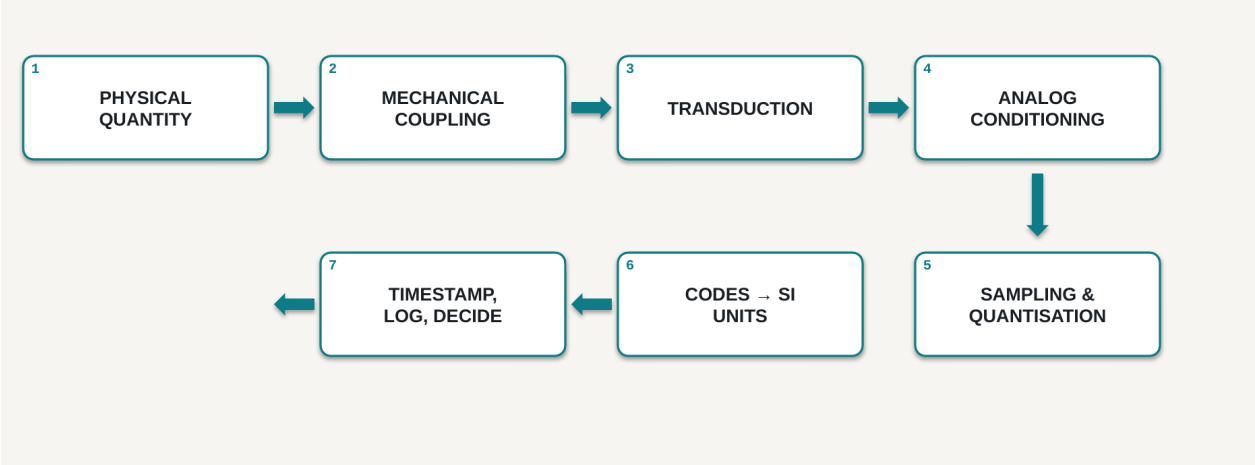

Figure 1.1 — The seven stages of a measurement chain. Read the top row left to right, then drop to the second row and read right to left.

Take the stages one at a time.

Stage 1 · The physical quantity. The only thing in the diagram that is unambiguously true. The motor really is vibrating. Everything to the right of this box is a story your instrument tells about it, and your job is to make the story faithful.

Stage 2 · Mechanical coupling. Before any electronics exist, the quantity has to reach the sensing element — through a bolt, a bracket, a housing, an adhesive pad, a length of tubing, a thermal interface. That path has its own transfer function: it attenuates, it delays, it resonates, and it does all three differently at different frequencies. It is part of your instrument, and it appears in no datasheet ever written. This is the stage that students omit and that experienced engineers check first.

Stage 3 · Transduction. Here physics becomes electricity: a capacitance changes, a resistance changes, a charge appears, a voltage develops. Chapter 4 covers the mechanisms; Chapters 5 to 11 cover which mechanism suits which measurand.

Stage 4 · Analog conditioning. Amplify, filter, level-shift, buffer. One filter in this stage — the **anti-alias filter** — is going to matter enormously by the end of this chapter. Remember that it lives here, in front of the converter.

Stage 5 · Sampling and quantisation. Two distinct operations that students merge for years, and they fail in different ways. **Sampling** chops *time*: the converter looks at the signal only at particular instants. **Quantisation** chops *amplitude*: each look is rounded to the nearest of a finite set of levels. Sampling too slowly and quantising too coarsely produce entirely different symptoms.

Stage 6 · Codes to SI units. A sensor never outputs an acceleration. It outputs an integer. Somebody multiplies that integer by a constant taken from a datasheet — and if the constant is wrong, everything downstream is confidently, precisely wrong. Chapter 2 is about where that constant comes from and what it depends on.

Stage 7 · Timestamp, log, decide. A sample without a time is nearly worthless. If you cannot say *when* a sample was taken, you cannot compute a frequency — and in the story that opened this chapter, frequency was the whole answer.

Figure 1.2 shows the same chain carrying one real vibration from housing to decision.

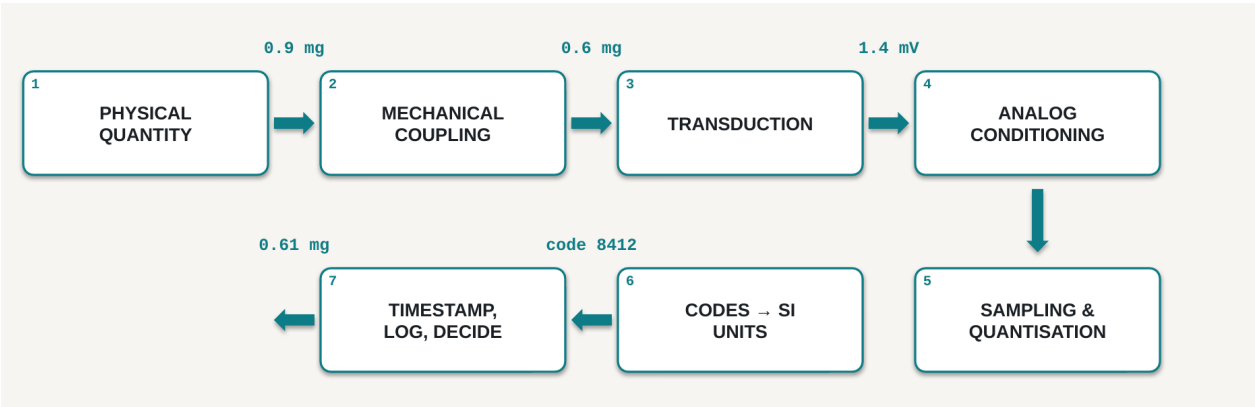

Figure 1.2 — One 0.9 mg vibration, all the way to a decision. Note stage 2: the mechanical path has already lost a third of the signal before a single electron has moved. Of all the numbers on this diagram, only those inside stages 3 to 6 would appear in any datasheet.

1.4 Where the truth dies, and whether you can recover it

Not all damage is equal. Some losses are annoying and can be repaired later by calculation or calibration; others are permanent and must be prevented by design. Knowing which is which is a large part of engineering judgement.

Stage	What is lost	Can you recover it?
2 Mechanical coupling	amplitude and phase, differently at each frequency	**No** — you must characterise it yourself
3 Transduction	cross-axis leakage, nonlinearity, temperature drift	Partly, by calibration
4 Conditioning	saturation clips peaks; a wrong filter removes the signal	**No**
5 Sampling	everything above half the sample rate, folded into your band	**Never**
5 Quantisation	detail below one least-significant bit	Partly, by averaging
6 Scaling to units	accuracy, if the sensitivity constant is wrong	Yes — recompute
7 Timestamping	the time axis, and therefore all frequency content	**No**

Table 1.1 — Losses along the chain. Three of these are permanent: they are design decisions, not calibration problems.

There is exactly one *never* in that table. Note which row it is on.

1.5 Why the microscopic world behaves differently

You have spent several years building mechanical intuition on machines you can hold. At the micrometre scale a good part of that intuition inverts, and it is worth knowing precisely which part.

What is actually inside

Figure 1.3 is a capacitive MEMS accelerometer in cross-section. A **proof mass** of mass m hangs on flexible suspension beams of stiffness k, between two fixed plates separated from it by gaps d. When the device accelerates, the mass lags behind its frame; one gap grows and the other shrinks; the difference in the two capacitances is the signal.

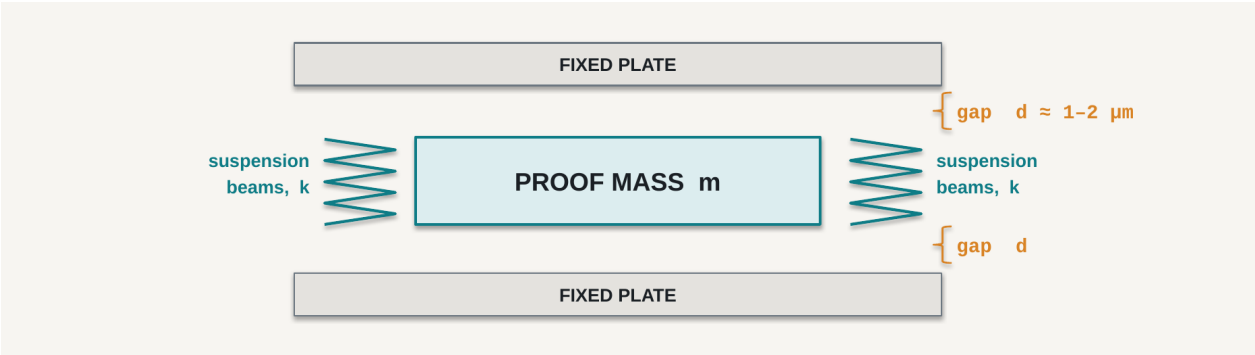

Figure 1.3 — A capacitive MEMS accelerometer. Acceleration displaces the proof mass; each gap changes; the differential capacitance $C = \varepsilon A/d$ is the output. Typical gaps are 1–2 µm — smaller than a red blood cell, which is why a single dust particle inside the package is a catastrophe and why Chapter 4 spends its time on packaging rather than on lithography.

If you understood the spring–mass system in mechanics, you already understand this device. Everything in a MEMS accelerometer datasheet is a consequence of m, k and d.

Scaling: the two exponents

Now shrink the whole structure. Scale every dimension — length, width, thickness — by the same factor s. This is called **isotropic scaling**, and it is the cleanest way to see what changes.

Mass follows volume:

$$m \propto L^3$$

Stiffness does not. For a beam of width w, thickness t and length L, the bending stiffness is

$$k = E \cdot w \cdot t^3 / 4L^3$$

Substitute w, t and L all scaled by s:

$$k \propto (s \cdot s^3) / s^3 = s$$

so

$$k \propto L^1$$

These two do not scale together, and that single fact is the whole of MEMS. Students assume the two effects cancel. They do not, and the gap between the exponents is where the entire field lives.

The consequence appears in the resonant frequency:

$$f_0 = (1 / 2\pi) \cdot \sqrt{(k / m)} \propto \sqrt{(L^1 / L^3)} = \sqrt{(L^{-2})} = 1 / L$$

Shrink a device by ten and its resonant frequency rises by ten.

Device	Size	Resonance
Bridge span	100 m	≈ 0.2 Hz
Tuning fork	10 cm	≈ 440 Hz
MEMS accelerometer	300 µm	≈ 5 kHz
MEMS gyroscope drive	100 µm	≈ 20 kHz
MEMS RF resonator	10 µm	≈ 100 MHz and above

Table 1.2 — Five orders of magnitude in size, eight in frequency.

This matters because a sensor cannot faithfully report signals near or above its own resonance. A high f_0 is not a curiosity; it is what buys you **bandwidth**, and bandwidth is what let anybody put an accelerometer on a bearing in the first place.

The other exponent, and the price you pay

Surface area also scales differently from volume:

$$A / V \propto L^2 / L^3 = 1 / L$$

As things get smaller, they become almost entirely surface. Gravity and inertia are volume effects; friction, adhesion and surface tension are area effects. Shrink far enough and the area effects win — which is why a microscopic mechanical structure can touch itself and simply stay stuck. That failure is called **stiction**, and it is a real production concern.

So the honest ledger of shrinking a device looks like this:

Gets better	Gets worse
Bandwidth, since $f_0 \propto 1/L$	Thermomechanical noise floor
Response time, thermal settling	Stiction; surface forces dominate
Power per device	Sensitivity to packaging stress
Cost per device at volume	Temperature drift and offset stability
Thousands per wafer, batch fabricated	Contamination: a dust speck is fatal

Table 1.3 — Shrinking is a trade, not an improvement.

The noise entry deserves a sentence, because it is the honest answer to *"why not just make them smaller?"*. The proof mass is what averages out the random thermal buffeting of the gas molecules around it. The noise-equivalent acceleration produced by that buffeting scales roughly as

$$a_n \propto \sqrt{(4 k_B T \omega_0 / m Q)}$$

so it **rises as the proof mass falls**. You buy bandwidth and you pay in noise floor. The datasheet will make you choose, and Chapter 2 is where you learn to read the price.

> **A macroscopic engineer worries about mass. A MEMS engineer worries about surfaces.**

1.6 Worked calculation: how a fast signal becomes a slow one

We can now explain the motor.

The **Nyquist–Shannon sampling theorem** states that to represent a signal containing frequencies up to f_{max}, you must sample at a rate

$$f_s > 2 f_{max}$$

Sample more slowly and the information is not merely missing — it is *misfiled*. A component at frequency f higher than $f_s/2$ reappears inside your measurement band at the **alias frequency**

$$f_{alias} = | f - n \cdot f_s | \quad (n = \text{the nearest integer multiple})$$

It arrives indistinguishable from a real signal at that lower frequency. Nothing downstream can separate them, because by the time the samples exist the distinction has already been destroyed.

The numbers from the post-mortem

Three facts were established after the motor failed:

- the bearing-fault signature sat at **1520 Hz**
- the system sampled at **100 Hz**
- there was **no anti-alias filter** in stage 4

To observe 1520 Hz honestly you would need to sample above 3040 Hz. The system sampled thirty times too slowly. So where did that energy go?

$$n = \text{round}(\ 1520\ /\ 100\ ) = 15$$

$$f_alias = |\ 1520 - 15 \times 100\ | = |\ 1520 - 1500\ | = 20\ \text{Hz}$$

There is the 20 Hz line. It was never load variation. It was the bearing destroying itself, faithfully recorded and filed under the wrong frequency.

Figure 1.4 shows the mechanism. The fast teal signal is the real vibration. The red dots are the only instants at which the converter looked. Draw the smoothest curve you can through just those dots and you get the dashed red line — a slow oscillation that is not present in the physical world at all.

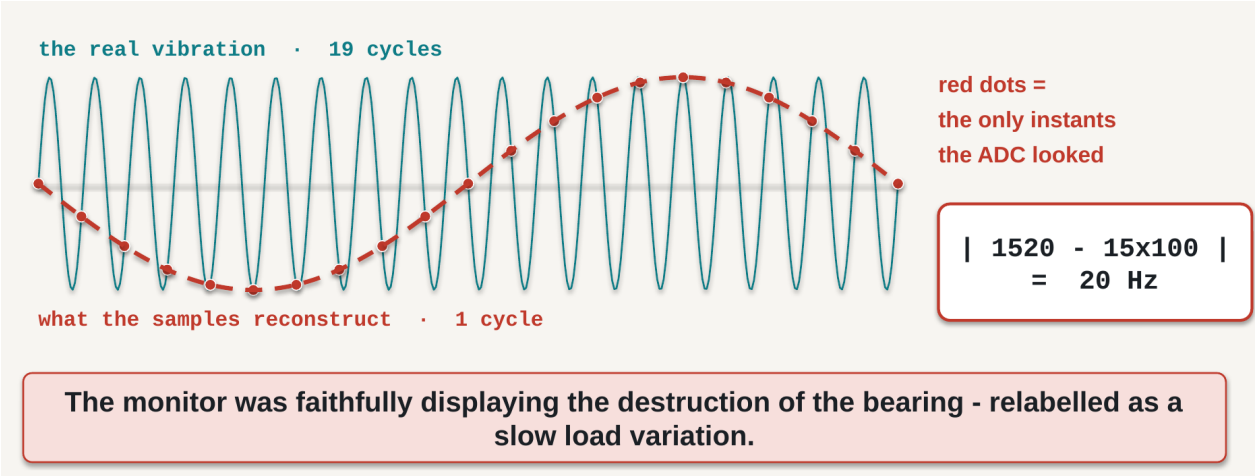

Figure 1.4 — Aliasing. Drawn at 19 signal cycles per 20 samples so the mechanism is visible; the motor's real ratio was 1520:100, which is identical arithmetic. The samples are perfectly accurate. Their *interpretation* is not.

Why nobody caught it

Here is the detail that turns this from a curiosity into a lesson. The shaft turned at 1500 rpm, which is **25 Hz**. A vibration component at 20 Hz was therefore entirely plausible to every engineer who looked at that screen — the same order as the running speed, drifting slowly upward, exactly what a gradually loading machine does.

The wrong answer was believable. That is why it survived six weeks of weekly review.

Two conclusions, and they are the reason this chapter exists:

1. **Aliasing is permanent.** No filter applied afterwards, no clever software, no amount of machine learning recovers a frequency you failed to sample. The only defence is an anti-alias filter in stage 4 and an adequate rate in stage 5 — both chosen *before* the system is built.
2. **Nothing was broken. Every part met its specification. The system was never specified.** No individual component owned the sampling decision, so nobody made it.

1.7 From a wish to a specification

Nobody will hand you a specification. They will hand you a sentence like:

> *"Just tell me if the motor is vibrating too much."*

There is not one number in it. Until there is, you cannot buy anything, you cannot design anything, and — importantly — you cannot be held to anything. Converting that sentence into an engineering document is the first task of every measurement project, and these seven questions do it.

	Question
Quantity	What physical variable, in what units?
Range	The smallest and largest value that must be reported?
Resolution	The smallest change that must be distinguishable?
Accuracy	How wrong may a reading be — **and over what temperature range**?
Bandwidth	How fast does it change? What is the highest frequency that matters?
Environment	Temperature, humidity, vibration, EMI, supply, power budget?
Output	Who consumes the number, in what form, how often, timestamped how?

Table 1.4 — Seven questions that turn a wish into a specification.

Two of these deserve emphasis. **Accuracy** without a temperature range is marketing, not engineering; Chapter 2 shows exactly how much damage that omission does. And **bandwidth** is the question that killed the motor.

Applied to the motor, the wish becomes this:

	Specification	Where the number came from
Quantity	housing acceleration, m/s² RMS	vibration is what a failing bearing radiates
Range	±16 g	impact transients, not steady vibration
Resolution	≤ 5 mg	earliest detectable fault ≈ 30 mg
Accuracy	±5 % of reading, 0–70 °C	trend detection, not absolute metrology
Bandwidth	**≥ 4 kHz, anti-aliased**	bearing signature spans 1–4 kHz
Environment	0–70 °C, 24 V rail, oily, EMI from a drive	the motor's actual cabinet
Output	RMS + spectrum, 1 Hz, timestamped ±1 ms	so that a frequency can be computed

Table 1.5 — The same request, specified.

Notice two things about Table 1.5. Every line has a **reason** — a specification value without a justification is a guess wearing a suit. And notice what is absent: no manufacturer, no part number, no price. Those come afterwards, and choosing them is the subject of Chapter 2.

The bandwidth line is the one that was missing from the real system. One line, in one document:

```
Bandwidth: ≥ 4 kHz, anti-aliased. Sample rate: ≥ 8 kHz.
```

One line. One motor.

1.8 Reading

This chapter's material corresponds to **Morris & Langari, *Measurement and Instrumentation*, 3rd edition (Elsevier, 2020), Chapter 1**, "Fundamentals of Measurement Systems". Read it for the formal treatment of the measurement-system block diagram and the vocabulary of measurement.

The sampling theorem and quantisation are developed properly in **Chapter 6**, "Data Acquisition and Signal Processing" — we return to that material in Chapter 3 of this reader, so a first pass now is enough.

For MEMS structures and fabrication, the free ITMO text *Микроэлектромеханические системы и датчики* (Университет ИТМО, 2020) covers device constructions and operating principles in its Chapters 1 and 2, with self-check questions. We use it properly in Chapter 4.

Chapter and section numbering should be checked against the edition available to you.

1.9 Exercises

1.1 List the stages of a measurement chain in order, from physical quantity to stored value.

1.2 Give one example of information lost at the analog-to-digital converter that cannot be recovered by any later processing. Explain why "cannot" is the right word.

1.3 A MEMS resonator is scaled down isotropically by a factor of 5. What happens to its resonant frequency, and why?

1.4 Name one sensor property that gets *worse* as a device shrinks, and explain the mechanism.

1.5 Classify each of the following, and name its measurand or its output action: (a) a MEMS microphone, (b) a piezoelectric buzzer, (c) a strain gauge, (d) a MEMS mirror. Which one produces no signal at all without external excitation?

1.6 A system samples at 500 Hz with no anti-alias filter. A real 1800 Hz component is present. At what frequency does it appear in the recorded data?

1.7 A drone must hold its altitude to ±0.5 m using a barometric pressure sensor. (a) What is the measurand — and is it altitude? (b) Roughly what pressure change corresponds to 0.5 m of altitude near sea level? (c) Explain why the *range* and the *resolution* required of this sensor differ by several orders of magnitude.

1.8 In one sentence: why did the motor's monitor display a calm 20 Hz line?

1.9 Write the seven specification lines for the following request: *"The greenhouse gets too hot in the afternoon — do something about it."* State an assumption wherever the request does not give you enough information.

Answers

1.1 Measurand → mechanical coupling and mounting → transduction → analog conditioning, including anti-alias filtering → sampling and quantisation → raw codes scaled to SI units → timestamp, log, decide.

1.2 Any frequency content above half the sample rate. Sampling folds it into the measurement band as an alias, arriving indistinguishable from a genuine low-frequency component; since no record of the original frequency survives in the samples, no later processing can separate them. Detail below one LSB is also lost, though averaging can partially recover it.

1.3 It rises by a factor of 5. Under isotropic scaling $k \propto L$ while $m \propto L^3$, so $f_0 \propto \sqrt{(k/m)} \propto 1/L$.

1.4 The thermomechanical noise floor: noise-equivalent acceleration rises as the proof mass falls, so a smaller device has a worse noise floor. Alternatively stiction, because $A/V \propto 1/L$ means surface forces come to dominate body forces; or increased sensitivity to package stress and thermal drift.

1.5 (a) Sensor; measurand sound pressure. (b) Actuator (and a transducer); converts an electrical signal into acoustic output. (c) Sensor; measurand strain — **this is the passive one**, since it only changes resistance and needs an excitation current or bridge voltage to produce a signal. (d) Actuator; converts a voltage into angular deflection of a mirror.

1.6 n = round(1800/500) = 4, so $f$alias = |1800 − 2000| = **200 Hz**.

1.7 (a) The measurand is **absolute air pressure**, in pascals or hectopascals; altitude is a *derived* quantity, computed from pressure using a model of the atmosphere. (b) Near sea level pressure falls by roughly 12 Pa per metre, so 0.5 m ≈ **6 Pa**, about 0.06 hPa. (c) The sensor must *operate over* the full range of weather and site pressure — perhaps 950 to 1050 hPa, a span of 10 000 Pa — while *resolving* changes of about 6 Pa. Range and resolution therefore differ by more than three orders of magnitude, and confusing the two is one of the most common specification errors.

1.8 The ≈1520 Hz bearing-fault energy was aliased by a 100 Hz sample rate with no anti-alias filter, appearing at |1520 − 15×100| = 20 Hz, which was misread as normal load variation because the shaft itself turned at 25 Hz.

1.9 Answers will vary; a defensible version is: **Quantity** air temperature in °C at canopy height (assumption: air, not soil or leaf temperature). **Range** −10 to +60 °C. **Resolution** 0.5 °C. **Accuracy** ±1 °C over 0–50 °C. **Bandwidth** very low — a greenhouse has a thermal time constant of minutes, so 0.01 Hz is ample, and one sample per minute is generous. **Environment** high humidity, condensation, direct sunlight requiring a radiation shield, mains supply available. **Output** one timestamped reading per minute to a controller, with a threshold that opens a vent. Credit belongs to any answer that states its assumptions and justifies its numbers — particularly one that notices the bandwidth is *low*, and that direct sunlight on the sensor is the dominant error source rather than the sensor itself.

2 · Specifications, and choosing a part

What you should be able to do after this chapter

1. **Distinguish** accuracy, precision, resolution and sensitivity, and sort a list of parameters into static and dynamic characteristics.
2. **Locate** a number in a real datasheet together with the conditions under which it was measured, and state at least one performance limit the datasheet cannot specify.
3. **Compute** noise-limited resolution from a noise-density figure and a bandwidth, and decide whether noise or quantisation dominates.
4. **Select** between candidate devices by building an error budget, and defend the choice by naming the dominant error term.

2.1 Two front pages

You have written a specification. Now you must buy something. Thousands of parts exist, every number is published, and the datasheets are free — so selection looks like a table lookup.

Here is the job:

> A solar-tracker frame must report its tilt angle to **±0.5°**, outdoors, over an ambient range of **0 to 40 °C**. It will be calibrated once on the bench, at 20 °C.

And here are two candidates, as their front pages present them.

PART A	
Accelerometer, 3-axis	
Resolution	16-bit
Full scale	±2 g
Sensitivity	0.061 mg/LSB
Noise density	100 µg/√Hz
Unit price	€1.80
"HIGH RESOLUTION · LOW NOISE"	

PART B	
Accelerometer, 3-axis	
Resolution	14-bit
Full scale	±4 g
Sensitivity	0.488 mg/LSB
Noise density	100 µg/√Hz
Unit price	€4.20

The job: report the tilt of a solar-tracker frame to ±0.5°, outdoors, 0 to 40 °C.

Figure 2.1 — Two candidate accelerometers as advertised. Part A offers eight times finer resolution at less than half the price, and says so. Part B says nothing for itself. *(These are representative commercial figures rather than a named product, chosen so that the arithmetic in this chapter is exact.)*

Which do you specify? Commit to an answer now, before reading on.

Most engineers choose Part A, and most of the time the reasoning is not lazy — sixteen bits genuinely is better than fourteen, and €1.80 genuinely is better than €4.20. By the end of this chapter you will be able to show that **Part A cannot meet the requirement**, that **Part B meets it with a fourfold margin**, and that the number which decides it appears on neither front page.

2.2 First, turn the requirement into a number

You cannot budget an error against an angle. So before reading any datasheet, convert the requirement into the physical quantity the sensor actually measures.

A tilted accelerometer at rest senses a component of gravity. For a tilt of angle θ from horizontal, the component along the sensitive axis is $g \sin \theta$. So a tilt of half a degree corresponds to

$$\sin(0.5°) \times 1\ g = 0.008727\ g = 8.73\ mg$$

Write that number down and keep it in front of you for the rest of the chapter.

> **8.73 mg is the entire signal we are trying to measure.**

Every error term from here on gets compared against those 8.73 mg. Any term larger than that is fatal. Any term far below it is irrelevant — no matter how impressive it looks on a front page. This single conversion turns *"which is better?"*, which is an opinion, into arithmetic.

2.3 Four words that get confused

Most engineering arguments about sensors are really arguments about these four words.

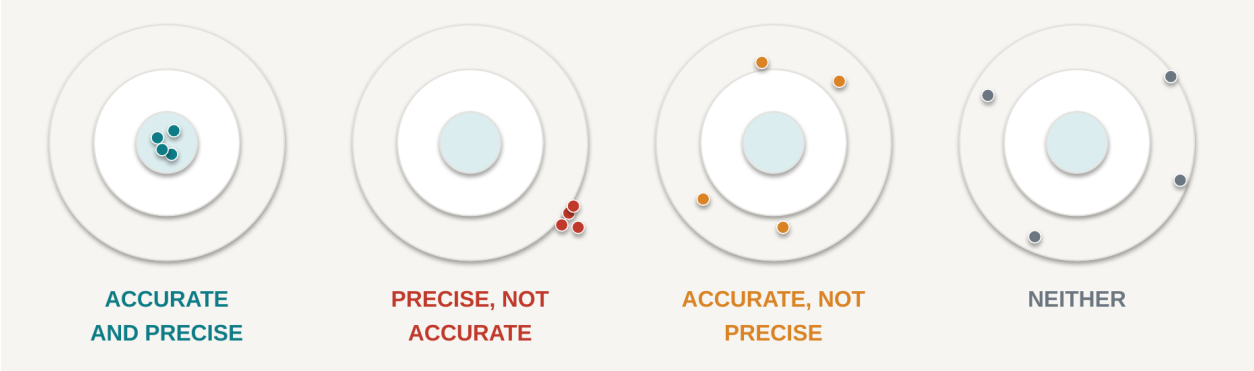

Figure 2.2 — The same target, four different instruments. The centre is the *true* value.

Precision is how tightly the shots group. **Accuracy** is where the group sits relative to the truth. They are independent properties: the second target — tightly grouped and completely wrong — is the dangerous instrument, because every reading agrees with every other reading and they are all wrong by the same amount.

Resolution does not appear on that picture at all. Resolution is how finely you can *report* where the shot landed. You can report a wrong position to six decimal places, and this is precisely the trap Part A is setting.

Sensitivity is different again: it is the change in output per unit change in input — for our accelerometer, mg per least-significant bit. Sensitivity tells you how the scale is drawn, not how truthful it is.

Consider a concrete case. A pressure sensor sits in a chamber held at a true, constant 1000.0 hPa. It is read one hundred times. Every reading falls between 1012.3 and 1012.5 hPa.

That device is **precise but not accurate**: superbly repeatable, and 12.4 hPa wrong. And it raises the question that carries the rest of this chapter:

> **Which of these two problems can a calibration fix?**

The 12.4 hPa offset — **yes**. Measure it once against a reference and subtract it.

The 0.2 hPa scatter — **no**. Random scatter cannot be removed by calibration, only reduced by averaging, and averaging costs you bandwidth. If you average sixteen samples to cut the noise by a factor of four, you have also made your instrument sixteen times slower.

That gives us a rule worth memorising, because it is the spine of Chapter 14 and of several laboratories:

> **Systematic error is a calibration problem. Random error is a bandwidth problem. Drift is a component-selection problem.**

The third line is the one that ruins projects. Drift passes every test you run on the bench at 22 °C and fails in the field in February, because it *moves after you calibrated it*. You cannot calibrate away a quantity that changes when you are not looking.

2.4 The vocabulary, sorted

Datasheets are organised around this vocabulary, so it is worth having sorted once. **Static** characteristics describe behaviour when nothing is changing; **dynamic** characteristics describe behaviour when things change.

Static	Meaning
Range	smallest to largest measurable value
Sensitivity	output change per unit of input
Resolution	smallest distinguishable change
Threshold	smallest input that produces any output at all
Offset	output when the input is zero
Linearity	deviation from a straight-line response
Hysteresis	does the reading depend on which direction you approached from?
Repeatability	same input, same output, measured again later

Dynamic	Meaning
Bandwidth	highest frequency reported faithfully
Response time	time to settle after a step input
Output data rate (ODR)	samples delivered per second — **not** the same as bandwidth
Noise density	noise per √Hz; becomes an RMS figure once you choose a bandwidth
Group delay	how late the filtered answer arrives
Cross-sensitivity	response to the axis or quantity you did *not* want
Drift	slow change with time and temperature

Table 2.1 — Static and dynamic characteristics. Two entries are worth flagging now.

ODR is not bandwidth. A sensor will happily hand you two hundred numbers per second while telling you nothing about signal content above 50 Hz. Confusing the two is the most common arithmetic error in this chapter, and §2.6 shows exactly what it costs.

Cross-sensitivity is real in every device: an accelerometer shaken along x reports a little z. We return to it in §2.8 for an important reason.

2.5 Read the conditions first, then the number

A datasheet is a legal document, not a promise. The headline is on page one; the truth is in the conditions.

Parameter	Min	Typ	Max	Unit	Conditions
Sensitivity	–	0.061	–	mg/LSB	FS = ±2 g, 25 °C
Zero-g level	–	±40	–	mg	25 °C, after soldering
Zero-g temp. coefficient	–	±0.5	–	mg/°C	−40 to +85 °C
Noise density	–	100	–	µg/√Hz	ODR = 200 Hz, BW = 50 Hz
Cross-axis sensitivity	–	±1	–	%	package level, not board level

▸ **Every number is "typ".** No min, no max. You are being told the average of a production run, not a

Figure 2.3 — A representative extract from an accelerometer datasheet. The rightmost column is the one to read first.

Two observations about Figure 2.3 that generalise to almost every sensor datasheet you will meet.

Every number is "typ". There is no minimum and no maximum. You are being told the *average of a production run*, not a guarantee. If your design works only with a typical part, then roughly half of your production does not work — and you will discover this at volume, which is the worst possible time.

The conditions differ from row to row. Sensitivity is quoted at 25 °C. The temperature coefficient is quoted over a 125 °C span. Noise density is quoted at one specific bandwidth. **As printed, these rows are not comparable with one another**, and they are certainly not comparable with the corresponding rows of a competitor's datasheet until you have checked that the conditions match.

So read a datasheet right to left: **conditions first, parameter name second, and the number last.** Most people read it in exactly the opposite order, and that is how they get surprised in February.

2.6 Worked calculation: the error budget

We now have everything needed to decide between Part A and Part B. There are four terms.

Term 1 · Noise

A noise-density figure is not yet a noise figure. It becomes one only when you choose a bandwidth:

```
noise_RMS = noise density × √bandwidth
```

We are measuring a static orientation, so we can afford a modest bandwidth. Take 50 Hz. Both parts specify 100 µg/√Hz:

```
100 µg/√Hz × √(50 Hz) = 100 × 7.07 = 707 µg = 0.707 mg
```

Check the units, because unit cancellation — not the physics — is what usually goes wrong here: µg/√Hz × √Hz = µg. If your units do not cancel, you have used the wrong number.

Note carefully what we did *not* use. The datasheet quoted this figure at an ODR of 200 Hz with a bandwidth of 50 Hz. Using √200 would give 1.41 mg — a wrong answer that looks entirely reasonable. **Bandwidth, not ODR.**

There is a useful corollary. Noise is not a property of the sensor alone; it is a property of the sensor *and the bandwidth you chose*. Halve your bandwidth and you improve noise by √2, for free, if the signal allows it. Widen it to chase fast events and you pay in noise. This is the same trade the motor's monitor got wrong in Chapter 1, seen from the other side.

Term 2 · Quantisation

Rounding to the nearest code contributes an RMS error of one least-significant bit divided by √12:

```
Part A: 0.061 / 3.464 = 0.018 mg Part B: 0.488 / 3.464 = 0.141 mg
```

Both are negligible against 8.73 mg. Note what this means: Part A's celebrated eight-times-finer resolution buys it an advantage of about 0.12 mg on a term that was already invisible.

Term 3 · Offset

Part A's zero-*g* offset is ±40 mg; Part B's is ±10 mg. Both are far larger than our 8.73 mg measurand — and both **go to zero**, because we calibrate the instrument once on the bench at 20 °C and subtract what we find. This is the calibration column, and it is why a large raw offset need not disqualify a part.

Term 4 · Offset drift

Here is the term nobody computed.

The offset does not stay where you calibrated it. It moves with temperature, at a rate given by the temperature coefficient. We calibrate at 20 °C and operate from 0 to 40 °C, so the worst-case excursion is ΔT = ±20 °C:

```
Part A: 0.5 mg/°C × 20 °C = 10.0 mg Part B: 0.1 × 20 = 2.0 mg
```

Part A drifts by 10 mg. The entire quantity we are trying to measure is 8.73 mg. The error is larger than the signal, and it appears *after* the calibration, so no bench procedure catches it.

Combining the terms

These four error sources are independent, so they add in quadrature — root-sum-square:

$$\varepsilon_total = \sqrt{(\ \varepsilon_noise^2 + \varepsilon_quant^2 + \varepsilon_drift^2\ )}$$

Term	Part A	Part B
Noise, 100 µg/√Hz × √50 Hz	0.707 mg	0.707 mg
Quantisation, LSB/√12	0.018 mg	0.141 mg
Offset drift, TC × ΔT	**10.000 mg**	**2.000 mg**
Total (RSS)	**10.02 mg**	**2.13 mg**
As a tilt angle, arcsin(mg/1000)	**0.574°**	**0.122°**
Against the ±0.5° requirement	**FAILS**	**passes, 4× margin**

Table 2.2 — The error budget. Bandwidth 50 Hz, calibrated at 20 °C, operated over 0–40 °C.

Look at what root-sum-square does. Combining 10 mg with 0.707 mg gives 10.02 mg: the noise term we spent half a page computing changed the answer by three hundredths of a milli-*g*.

> **In any error budget, find the largest term first. Everything else is decoration.**

The verdict

Part A resolves eight times finer than Part B and cannot do the job. Part B's 0.488 mg/LSB already resolves 0.028° of tilt — eighteen times finer than the half-degree we need — so resolution was never the deciding variable. It only looked like it was, because it was the number printed on the front page.

And note the incentives. Part A was cheaper, had finer resolution, and carried a marketing line. Every visible signal pointed at the part that fails.

If you chose "I cannot decide from the front pages alone" at the start of this chapter, you were right, and right for the best possible reason. If you chose Part A, you did what most engineers do. If you now choose Part B, you can defend it with a number — which is the only defence that survives a design review.

2.7 The part on your bench

Parts A and B were built to make the arithmetic clean. Now do it for real, on the device you will wire up in the laboratory: the **ST ISM330DHCXTR**, an industrial-grade 6-axis inertial module. Here is what its datasheet actually says about the accelerometer.

Parameter	Min	Typ	Max	Unit
Sensitivity, ±2 g full scale		0.061		mg/LSB
Sensitivity tolerance	−2		+2	%
Acceleration noise density, high-performance mode		**60**	**100**	µg/√Hz
Zero-g level offset accuracy	−65	**±10**	+65	mg
Zero-g level change vs. temperature	−0.5	**±0.1**	+0.5	mg/°C
Sensitivity change vs. temperature, −40 to +105 °C	−0.01	±0.005	+0.01	%/°C
Cross-axis sensitivity, **at T = 25 °C**		±0.5		%
Operating temperature range	−40		+105	°C

Table 2.3 — ISM330DHCX accelerometer characteristics, from datasheet DS13012 Rev 6. Note that the two columns that decided our design — noise density and zero-g temperature drift — each carry both a typical *and* a maximum value.

Now run the same error budget twice: once with the typical column, once with the maximum. Bandwidth 50 Hz, calibrated at 20 °C, operated 0–40 °C, so ΔT = ±20 °C, exactly as before.

Term	Using typ	Using max
Noise, density × √50 Hz	0.424 mg	0.707 mg
Quantisation, 0.061/√12	0.018 mg	0.018 mg
Offset drift, TC × 20 °C	**2.000 mg**	**10.000 mg**
Total (RSS)	**2.05 mg**	**10.02 mg**
As a tilt angle	**0.117°**	**0.574°**
Against the ±0.5° requirement	**passes, 4× margin**	**FAILS**

Table 2.4 — The same part, the same requirement, the same arithmetic. One datasheet column apart.

Read that again, because it is the most important table in this chapter.

This is one device. It is not a comparison between a good part and a bad one. A typical ISM330DHCX meets our tilt requirement comfortably; a part at the limit of its own published specification **misses it** — and both parts are entirely within specification, both would pass incoming inspection, and both would be shipped to you in the same reel.

So what do you actually do?

- **If you must guarantee the specification**, you budget with the **max** column. On that basis this design does not close, and you need a different approach: tighten the temperature range, measure temperature and compensate the offset (Chapter 14), or relax the ±0.5° requirement.
- **If you are building ten units** and can measure each one, the typical column plus a per-unit calibration is a defensible engineering decision — provided you write down that you made it.
- **What you must not do** is budget with `typ` and describe the result as a guarantee. That is the difference between an engineering document and a hope.

Notice also what our fictional parts were really doing. Part A's figures (±40 mg offset, ±0.5 mg/°C, 100 µg/√Hz) were essentially this real part's **worst case**, and Part B's (±10 mg, ±0.1 mg/°C) were its **typical case**. The lesson has not changed; it has just become harder to dismiss, because it is now a single part number rather than a choice between two.

> **A datasheet does not describe your device. It describes the population your device was drawn from.**

2.8 What the datasheet cannot tell you

Look again at the cross-axis row of Table 2.3. The ISM330DHCX quotes ±0.5 % — a real, published number. Now ask a slightly different question:

> **What is this device's cross-axis sensitivity after it has been soldered onto my board?**

That is not there, and it cannot be. Look at the condition attached to the published figure: **T = 25 °C**, for the packaged die, as measured by the manufacturer. But board-level mounting stress, solder-joint asymmetry, thermal gradients across the PCB and flexing of the board all change it — and the manufacturer cannot know what your board does to their part.

The honest engineering answer is therefore in the first person: *the datasheet cannot tell me this; I must measure it on my own assembled board.* Installing that habit is the most important thing this chapter does, and it is why the laboratory programme includes characterisation work rather than only bring-up work.

The same applies to the mechanical coupling of Chapter 1, to the thermal environment inside your enclosure, and to the actual noise you achieve after routing a sensitive analog trace next to a switching regulator. **The datasheet describes a component. You are building a system.**

2.9 Selecting, and over-selecting

Selection is a calculation against a specification, documented so that somebody else can check it. A simple criteria table is enough, provided the weights are honest.

Consider a third candidate. **Part C**: 16-bit, ±2 g, 60 µg/√Hz, offset ±5 mg, temperature coefficient ±0.05 mg/°C, 0.9 mA, €11.50.

Criterion	Weight	A	B	C
Meets ±0.5° over 0–40 °C	pass/fail	fail	pass	pass
Total error as an angle	high	0.574°	0.122°	0.06°
Unit price at 500 off	medium	€1.80	€4.20	€11.50
Current draw	medium	0.15 mA	0.18 mA	0.90 mA
A library already exists	**low**	yes	no	no

Table 2.5 — A criteria table. Note the weight on the last row.

Part C is technically the best part in the table. It is also **the wrong answer**, and for an instructive reason: at 500 units it costs €3 650 more than Part B, draws five times the current, and buys no capability that the requirement asks for. Over-specification is an engineering failure too — a quieter one than under-specification, but you still have to explain it.

Which brings us to the last row, and to a rule this course applies throughout:

> **Never choose a component only because a software library exists for it.**

A library is a genuine asset and it may save you a week. It is not a specification. Rank it last, not zero. Part A had a library; Part A cannot do the job.

Choose instead on the features that expose the engineering you need to do: selectable range and output data rate, documented bandwidth and noise, a self-test function, a data-ready interrupt, a FIFO where relevant, an accessible register map, and a complete datasheet.

2.10 Reading

The core of this chapter corresponds to **Morris & Langari, *Measurement and Instrumentation*, 3rd edition (Elsevier, 2020), Chapter 2**, "Instrument Types and Performance Characteristics". Read it for the formal definitions of the static and dynamic characteristics in Table 2.1, and for its worked examples.

Chapter 3 ("Measurement Uncertainty") and **Chapter 4** ("Statistical Analysis of Measurements Subject to Random Errors") develop error combination — including the root-sum-square rule used in Table 2.2 — much more fully than we have here. We return to that material in Chapter 14; a first reading now will make the error budget above feel less like a recipe.

Then read the electrical-characteristics section of the **ISM330DHCX datasheet (DS13012)** — the part on your own bench. That is not an optional extra: from this chapter onward every laboratory assumes you can find a number, and its conditions, in a real document. Table 2.3 above is where you should end up; arriving there yourself is the skill.

Chapter and section numbering should be checked against the edition available to you.

2.11 Exercises

2.1 Define accuracy, precision and resolution so that the three definitions cannot be confused with one another. Give an example of a device that is precise but inaccurate.

2.2 Which of the following can a single-point bench calibration remove: offset, random noise, temperature drift, quantisation? Explain each answer.

2.3 A sensor specifies 150 µg/√Hz. You need 20 Hz of bandwidth. (a) Compute the RMS noise. (b) State what happens if you widen the bandwidth to 80 Hz, and by what factor.

2.4 A datasheet gives an offset temperature coefficient of ±0.3 mg/°C. The device is calibrated at 25 °C and used from −10 to +60 °C. Compute the worst-case offset error from drift alone, and express it as a tilt angle.

2.5 Your requirement is a signal of 8.73 mg. A candidate part offers 0.061 mg resolution and 10 mg of offset drift over your temperature range. In one sentence, explain what is wrong with calling this a "high-resolution solution".

2.6 A sensor is quoted as "±0.05 % of full scale" accuracy at 25 °C, with no temperature coefficient given anywhere in the datasheet. Full scale is 20 kPa. (a) What is the accuracy in pascals at 25 °C? (b) Why can you *not* use this part in an error budget for outdoor operation, even though its quoted accuracy is excellent?

2.7 Give one performance-limiting property that a datasheet cannot specify for your finished product, and say who must determine it.

2.8 You must measure water level in an outdoor tank 2 m deep, to ±20 mm, with water between 5 and 35 °C. Sensor P has a total error band of ±0.25 % of full scale over 0–50 °C. Sensor Q is specified as ±0.05 % of full scale at 25 °C, with no temperature coefficient stated. Full scale for both is 20 kPa. Which do you specify, and why? (1 mm of water ≈ 9.81 Pa.)

2.9 Using Table 2.3, compute the ISM330DHCX's RMS noise at a bandwidth of 200 Hz, for both the typical and the maximum noise density. Express each as a tilt angle.

2.10 The ISM330DHCX's sensitivity tolerance is ±2 %. You measure 1 g of gravity and the device reports 1015 mg. (a) Is this part within specification on sensitivity? (b) Which single bench measurement would let you correct this error, and what would remain uncorrected afterwards?

2.11 An accelerometer offers selectable ranges of ±2 g, ±4 g, ±8 g and ±16 g at a fixed 14-bit output. (a) Compute the resolution in mg/LSB at each range. (b) You need to resolve 5 mg and survive 12 g shocks without clipping. Which range do you configure, and what is the consequence for resolution?

Answers

2.1 Accuracy is closeness of a reading to the true value — a systematic property. **Precision** is the repeatability of readings about their own mean — a random-scatter property, which says nothing about truth. **Resolution** is the smallest change the output is able to represent, which is a property of the reporting scale and not of correctness at all. Example: a sensor with a large uncorrected offset but very low noise, such as the 1012.4 hPa device in §2.3.

2.2 Offset — yes, at the temperature at which you calibrated. **Random noise — no**; only averaging reduces it, and averaging costs bandwidth. **Temperature drift — no**; a single point captures a single temperature, so removing drift requires measuring temperature and applying a compensation model. **Quantisation — no**; it is fixed by the LSB, although dithering and averaging can recover some detail below one LSB.

2.3 (a) 150 × √20 = 150 × 4.472 = 671 µg ≈ **0.67 mg**. (b) 150 × √80 = 1342 µg ≈ **1.34 mg** — exactly double, because the bandwidth quadrupled and noise follows its square root.

2.4 The worst-case excursion from the 25 °C calibration point is to 60 °C, so ΔT = 35 °C. Drift = 0.3 × 35 = **10.5 mg**, which as a tilt is arcsin(0.0105) ≈ **0.60°**.

2.5 The resolution is roughly 140 times finer than the measurand while the drift alone exceeds the entire measurand, so the part resolves — very precisely — a number that is wrong; resolution is not accuracy.

2.6 (a) 0.05 % of 20 kPa = **10 Pa**. (b) Because the figure is guaranteed only at 25 °C, and no temperature coefficient is published, the error at any other temperature is not merely large — it is **unknown**. An unspecified coefficient is not a small error, it is an unbounded one, and you cannot put an unbounded term into a budget you have to sign.

2.7 Any board-level effect: cross-axis sensitivity after reflow soldering, package stress from PCB flex, thermal gradients across the assembly, mounting compliance, or noise picked up from your own power supply. The **integrator** must determine it, by measurement on the assembled unit.

2.8 Sensor P. Its ±0.25 % of 20 kPa = 50 Pa ≈ **5.1 mm**, comfortably inside the ±20 mm requirement and, critically, *guaranteed across the whole operating temperature range*. Sensor Q's ±0.05 % = 10 Pa ≈ 1.0 mm at 25 °C, but its behaviour at 5 °C is unknown because no temperature coefficient is specified. You cannot write a defensible error budget for Q at all. A total error band over the operating range beats a headline accuracy at a single temperature.

2.9 At 200 Hz: typ = 60 × √200 = 60 × 14.14 = **849 µg ≈ 0.85 mg**, which as a tilt is arcsin(0.00085) ≈ **0.049°**. Max = 100 × √200 = **1414 µg ≈ 1.41 mg**, or ≈ **0.081°**. Quadrupling the bandwidth from 50 Hz doubled the noise, as expected.

2.10 (a) ±2 % of 1000 mg is ±20 mg, so a reading of 1015 mg is **within specification** — it is a 1.5 % scale-factor error. (b) A **known-reference measurement**: with the axis aligned to gravity you know the true value is 1 g, so dividing by 1.015 removes the scale-factor error. What remains uncorrected is the *change* of sensitivity with temperature (±0.005 %/°C typ), the zero-g offset drift, and the noise — a single-point scale calibration fixes the gain at one temperature and nothing else.

2.11 (a) 14 bits gives 16 384 codes across the full span, so resolution = 2 × range / 16 384: **±2 g → 0.244 mg/LSB**; ±4 g → 0.488; ±8 g → 0.977; **±16 g → 1.953 mg/LSB**. (b) Surviving 12 g without clipping forces **±16 g**,

giving 1.953 mg/LSB. Since quantisation contributes $1.953/\sqrt{12} = 0.56$ mg RMS, resolving 5 mg is still comfortable — so in this case the range requirement costs you nothing that matters. That is the point of doing the arithmetic rather than assuming the trade-off hurts.

3 · From a physical quantity to trustworthy samples

What you should be able to do after this chapter

1. **Convert** a raw two's-complement register pair into a value in SI units, stating the sensitivity, the sign convention and the byte order you assumed.
2. **Distinguish** output data rate, measurement bandwidth and anti-alias cut-off frequency, and say which of the three, if set wrongly, produces an error that cannot be undone.
3. **Compute** the noise-limited effective resolution of an acquisition chain, and state how many of the bits your datasheet advertises actually carry information.
4. **Choose** between polling and a data-ready interrupt for a stated measurement, and justify the choice in terms of timestamp error rather than convenience.
5. **Classify** the errors in an acquisition chain as calibratable or irreversible.

3.1 One log file, and three numbers

Chapter 2 ended with a part chosen and defended. This chapter begins with that part on the bench.

You have configured it. The register map is open, the sensitivity is written on a sticky note, and a `while()` loop is filling a CSV file with numbers. The measurement looks finished — the difficult judgement was the selection, and that is behind you.

Here is the configuration, and it is the one Laboratory 2 will use:

> An **ISM330DHCX** accelerometer, configured for **±2 g full scale, 16-bit output, ODR = 104 Hz**. Its datasheet gives the sensitivity as **0.061 mg/LSB**. You log for ten minutes by calling `HAL_Delay(10)` in a loop and reading six bytes over I²C each time.

Now the question this chapter exists to answer:

> **How much of what you logged is true?**

Not "is the sensor good" — Chapter 2 settled that. The sensor is fine. The question is what survived the journey from the silicon to the file, and the answer is three numbers:

12.4	the number of bits, out of sixteen, that carry information
12 Hz	the frequency of a signal in your data that does not exist in the world
12.1 s	how wrong your timestamps are by the end of the log

Every one of those three was computable before a line of firmware was written, from the same datasheet you already had open. None of the three announces itself in the data. And two of the three cannot be repaired afterwards at any price.

That is the shape of this chapter: acquisition does not fail loudly. It fails by handing you a file full of plausible numbers.

3.2 What comes out of a sensor

Two kinds of thing, and the boundary between them is where this chapter lives.

An analog output is a continuous electrical quantity proportional to the measurand. It comes in three common forms:

- **Voltage.** The simplest and the most fragile. A thermocouple gives tens of microvolts per degree; a bridge gives a few millivolts. Both are small enough that the wire between the sensor and the amplifier is part of the measurement.
- **Current, usually 4–20 mA.** An industrial convention worth understanding, because it solves two problems at once. Current is unchanged by the resistance of a long cable, so the reading does not depend on how far away the sensor is; and because the *zero* of the scale is 4 mA rather than 0 mA, a broken wire reads 0 mA and is therefore distinguishable from a genuine zero. That is a designed-in fault detection, and it is the reason the convention has outlived the technology that prompted it.
- **A bridge.** Four resistive elements in a diamond, excited by a voltage, producing a difference voltage proportional to the imbalance. Strain gauges, piezoresistive pressure sensors and many load cells are bridges. Section 3.3 is about a property of bridges that most people discover by accident.

A digital output means the analog-to-digital conversion has already happened, inside the sensor's own package, and what you receive is a number over I²C or SPI. Almost every MEMS sensor you will meet in this course is of this kind.

It is tempting to conclude that a digital output removes the problems of an analog one. It does not. It **moves them inside the package, where you cannot see them and cannot change them** — and it adds two new ones of its own, which are the byte order of the result and the question of *when* the conversion happened. The rest of this chapter is largely about those two.

3.3 The reference you forgot

Every analog-to-digital converter answers one question: *what fraction of my reference voltage is this input?* The count it returns is

$$\text{count} = \frac{V_{in}}{V_{ref}} \times 2^{N}$$

Read that formula for what it says. The count does not depend on the input. It depends on the **ratio** of the input to the reference. An ADC has no idea what a volt is.

This has a consequence that decides whether an entire class of measurement works.

Consider a strain-gauge bridge excited from the microcontroller's 3.3 V rail, read by an ADC that uses **the same rail** as its reference. Under load the rail sags to 3.2 V — a 3 % drop. What happens to the reported strain?

Nothing. The bridge output is proportional to its excitation, so it falls by 3 %. The ADC's counts are proportional to $1/V_{ref}$, so they rise by 3 %. The two cancel exactly, because the quantity the code computes was a ratio all along. This is a **ratiometric measurement**, and it is free.

Now change one wire. Give the ADC a precision 2.500 V reference and leave the bridge on the rail. The sag no longer cancels: it becomes a 3 % error in the reported strain, it varies with the load on the rail, and **no calibration can remove it**, because it is not a constant — it is a function of whatever else the board is doing.

Same components. Same cost. One wire's difference between a measurement that is immune to supply variation and one that is at its mercy.

The rule is short enough to remember: **excite the sensor and reference the converter from the same thing, or from two things that are both stable.** Mixing one stable and one unstable is the only combination that fails, and it is the one that looks most sophisticated on a schematic.

3.4 Quantisation: the error that does not matter here

The converter has a finite number of codes, so it must round. For our accelerometer:

$$\text{LSB} = \frac{\text{full scale}}{2^{N}} = \frac{4000\ \text{mg}}{65\,536} = 0.061\ \text{mg}$$

which is exactly what the datasheet prints. Rounding to the nearest code introduces an error uniformly distributed over one LSB, whose RMS value is

$$\sigma_{q} = \frac{\text{LSB}}{\sqrt{12}} = \frac{0.061}{3.464} = 0.018\ \text{mg}$$

The $\sqrt{12}$ is the standard deviation of a uniform distribution of unit width; it is worth knowing where it comes from, and it is not worth deriving twice.

Hold on to 0.018 mg. In Section 3.7 we will compare it with the sensor's own noise and find it smaller by a factor of forty. **Almost every argument you will hear about the last bit of an ADC is, in a real sensor system, an argument about nothing.** Quantisation is the error engineers discuss because it is easy to calculate, and it is very rarely the error that decides anything.

3.5 Output data rate is not bandwidth

Here are three numbers that students, datasheets and marketing material routinely conflate. They have three different jobs.

	What it is	Who sets it
ODR — output data rate	how often a new number appears at the output	you, by register
Measurement bandwidth	the highest signal frequency the device reports faithfully	you, by register — but often a *different* register
Anti-alias cut-off	the frequency above which the filter *before the sampler* attenuates	the designer of the analog path — sometimes you, sometimes nobody

The Nyquist–Shannon sampling theorem says a signal is recoverable only if it is sampled at more than twice its highest frequency component. Chapter 1 showed what happens when it is violated: a 1520 Hz bearing tone, sampled at 100 Hz, appeared as a perfectly convincing 20 Hz oscillation.

Chapter 1 presented that as a discovery. This chapter presents it as a **design rule with a register bit attached**, which is a different thing entirely.

The ordering that cannot be repaired

The anti-alias filter must sit **before** the sampler. Not after. There is no software equivalent.

The reason is arithmetic rather than engineering. Once a 300 Hz component has been sampled at 104 Hz, it is represented in the data by exactly the same sequence of numbers as a genuine 12 Hz component. They are not similar; they are *identical*. No filter, no transform, no amount of computation can separate two things that are the same. The information was destroyed at the instant of sampling, and it was destroyed by the *absence* of a component costing a few cents.

This is the only ordering constraint in the whole measurement chain with that property. Everything else — a wrong gain, a missing offset correction, a sign error, a unit muddle — can be fixed on a laptop a week later, by someone who was not there. This one cannot.

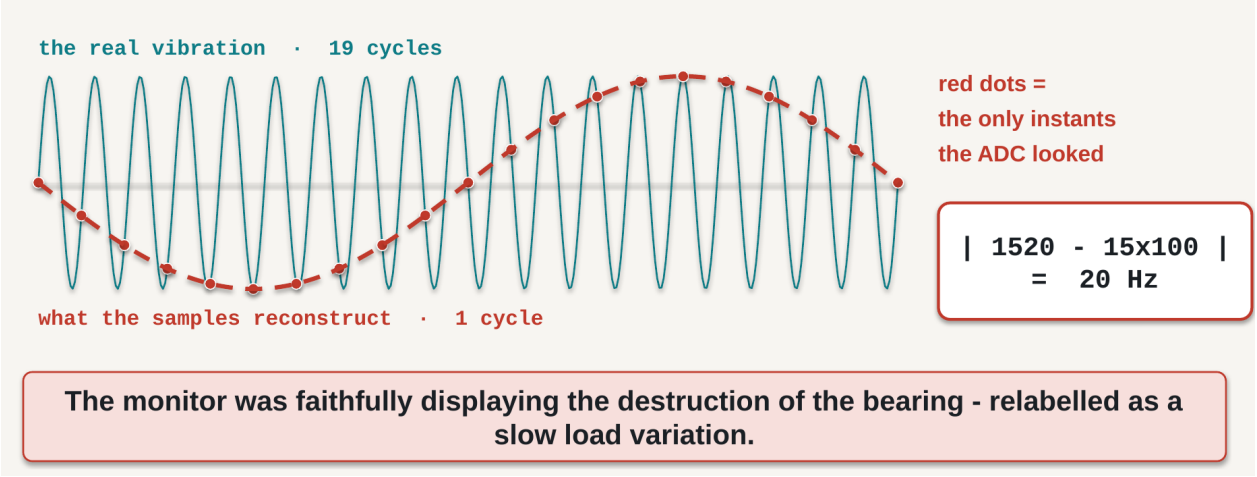

Figure 3.1 — The mechanism, from Chapter 1. The samples are entirely correct; their *interpretation* is not. Reproduced here because this chapter is where it becomes a register setting rather than a cautionary tale.

The register bit that catches everybody

The ISM330DHCX's `CTRL1_XL` register holds the ODR selection in bits 7:4, the full-scale selection in bits 3:2, and in bit 1 a flag named `LPF2_XL_EN`.

After reset, that bit is 0. The second low-pass filter is disabled.

So the natural sequence of events is this. A student sets ODR = 104 Hz, reasons correctly that the Nyquist limit is 52 Hz, concludes that the measurement bandwidth is 52 Hz, and writes it in the report. The device, meanwhile, is sampling a much wider analog band with no anti-alias filtering, and everything above 52 Hz is folding back into the data.

Suppose a pump on the same frame runs at 300 Hz:

$$f_{alias} = |300 - 3 \times 104| = |300 - 312| = 12\ \text{Hz}$$

Twelve hertz. Squarely inside the band that a tilt or vibration measurement cares about, comfortably slow enough to look like real mechanical behaviour, and permanent.

It gets worse if the numbers are rounder. At an ODR of 100 Hz:

$$f_{alias} = |300 - 3 \times 100| = 0\ \text{Hz}$$

The pump becomes a **DC offset**. Somebody will now spend a productive week calibrating it out, and will succeed — at that one pump speed. The calibration will be wrong at every other speed, and the failure will be blamed on the sensor.

And Chapter 1's bearing, at this ODR:

$$f_{alias} = |1520 - 15 \times 104| = 40\ \text{Hz}$$

Same story, different number. The lesson is not that 104 Hz is a bad choice. It is that **the sample rate alone tells you nothing about what is in your data.**

3.6 From raw codes to a physical quantity

Four steps, each with a way to get it wrong that produces a plausible answer.

Suppose a burst read of the X-axis returns two bytes:

`OUTX_L_A (0x28) = 0x2C OUTX_H_A (0x29) = 0xFF`

Step 1 — assemble the word, in the right order. The device is little-endian: the low byte comes from the lower address. So the 16-bit word is `0xFF2C`, not `0x2CFF`.

Step 2 — interpret the sign. The value is 16-bit two's complement. Bit 15 is set, so the number is negative:

$$0\text{xFF2C} = 65\,324 \quad \rightarrow \quad 65\,324 - 65\,536 = -212\ \text{counts}$$

Step 3 — apply the sensitivity.

$$a = -212 \times 0.061\ \text{mg} = -12.9\ \text{mg}$$

Step 4 — convert to SI.

$$a = -0.0129\ g \times 9.80665 = -0.127\ \text{m/s}^2$$

Now the four failures, all of which appear in real laboratory submissions:

- **Wrong byte order.** Reading high-then-low gives `0x2CFF` = 11 519 counts = **+703 mg**. Not obviously absurd. A wrong answer of a believable magnitude is far more dangerous than one that is obviously broken.
- **Sign ignored.** Treating the word as unsigned gives 65 324 counts = **+3985 mg**: a stationary device reporting nearly 4 g. This one at least announces itself, provided somebody looks.
- **Auto-increment misunderstood.** A burst read of six consecutive registers requires the address pointer to advance, controlled by `IF_INC` in `CTRL3_C`. That register resets to `0x04`, so **auto-increment is already enabled** and a burst read works out of the box. The failure mode is the diligent student who writes `0x00` to `CTRL3_C` to "start from a clean state", and then reads the same byte six times.
- **Units left implicit.** mg, g and m/s² all appear in the same laboratory. Put the unit in the column header, once, and check plausibility against the one reference every student has: at rest, one axis should read about 9.81 m/s², and the vector sum of all three should be 1 g whatever the orientation.

That last check is worth more than it looks. It is the only step in this whole chain that tests the *entire* chain at once, against a physical constant you cannot misconfigure.

3.7 Worked calculation: the bits you own

We can now answer the first of the chapter's three numbers, and we do it entirely from figures already in hand.

Given. ±2 g full scale, 16-bit output, ODR = 104 Hz with `LPF2_XL_EN` set so that the measurement bandwidth is ODR/2. Noise density 100 µg/√Hz — the datasheet's *max* column.

Step 1 — the quantisation step.

$$\text{LSB} = \frac{4000\ \text{mg}}{65\,536} = 0.061\ \text{mg}$$

Step 2 — the quantisation noise.

$$\sigma_{q} = \frac{0.061}{\sqrt{12}} = 0.018\ \text{mg}$$

Step 3 — the bandwidth.

$$BW = \frac{ODR}{2} = \frac{104}{2} = 52\ \text{Hz}$$

Step 4 — the sensor noise in that bandwidth.

$$\sigma_{n} = 100\ \mu g/\sqrt{\text{Hz}} \times \sqrt{52\ \text{Hz}} = 100 \times 7.211 = 721\ \mu g = 0.721\ \text{mg}$$

Step 5 — combine them. They are independent, so root-sum-square:

$$\sigma_{total} = \sqrt{0.721^2 + 0.018^2} = 0.721\ \text{mg}$$

The quantisation term has changed the answer in the fourth decimal place. This is the promised comparison from Section 3.4: **the sensor noise is forty times the quantisation noise.** The converter is not the limitation and never was.

Step 6 — express the noise in codes.

$$\frac{0.721\ \text{mg}}{0.061\ \text{mg/LSB}} = 11.8\ \text{LSB}$$

Step 7 — count the bits. A noise floor of 11.8 codes means the bottom

$$\log_{2}(11.8) = 3.6\ \text{bits}$$

of every sample are noise. Therefore

$$\text{effective bits} = 16 - 3.6 = \mathbf{12.4}$$

The verdict. You bought a 16-bit number wrapped around a 12.4-bit measurement. The smallest acceleration change you can actually distinguish in a single sample is not 0.061 mg but **0.72 mg**, twelve times larger.

And the datasheet did not lie to you. It printed the noise density on the same page as the resolution. It simply did not do the multiplication, because the bandwidth was your choice, not theirs.

Now run it again with the other column. With the `typ` noise density of 60 µg/√Hz:

$$\sigma_{n} = 60 \times \sqrt{52} = 433\ \mu g = 0.433\ \text{mg} = 7.1\ \text{LSB} \quad \rightarrow \quad 2.8\ \text{bits} \quad \rightarrow \quad \mathbf{13.2\ bits}$$

The datasheet's own two columns move the answer by 0.8 bits. That is Chapter 2's lesson — *a number without its conditions is not a number* — restated in a different quantity, and it will recur in every remaining chapter.

A note on names

Three quantities in this area have similar names and different definitions, and a reader who meets them elsewhere should not be confused:

- **Effective resolution**, computed above, compares the total noise with the LSB. It answers "how many bits mean anything?"
- **ENOB** (effective number of bits), as datasheets for standalone ADCs use it, compares the total noise with the *ideal quantisation* noise: $16 - \log_2(0.721/0.018) = 10.7$ bits here. It answers "how much worse than a perfect converter of this width is this one?"
- **Noise-free resolution** uses a peak-to-peak noise figure, conventionally 6.6σ, and comes out lower again.

All three are defensible; none is interchangeable. Whenever you quote one, name it. This chapter uses effective resolution throughout because it is the one that answers a design question.

3.8 When did it happen?

The third of the chapter's numbers, and the one most often discovered too late.

You called `HAL_Delay(10)`, so the samples are 10 ms apart and the rate is 100 Hz. Except that the loop also has to read the sensor. A six-byte burst read over I²C at 400 kHz is about 81 bit-times of protocol:

$$t_{I^{2}C} = \frac{81}{400\,000} = 202\ \mu s$$

So the loop period is not 10.000 ms. It is

$$T = 10.000 + 0.202 = 10.203\ \text{ms} \quad \rightarrow \quad f = 98.0\ \text{Hz}$$

Two consequences, both quantitative.

Frequencies come out wrong. A genuine 20 Hz vibration advances $20 \times 10.203\ \text{ms} = 0.2041$ of a cycle per sample. Analysed on the assumption of 10.000 ms spacing, that reads as

$$\frac{0.2041\ \text{cycles}}{0.010\ \text{s}} = 20.41\ \text{Hz}$$

a 2 % error, which is enough to misidentify a machine order or a resonance.

Time itself comes out wrong, and cumulatively. After 60 000 samples your timestamps, generated as $n \times 10$ ms, say that 600.0 s have elapsed. In reality

$$60\,000 \times 10.203\ \text{ms} = 612.1\ \text{s}$$

have passed. **Every event in the last part of that log is stamped 12.1 seconds earlier than it happened.** Correlate that log against anything else — a second sensor, an operator's note, a video, another team's data — and the correlation is meaningless.

It is worse than a constant scaling error, because the loop period is not actually constant: it depends on which branches ran, whether an interrupt arrived, and what the bus was doing. You can *bound* the error. You cannot invert it.

The fix, which is free

Configure the sensor's **data-ready interrupt**. Read the sample in response to it, and timestamp it from a hardware timer rather than from a loop counter. The sample interval is then set by the sensor's own internal timebase, and your timestamp error becomes that timebase's tolerance — a specified quantity, in the datasheet, typically a fraction of a percent — instead of a property of your software's mood.

Many devices, the ISM330DHCX among them, go further and offer a timestamp register and a FIFO, so that samples are stamped at the source and read in batches. If you take one habit from this chapter into the laboratory, take this one: **the sensor knows when it sampled; ask it, rather than guessing.**

	Polling with a delay	Data-ready interrupt
Sets the interval	your software	the sensor's timebase
Interval error	loop-dependent, unbounded	datasheet-specified
Cumulative time error	grows without limit	bounded by the timebase tolerance
CPU cost	high — busy waiting	low
Effort to implement	slightly less	slightly more

3.9 Knowing that the data is real

Four cheap checks, each of which catches a failure that otherwise presents as bad data rather than as an error.

- **Device identity.** Read the `WHO_AM_I` register and compare it against the expected constant — `0x6B` for the ISM330DHCX. This confirms the address, the bus, the pull-ups and the power supply in one transaction. Note in passing that the expected value `0x6B` is also one of the part's two I²C addresses, which has confused a great many people; they are unrelated coincidences of numbering.
- **Self-test.** Most MEMS sensors can electrostatically deflect their own proof mass by a known amount. The output should change by a specified range. This tests the mechanical structure, not just the electronics — the only check in this list that does.
- **FIFO overrun.** If the buffer overflowed, samples were lost, and the ones you have are not evenly spaced. A silently dropped sample is a timing error disguised as data. Read the status flag; do not assume.
- **Bus error and recovery.** I²C can hang with a slave holding the data line low. A design that cannot recover — by clocking the bus until it releases, or by cycling power — will one day stop logging and give no reason.

None of these is difficult. All four are omitted from most first attempts, and all four turn "the data looks strange" into "the sensor was never responding".

3.10 A datasheet case: two registers decide the chapter

Everything in Sections 3.5 to 3.8 comes down to bits in two registers of the part on your bench.

`CTRL1_XL` (0x10)

Bits	Field	What it decides
7:4	`ODR_XL`	the output data rate — `0100` is 104 Hz
3:2	`FS_XL`	full scale, and therefore the LSB and the noise in mg
1	`LPF2_XL_EN`	**whether there is an anti-alias filter at all** — 0 after reset
0	—	must be written 0

Two things are worth noticing. First, the full-scale encoding is **not in ascending order**: `00` = ±2 g, `01` = ±16 g, `10` = ±4 g, `11` = ±8 g. A student who assumes the obvious mapping selects ±16 g while believing they selected ±16 g's neighbour, and every subsequent number is wrong by a factor of two or four. Read the table; do not infer it.

Second, `0x40` gives ±2 g at 104 Hz — and leaves bit 1 clear, which is the configuration discussed at length in Section 3.5.

`CTRL3_C` (0x12) resets to `0x04`, which means `IF_INC` is already set. Burst reads work by default. This is the register the diligent student breaks.

The whole of this chapter, then, is contained in about twelve bits. That is characteristic of embedded sensing, and it is why the course insists on register-level configuration rather than a library call: the library sets those bits too, but it does not tell you which ones it chose.

3.11 An integration failure: the log that was thrown away

A team instrumented a test rig with two sensors: an accelerometer on the frame at 104 Hz, and a pressure transducer on the hydraulic line read by the microcontroller's own ADC at "the same rate". Both logged to the same file, one row per loop iteration, with a single timestamp column generated from a loop counter.

The purpose was to find out whether a pressure spike preceded or followed a mechanical shock. That is a question about ordering, at millisecond resolution.

Three problems, none of which produced an error message:

1. The accelerometer was read over I²C and the pressure transducer through the internal ADC, so the two readings in each row were taken about 200 µs apart. Nothing in the file recorded that.
2. The loop period was 10.2 ms, not 10.0 ms, so the timestamps drifted — by twelve seconds over a ten-minute run.
3. The pressure transducer's analog path had no anti-alias filter, and the pump ran at 300 Hz.

The team's conclusion — that pressure spikes followed the shocks by about 15 ms — was arithmetic on all three errors at once. The 15 ms was neither measured nor measurable with that setup. The rig was rebuilt, and the second attempt used the accelerometer's data-ready interrupt to trigger both readings, timestamped from a hardware timer, with an RC filter on the pressure input. The answer changed sign.

The lesson is not that they were careless. Every individual decision was reasonable, and the file looked perfect: no gaps, no outliers, plausible magnitudes, a monotonic timestamp column. **A log file cannot tell you that it is**

wrong. The only defence is to compute what the acquisition chain can and cannot resolve *before* trusting what comes out of it, which is the entire content of this chapter.

3.12 Calibratable, or gone

The table to keep. Everything in the acquisition chain belongs in one of these two columns, and knowing which is the difference between a problem and a disaster.

Error	Removable afterwards?	How, or why not
Offset	**yes**	one-point calibration against a known reference
Scale factor	**yes**	two-point calibration
Non-linearity	**yes**, with effort	a fitted correction, if you characterise it
Wrong byte order	**yes**	reinterpret the file; the bits are all there
Sign error	**yes**	reinterpret the file
Unit muddle	**yes**	a multiplication
Quantisation	**no** — but negligible here	0.018 mg against 0.72 mg of noise
Random noise	**partly**	averaging reduces it by √N, and costs bandwidth
Temperature drift	**partly**	only if temperature was logged too — so log it
Aliasing	**NO**	the alias and the signal are the same numbers
A lost or wrong timestamp	**NO**	the timing information was never recorded
Saturation / clipping	**NO**	the value was outside the range; nothing was stored

Notice the pattern. Everything in the top group is a *transformation* of data that is still present. Everything in the bottom group is *missing information*. No amount of processing creates information, and this is the one law of the subject with no exceptions and no workarounds.

Notice also the third row from the bottom: temperature drift is recoverable *if and only if* you logged the temperature. Almost every sensor in this course has a temperature output, most people ignore it, and it costs one extra column in the file. Log it.

3.13 The four claims a trustworthy sample makes

A sample you can defend answers four questions, and you should be able to answer all four before writing firmware:

1. **What is it?** A value in SI units, with the sensitivity, sign convention and byte order you assumed stated explicitly.
2. **How much of it is information?** The effective resolution, computed from the noise density and *your* bandwidth — 12.4 bits, not 16.

3. **What band does it represent?** The measurement bandwidth, and evidence that an anti-alias filter was present below it. Not the ODR.
4. **When did it happen?** A timestamp whose error you can state, from a source you can name.

A log file that cannot answer these four is not data. It is a plausible file.

3.14 Reading

This chapter is the one place in Module A where the adopted textbook covers the ground more thoroughly than we can in eighty minutes, and you should read it.

- **Morris & Langari, *Measurement and Instrumentation*, 3rd ed., Chapter 6 — "Data Acquisition and Signal Processing".** Sampling, aliasing, ADC architectures and quantisation, at more depth and with more mathematics than we have used. Read it after this chapter, not before: it is a better second pass than first.
- **Morris & Langari, Chapter 3 — "Measurement Uncertainty".** Where the root-sum-square combination of Section 3.7 comes from, properly justified.
- **Kuphaldt, *Lessons in Industrial Instrumentation*** (freely licensed, CC-BY) has the best practical treatment of 4–20 mA current loops in print, including why the convention exists and how it fails. Recommended if Section 3.2 interested you.
- **ISM330DHCX datasheet DS13012 Rev 6**, the `CTRL1_XL`, `CTRL3_C` and `WHO_AM_I` register descriptions, and the accelerometer noise table. This is not background reading; it is the working document for Laboratory 2.

What the textbooks will *not* give you is Section 3.8. The relationship between a polling loop and a timestamp is embedded-systems practice rather than instrumentation theory, it is where a great deal of real sensor data goes wrong, and as far as we can find it is not treated at undergraduate level anywhere. That gap is the reason this reader exists.

3.15 Exercises

3.1 An ADC has a 12-bit output and a 3.300 V reference. What is its LSB in millivolts, and what is its quantisation noise in millivolts RMS?

3.2 The same ADC reads a sensor whose output noise is 4 mV RMS. How many of the twelve bits carry information?

3.3 A sensor is sampled at 200 Hz. An interfering tone exists at 750 Hz. At what frequency does it appear in the data? At what frequency would it appear if the sample rate were changed to 250 Hz?

3.4 A bridge is excited from a 5.000 V precision reference. The ADC reading it uses the 3.3 V rail as its reference, and the rail sags by 2 %. By how much, and in which direction, does the reported value change? Is the error calibratable?

3.5 A 16-bit accelerometer at ±4 g has a noise density of 90 µg/√Hz. You need a measurement bandwidth of 10 Hz. Compute the LSB, the noise in mg, the effective resolution in bits, and state which of quantisation or noise dominates.

3.6 A logger polls a sensor every 20 ms using a delay call. Each read takes 350 µs. Over an eight-hour run, how far out is the final timestamp? Express it in seconds and as a percentage.

3.7 For each of the following, say whether it can be corrected after the data has been logged, and in one sentence why: (a) a 3 % scale-factor error; (b) a 40 Hz component that was really at 360 Hz sampled at 100 Hz; (c) readings that saturated at full scale for 200 ms; (d) an axis whose sign was inverted; (e) 0.5 mg of random noise.

3.8 You must measure a vibration whose highest component of interest is 40 Hz, using a sensor with a selectable ODR of 26, 52, 104, 208 or 416 Hz and a selectable low-pass filter. Choose the ODR and the filter cut-off, and justify both. State what you would do if the filter could not be enabled.

Answers

3.1 $\text{LSB} = 3.300/4096 = 0.806$ mV. $\sigma_q = 0.806/\sqrt{12} = 0.233$ mV RMS.

3.2 The noise is $4/0.806 = 4.96$ LSB, so $\log_2 4.96 = 2.3$ bits are noise and $12 - 2.3 = 9.7$ bits carry information. Note that the sensor noise is seventeen times the quantisation noise: a 10-bit converter would have measured this sensor essentially as well.

3.3 At 200 Hz: the nearest multiple is $4 \times 200 = 800$, so $|750 - 800| = 50$ Hz. At 250 Hz: $3 \times 250 = 750$, so $|750 - 750| = 0$ Hz — the tone becomes a DC offset, which is the more dangerous of the two outcomes because it is easy to "calibrate away" and thereby make permanent.

3.4 The bridge output is unchanged, since its excitation is stable. The ADC's reference falls by 2 %, so the counts rise by 2 % and the reported value reads **2 % high**. It is not calibratable, because the sag depends on the load on the rail and therefore varies during operation. The fix is to reference the ADC from the same 5.000 V source, or to make the rail stable.

3.5 $\text{LSB} = 8000/65\,536 = 0.122$ mg; $\sigma_q = 0.122/\sqrt{12} = 0.035$ mg; $\sigma_n = 90 \times \sqrt{10} = 285\ \mu g = 0.285$ mg. Noise is $0.285/0.122 = 2.3$ LSB $\rightarrow 1.2$ bits, so **14.8 effective bits**. Noise still dominates, but only by a factor of eight rather than forty — at 10 Hz of bandwidth the converter is beginning to be relevant, which is exactly the regime where a narrow bandwidth is worth buying.

3.6 Loop period $= 20.350$ ms against an assumed 20.000 ms. Eight hours is 28 800 s of intended time, i.e. $28\,800/0.020 = 1\,440\,000$ samples, whose real duration is $1\,440\,000 \times 0.020350 = 29\,304$ s. The final timestamp is **504 s — over eight minutes — early**, an error of **1.75 %**.

3.7 (a) Yes: a multiplication, once you have a two-point calibration. (b) No: the alias is arithmetically identical to a genuine 40 Hz signal. (c) No: the values were never stored, so the information does not exist. (d) Yes: reinterpret the file, the magnitudes are correct. (e) Partly: averaging N samples reduces it by √N, at the cost of bandwidth — so it is removable only to the extent that you did not need the bandwidth.

3.8 Choose **ODR = 104 Hz** with the low-pass cut-off at or just above 40 Hz. Justification: 104 Hz gives a Nyquist limit of 52 Hz, comfortably above the 40 Hz of interest with margin for the filter's finite roll-off; 52 Hz would put Nyquist at 26 Hz, below the signal, and 208 Hz doubles the noise-bandwidth product for nothing (noise rises as √BW). If the filter cannot be enabled, an external RC filter must be fitted **before** the sensor's own sampling — which for a digital sensor is impossible, so the correct answer is to raise the ODR to 416 Hz so that the un-filtered analog bandwidth is a smaller multiple of Nyquist, digitally low-pass filter, and decimate to 104 Hz in software. Note carefully that this reduces the problem and does not solve it: anything above 208 Hz still folds. Choosing a sensor whose anti-alias filter you can enable is the real answer, and it is a selection criterion, which puts it back in Chapter 2.

4 · Structures, transduction, and what the package does to them

What you should be able to do after this chapter

1. **Identify** which of five canonical structures a given MEMS sensor uses, and name the relation that governs its behaviour.
2. **Predict** from the transduction principle alone whether a device can measure a *static* quantity, and justify the prediction.
3. **Order** the four fabrication steps and state, for each, one device limitation it imposes.
4. **Explain** why a packaged sensor performs differently on your board than in its datasheet, naming three distinct stress paths.
5. **Decide** which specifications must be measured on the final assembly rather than taken from a datasheet.

4.1 The question that had no answer

Chapter 2 put a question on the page and refused to answer it:

> *"What is this device's cross-axis sensitivity after it is soldered to my board?"*

The answer given was that the number is not in the datasheet and cannot be. That was left deliberately unpaid, and this chapter pays it.

The reason is not commercial secrecy, and it is not an omission. It is that **a MEMS sensor is not a component.** It is a mechanical structure two micrometres thick, suspended over an etched cavity, glued into a small plastic box with an adhesive that shrinks as it cures, and soldered to a board that bends when you tighten a screw. The manufacturer can characterise everything up to the edge of the box. Beyond that edge the device is in your hands, and what happens there is a property of *your* assembly, which the manufacturer has never seen.

Everything the datasheet cannot promise you comes from that. So this chapter opens the box — and, per the course's design, opens it only as far as the drift, the stress and the failure modes require. This is not a fabrication course, and a chapter that turned into a process tour would be a chapter that failed.

The anchor: the packaging tax

Here are two rows from the ISM330DHCX datasheet (DS13012 Rev 6), and this time the interesting part is the conditions column.

Parameter	Typ	Max	Conditions
Zero-g level	±10 mg	±65 mg	25 °C, **after soldering**

Recall Chapter 2's anchor: a tilt of 0.5° is 8.73 mg of signal. So:

typical zero-g offset ÷ signal	10 / 8.73	**1.15×**
maximum zero-g offset ÷ signal	65 / 8.73	**7.45×**

Even the typical value exceeds the entire quantity being measured, and the maximum exceeds it more than sevenfold. And the manufacturer is telling you why, in the specification itself: *after soldering*. The assembly process is part of the error, they know it, and they have measured it so that you do not have to guess.

The twist worth eighty minutes

Chapter 2's killer term was also 10 mg — the offset *drift*, ±0.5 mg/°C over ±20 °C. The same number appears twice in Module A for entirely different reasons, and the difference is the point of this chapter:

	Chapter 2's 10 mg	Chapter 4's 10 mg
What it is	offset **drift** over temperature	zero-g **offset** after soldering
Where it comes from	the temperature coefficient of the structure and its package	die attach, package body and solder stress
Against the 8.73 mg signal	1.15×	1.15× typ, 7.45× max
Can you remove it?	**No.** It moves after you calibrate	**Yes.** One-point calibration on your board
What it costs to remove	—	one measurement, once, per unit

So fabrication and packaging hand you a **large but removable** error, while the temperature coefficient hands you a **small-looking but irremovable** one. If you take one thing from this chapter, take that asymmetry: it is why Chapter 2's selection rule weighted drift so heavily, and it is why Lecture 14 and Laboratory 3 exist.

That the two numbers collide at 10 mg is a coincidence of this particular part. It is flagged here because an unremarked coincidence becomes a permanent confusion.

4.2 Five structures do all the work

Eleven of this course's sixteen lectures are about sensor families. Between them, those families use five mechanical structures. Learning the five is a better investment than learning the families, because the families are combinations of them.

Proof mass on a flexure

A block of silicon — the **proof mass**, also called the seismic mass — suspended by thin beams that act as springs. When the frame accelerates, the mass lags behind:

$$x = \frac{ma}{k} \qquad f_{0} = \frac{1}{2\pi}\sqrt{\frac{k}{m}}$$

Displacement is proportional to acceleration, so measuring displacement measures acceleration. The resonant frequency f_0 sets the usable bandwidth: you must work well below it, or the structure's own dynamics distort what you are trying to measure.

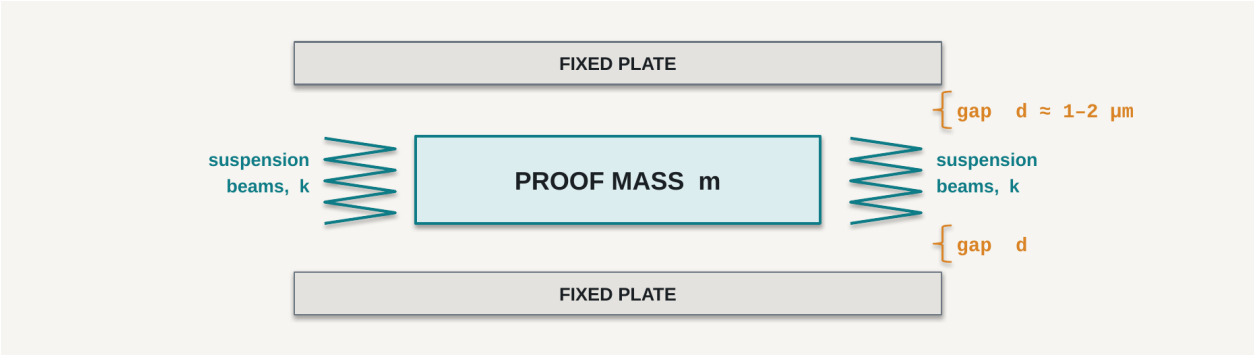

Figure 4.1 — The structure from Chapter 1, revisited. The proof mass of mass *m* hangs on flexures of stiffness *k* between two fixed plates separated from it by gaps *d*. Under acceleration one gap grows and the other shrinks, and the differential capacitance is the output. Typical gaps are 1–2 µm, smaller than a red blood cell — which is why one dust particle inside the package is catastrophic, and why Section 4.7 spends more time on packaging than on lithography.

Senses: acceleration, and — with a second mass driven into vibration — angular rate. *Course examples:* the accelerometer of Lecture 5, the gyroscope of Lecture 6.

Cantilever beam

A beam clamped at one end and free at the other. For a rectangular section of width *w* and thickness *t*, loaded at the tip:

$$k = \frac{Ewt^{3}}{4L^{3}}$$

The cube on the thickness is the whole story of MEMS mechanical design. Doubling the thickness makes the beam eight times stiffer; halving the length makes it eight times stiffer again. Small changes in geometry produce large changes in behaviour, which is both the opportunity and the reason tolerances matter so much.

The cantilever is also the flexure of the structure above, so this is not really a sixth structure — it is the previous one's suspension, considered on its own.

Senses: force, displacement. *Course examples:* force and tactile sensing, Lecture 8.

Diaphragm over a cavity

A thin membrane clamped at its edge, with a sealed or vented cavity beneath. A pressure difference across it deflects it. For a clamped circular plate of radius *a* and thickness *t*, at small deflections:

$$w_{centre} \propto \frac{\Delta P \, a^{4}}{E t^{3}}$$

Note that there is no proof mass here at all. A diaphragm device responds to a pressure difference, not to inertia — which is why the answer to "which of these has no proof mass?" is the pressure sensor, and why microphones and barometers are close relatives.

What is on the other side decides what the device measures: a sealed vacuum cavity gives **absolute** pressure, a vent to atmosphere gives **gauge** pressure, and a second port gives **differential** pressure. Same structure, three products.

Senses: pressure, sound. *Course examples:* pressure sensors in Lecture 8, MEMS microphones in Lecture 11.

Resonator

A structure deliberately driven at its resonant frequency, where the *measurand shifts the frequency* rather than producing a displacement. Added mass lowers f_0; axial stress raises it; temperature changes the modulus and moves it too.

This is a different measurement philosophy from the other four, and a powerful one: frequency can be measured to parts per million with a counter, far more precisely than an amplitude can be measured with an ADC. It is also the principle behind sensors that need extreme stability.

Senses: mass (hence gas concentration), temperature, time, stress. *Course examples:* the gas-sensing microheaters of Lecture 9; timing references.

Comb structure

Interdigitated fingers, like two combs pushed together without touching. With n finger pairs of overlap area A and gap g:

$$C = \frac{n \varepsilon A}{g} \qquad F \propto \frac{n \varepsilon h}{g} V^{2}$$

Two things make this the most important structure in the catalogue. First, n can be in the hundreds, so a comb multiplies a tiny capacitance into a measurable one. Second — and this is the part worth remembering — **the comb both senses and drives.** Apply a voltage and it produces a force; measure its capacitance and it reports a displacement.

That dual nature is why gyroscopes are possible at all. A drive comb sustains a proof mass in vibration; a sense comb, at right angles, detects the Coriolis deflection when the device rotates. Neither function is available from any of the other four structures, and Lecture 6 depends entirely on it.

Senses and actuates: displacement, and force. *Course examples:* accelerometers and gyroscopes, and the scanning mirrors of Lecture 10.

4.3 Six ways to turn motion into a signal

A structure deflects. Something must notice. There are six mechanisms in common use, and the property that most often decides between them is not sensitivity or cost — it is whether the mechanism responds to a **static** input at all.

Capacitive

Measure the capacitance between a moving element and a fixed one, $C = \varepsilon A / d$. Nothing flows and nothing is consumed, so it responds to a constant displacement indefinitely.

Its weakness is that the capacitances are tiny — femtofarads — and the parasitic capacitance of a bond wire is larger. So the sense electronics must be **on the same die**, which is precisely why capacitive MEMS and

integrated circuits grew up together. It is also why the datasheet quotes an output in mg rather than in farads: you never see the capacitance.

Static response: yes. This is the reason the part you selected in Chapter 2 can measure tilt.

Piezoresistive

Strain changes a resistor's resistance:

$$\frac{\Delta R}{R} = G \, \varepsilon$$

where G is the gauge factor. For metal foil $G \approx 2$; for doped single-crystal silicon it is 50 to 200. That factor of fifty is the entire reason silicon strain sensing is worth doing, and it is one of the happy accidents of the material.

Four such resistors in a Wheatstone bridge give a millivolt-level differential output — large enough to carry off-chip, which is why piezoresistive sensors can be simpler devices than capacitive ones.

The weakness is temperature. A resistor's resistance and its gauge factor both depend on temperature, strongly. This is why pressure sensor datasheets are full of compensation coefficients and why they specify a total error band over temperature rather than an accuracy at 25 °C — a point Chapter 2's final poll turned on.

Static response: yes.

Piezoelectric

Certain crystals produce charge in proportion to applied stress: $q = d\,F$. No excitation is needed; the material generates its own signal. The sensitivity can be excellent and the noise very low.

And it cannot measure anything static. Charge is produced by a *change* in stress. Held at constant stress, the charge leaks away through the finite input impedance of whatever is connected, and the output decays to zero with a time constant of seconds to minutes.

So a piezoelectric accelerometer on a frame tilted to 0.5° and held there reads 0.5° briefly and then reads **zero**. Not inaccurately — zero. This is the single most important discriminator in this chapter, because it is the only place in Module A where the transduction principle *forbids* an application regardless of every specification on the front page. No amplifier fixes it: a higher input impedance lengthens the decay, and the frame will be tilted for years.

Static response: no. Excellent for vibration and impact; useless for tilt, and the datasheet says so in the form of a low-frequency limit that people skip past.

Thermal

Convert the measurand into a temperature difference and measure that, or measure how the measurand changes heat flow. Thermal anemometers, thermopile infrared sensors and the microheaters of MEMS gas sensors all work this way.

MEMS suits this well, for a reason Chapter 1 already gave: as a device shrinks, its surface-to-volume ratio rises as $1/L$, so small structures exchange heat quickly. A micro-scale heater reaches temperature in milliseconds where a macroscopic one takes minutes.

The weaknesses are speed at the system level, and self-heating — the device is a heat source inside its own measurement. It also explains the warm-up specification on gas sensors, which is a genuine operational constraint and not a datasheet formality.

Static response: yes, but slowly.

Electromagnetic

Induction, Hall effect, magnetoresistance. Signals can be large and low-impedance, which makes the electronics easy.

It shrinks badly. Induced voltage depends on flux and on the number of turns, and neither survives miniaturisation well, so this is the one principle in the list where MEMS has *less* advantage. It is central to Lecture 7 nonetheless, because magnetic field is itself a measurand worth having, and hard-iron and soft-iron distortion — the calibration problem of that lecture — is a direct consequence of how these devices work.

Static response: yes (Hall and magnetoresistive; induction needs change).

Optical

Interrupt, reflect or interfere with a light beam. Signal levels can be enormous and the noise floor very low; optical MEMS gives the highest resolution of anything in this list.

The cost is a window, a light source and an optical path — all of which must survive assembly, stay clean, and be designed around ambient light. The cover-window design problem of Lecture 10 exists entirely because of this.

Static response: yes.

The comparison that matters

Principle	Static?	Signal	Main weakness	The datasheet line it explains
Capacitive	**yes**	fF — needs on-chip electronics	stray capacitance	why your part can measure tilt
Piezoresistive	**yes**	mV from a bridge	strong temperature coefficient	the compensation coefficients on a pressure sensor
Piezoelectric	**no**	charge, self-generating	no DC response	the low-frequency limit on a vibration sensor
Thermal	yes, slowly	µV or ΔR	slow, self-heating	the warm-up time on a gas sensor
Electromagnetic	yes	large, low-Z	does not miniaturise	hard-iron and soft-iron terms
Optical	yes	can be very large	needs a clean window	cover-window and crosstalk specifications

Every row's rightmost column is a specification the reader has already met. That is the argument of this chapter in one table: **the transduction principle is not background physics, it is the explanation of the datasheet.**

4.4 How it is made, in four steps

Four steps, one consequence each. That is the whole treatment, and the brevity is deliberate — the semester plan restricts fabrication to the depth that explains device behaviour, and the ITMO text listed in Section 4.9 covers the subject properly for anyone who wants it.

Lithography — վիմագրություն

A pattern is projected onto a light-sensitive **photoresist** layer, which is developed away to leave a mask over the regions to be protected.

The consequence. Lithography sets the minimum feature size. Minimum feature size sets the comb gap *g*. The gap sets the capacitance $C = n\varepsilon A/g$. The capacitance sets the signal level, and therefore the noise floor. So the noise density you budgeted with in Chapter 2 traces back, through four steps, to how finely the manufacturer can print a line.

Deposition — նստեցում

Thin films are grown or deposited on the wafer: structural silicon, insulators, metals.

The consequence. Films are deposited hot and used cold, and they and the substrate contract by different amounts. The result is **residual stress**, so the structure is already curved and pre-loaded before it has measured anything. Residual stress is a large part of why zero-g offset exists at all, and why it varies from wafer to wafer.

Etching — փորագրում

Material is removed where the mask does not protect it. **Isotropic** etching removes material equally in all directions, producing rounded profiles; **anisotropic** etching removes it preferentially along one direction, producing the deep vertical walls that comb fingers require.

The consequence. Which etch is available decides which shapes exist. Deep, narrow, vertical-walled trenches are what make a high-*n* comb possible, and they are the reason a modern capacitive accelerometer outperforms one from twenty years ago far more than the electronics do.

Release — ազատում

A **sacrificial layer** beneath the structure is dissolved, leaving it free to move. This is the step that turns a stack of films into a machine.

The consequence. A freed structure separated from the substrate by a micrometre is vulnerable to **stiction** — surface forces that hold it down permanently once it touches. At this scale surface forces exceed the restoring force of the spring, which is Chapter 1's area-to-volume scaling argument arriving with teeth. Stiction is a principal cause of MEMS failure under shock, and it is why datasheets specify a shock survival rating in thousands of g.

Bulk or surface

Two philosophies, and the trade-off is mass.

	Bulk micromachining	Surface micromachining
Method	etch into the wafer itself	build up from thin films on top of it
Structures	thick, large, heavy	thin, small, light
Consequence	**lower noise** — more proof mass	**cheaper, smaller, integrable** with electronics
Typical use	pressure sensors, high-performance inertial	consumer IMUs, including your part

A reader who has Chapter 1's scaling laws can predict that column before being told it: noise falls as proof mass rises, so the bulk-machined device is quieter, and the surface-machined one is cheaper and sits in a phone. Both facts follow from $m \propto L^{3}$.

4.5 The package is part of the sensor

The die is now a working mechanical structure a few micrometres thick, with gaps of one to two micrometres, which must survive being handled, soldered, and used for ten years. Packaging is what makes that possible, and it is where most of Section 4.1's error comes from.

- **The cavity.** The structure needs room to move. So the package contains a sealed void, and its internal pressure sets the damping — and therefore the bandwidth and the noise. Gyroscopes are often sealed near vacuum so their resonator has a high quality factor; accelerometers are damped deliberately.
- **Die attach.** The die is glued down. The adhesive cures, shrinks, and pulls on the die — and it does so by a different amount at every temperature. This is a direct contributor to zero-g offset and to its temperature coefficient.
- **Electrical connection.** Wire bonds or through-silicon vias. Bond wires add stray capacitance, which for a femtofarad-level capacitive sensor is the reason the electronics had to be on-die in the first place.
- **Hermetic sealing and the getter.** The cavity must stay sealed, or damping changes over years. A **getter** — a material that absorbs stray gas molecules — is sometimes included to absorb what **outgasses** from the adhesives inside. A slow leak or slow outgassing appears as multi-year drift, and this is one of the mechanisms behind ageing specifications.
- **Moulding.** The plastic body has its own thermal expansion coefficient, different from silicon's, and it is what physically couples your board's deformation into the die.

None of this protects the sensor from your assembly. It couples the sensor *to* it.

4.6 The stress chain

Six links between the transduction and the world, and only the first is specified by anybody.

```
die → die-attach adhesive → package body → solder joints → PCB → mounting screw
```

Link	What it contributes	Specified by anyone?
die	the transduction itself	**yes** — the datasheet
die attach	offset shift, hysteresis	partly — this is what "after soldering" covers
package body	temperature coefficient	partly
solder joints	**asymmetric** stress → cross-axis error	**no**
PCB flex	offset and cross-axis, varying with load	**no**
mounting screw	an offset that changes when it is retightened	**no**

Three of the six are unspecifiable in principle, because they are properties of a board and an enclosure the manufacturer has never seen. This is the answer to Section 4.1's question, and it is not an evasion: **no measurement the manufacturer could make would predict your assembly.**

And there is a seventh link. Chapter 1's war story ended with an accelerometer bolted to a motor housing through a 5 mm rubber pad, which had already discarded two-thirds of the vibration before the first electron moved. That pad is the outer end of this same chain. The course's first story and its fourth chapter close on one diagram.

The engineering response is not despair, and it is not to demand better datasheets. It is to **measure the last three links yourself, once, on the real assembly** — which is Laboratory 3, and which is why Chapter 2's selection rule preferred parts with a self-test function and an accessible register map.

4.7 Worked calculation: the packaging tax, and what to do about it

Given. The part from Chapter 2, on your board. Zero-g level ±10 mg typ, ±65 mg max, at 25 °C after soldering. Zero-g temperature coefficient ±0.1 mg/°C typ, ±0.5 mg/°C max. The requirement is still 0.5° of tilt, calibrated once at 20 °C, used over 0–40 °C.

Step 1 — the signal.

$$0.5° \rightarrow \sin(0.5°) \times 1\,g = 8.73\ \text{mg}$$

Step 2 — the offset, uncalibrated. At the maximum figure, 65 mg. As an angle:

$$\arcsin(65/1000) = 3.73°$$

Seven and a half times the requirement. A device that has been soldered but not calibrated reports a tilt of nearly four degrees on a level frame.

Step 3 — calibrate once, on your board, at 20 °C. Measure the output with the frame known level and subtract it. The offset term becomes, in principle, zero — limited only by how well you knew "level" and by the repeatability of the measurement, which is a few tenths of a milli-*g*.

Step 4 — what calibration did not fix. The temperature coefficient. Over ±20 °C from the calibration point, at the maximum figure:

$$\pm 0.5\ \text{mg/°C} \times 20\ °\text{C} = 10\ \text{mg} \rightarrow 0.57°$$

The verdict. A 65 mg error — 7.45× the requirement — was removed by one measurement taking a few seconds. A 10 mg error, six and a half times smaller, remains, and no amount of bench work removes it because it is not there when you are on the bench.

This is the asymmetry of Section 4.1, in arithmetic. **The large error was cheap and the small error was fatal**, and the only way to know which was which was to understand where each came from physically. That understanding is what this chapter is for.

For completeness, the route to fixing the residual term is temperature compensation: log the die temperature — the part has a temperature output, and almost nobody uses it — characterise the coefficient over the range, and correct in software. That is Lecture 14, and it is the reason Chapter 3 insisted that you log the temperature column.

4.8 The honest ledger

Chapter 1 closed with a ledger of what shrinking a device buys and costs. This chapter can now complete it, because two of the rows were not available then.

Shrinking the device	buys you	costs you
$m \propto L^{3}$	—	**less mass → a higher noise floor**
$f_{0} \propto 1/L$	**higher bandwidth**	resonance moves toward your signal band
area/volume $\propto 1/L$	fast thermal response	**surface forces dominate → stiction**
batch fabrication	**unit cost; electronics on the same die**	tolerances you cannot control
packaging	protection and handling	**stress you cannot specify**

> Micro-scale is a trade, not an improvement. Everything gained was gained by giving something up — and the datasheet quantifies the gains, while the last two rows are yours to measure.

4.9 Reading

- **Morris & Langari, *Measurement and Instrumentation*, 3rd ed.** — Chapter 13 on pressure measurement and Chapter 20 on displacement, for the diaphragm and capacitive structures treated as instruments rather than as devices. The book is deliberately not a MEMS text, which is why this chapter exists.
- **Fraden, *Handbook of Modern Sensors*, 5th ed.** — the reference for transduction principles. Consult it per principle, not cover to cover; the piezoelectric and piezoresistive chapters are the ones worth reading in full alongside Section 4.3.

- **Senturia, *Microsystem Design*** — the standard graduate text. Its introductory, sensing, noise and packaging sections are readable now; the fabrication-heavy chapters are optional and beyond this course's scope.
- **The ITMO text** (books.ifmo.ru/file/pdf/2673.pdf, Russian, 75 pp) covers MEMS device physics and fabrication properly and freely, and satisfies the accredited programme's bibliography. If Section 4.4 left you wanting the real process detail, this is where to get it.
- **ISM330DHCX datasheet DS13012 Rev 6** — the zero-g level rows and, this time, their conditions column. Read "25 °C, after soldering" and "package level, not board level" as the technical statements they are.

4.10 Exercises

4.1 For each device, name the structure it uses and the transduction principle: (a) a barometric pressure sensor in a phone; (b) a MEMS microphone; (c) a vibrating-structure gyroscope; (d) a MEMS gas sensor with a microheater; (e) a scanning mirror in a laser projector.

4.2 A cantilever is 200 µm long, 20 µm wide and 2 µm thick. By what factor does its stiffness change if the thickness is increased to 3 µm? If instead the length is reduced to 150 µm?

4.3 A piezoelectric accelerometer with a low-frequency limit of 0.5 Hz is proposed for (a) detecting bearing faults at 1520 Hz, (b) measuring the tilt of a solar tracker, (c) detecting whether a machine has been switched on. Which are viable, and why?

4.4 A capacitive accelerometer has a gap of 1.5 µm. A dust particle 3 µm across enters the cavity during assembly. What happens, and why is this a packaging problem rather than a fabrication one?

4.5 Two accelerometers have identical specifications except that one is bulk-micromachined and one is surface-micromachined. Which would you expect to have the lower noise density, and which the lower unit price? Justify both from a scaling law.

4.6 A pressure sensor's datasheet specifies ±0.05 % FS accuracy at 25 °C, and a total error band of ±0.25 % FS over 0–50 °C. Your enclosure clamps the sensor's body with an O-ring at an unknown torque. Which of the two figures is useful to you, and what third number do you now need? How would you obtain it?

4.7 Explain, in three sentences, why the cross-axis sensitivity of a soldered device cannot be specified by its manufacturer, and state what an engineer should do instead.

4.8 Your capstone project mounts an IMU on a bracket held by two M3 screws. List three specifications that will differ from the datasheet once assembled, name the stress path responsible for each, and state which of the three you can remove by calibration.

Answers

4.1 (a) Diaphragm over a sealed cavity, piezoresistive or capacitive. (b) Diaphragm, capacitive or piezoelectric. (c) Proof mass on flexures with drive and sense combs, capacitive — the combs both drive the vibration and sense the Coriolis deflection. (d) Resonator or a heated membrane, thermal. (e) Comb structure, capacitive — used as an actuator here rather than a sensor, which the comb uniquely permits.

4.2 Stiffness $\propto t^{3}/L^{3}$. Thickness 2 → 3 µm: $(3/2)^{3} = 3.4$ times stiffer. Length 200 → 150 µm: $(200/150)^{3} = 2.4$ times stiffer. Both are large changes from small geometric ones, which is why lithographic tolerance matters mechanically and not just electrically.

4.3 (a) Viable — 1520 Hz is far above the 0.5 Hz limit, and vibration is exactly what piezoelectric devices are best at. (b) Not viable at all: tilt is a static measurand and the device has no DC response, so it will read zero regardless of specifications. (c) Viable, and in fact a good fit — switch-on is a *change*, which is precisely what the principle detects; note that it cannot then tell you the machine is *still* on.

4.4 The particle is twice the gap, so it either jams the proof mass — the device reads a constant value and never moves — or it holds the structure to one plate permanently. It is a packaging problem because the structure is released and mechanically free before the package is sealed, so cleanliness during assembly and sealing, not during lithography, determines whether a particle can reach the cavity. This is the concrete reason MEMS packaging is done in clean conditions and why the seal integrity in Section 4.5 matters.

4.5 The bulk-micromachined device should have the **lower noise density**, because bulk etching produces a thicker, heavier proof mass and $m \propto L^{3}$; more mass means a larger displacement per unit acceleration and therefore a better signal against the same electronic noise. The surface-micromachined device should have the **lower unit price**, because it is built from thin films on the wafer surface, occupies less area, and can integrate its electronics on the same die.

4.6 The **total error band over 0–50 °C** is the useful figure — it bounds the error over the range you will actually operate in, which is Chapter 2's final lesson. Neither figure covers the O-ring: you now need the **zero-offset shift caused by your own clamping**, which appears in no datasheet. Obtain it by measuring the sensor's output at a known reference pressure both unclamped and clamped at the production torque, on several units, and treating the difference as a systematic error to be calibrated out per unit — plus its spread as an additional uncertainty term.

4.7 Cross-axis sensitivity after soldering depends on the asymmetry of the solder joints, the stiffness and flex of the board, and how the board is mounted — none of which the manufacturer has seen or can control. Any figure they published would describe their evaluation board, not yours, and would therefore be misleading rather than merely incomplete. The engineer's response is to measure it once on the real assembly, using a two-axis reference (a level surface and a known tilt), and to record it as a project-specific calibration constant.

4.8 Three defensible answers, among others. **Zero-g offset** — via die attach, package body and solder-joint stress; **removable** by one-point calibration on the assembled unit. **Cross-axis sensitivity** — via asymmetric solder-joint stress and PCB flex; removable only by measuring the full 3×3 misalignment matrix, which is Laboratory 3, so partly. **Zero-g offset temperature coefficient** — via the mismatch in thermal expansion between die, adhesive, package and board; **not removable** by calibration at a single temperature, and requiring temperature logging and compensation instead. A fourth worth mentioning: the offset will *change* if the M3

screws are retightened, which means the calibration must be performed after final assembly and repeated if the bracket is ever disturbed.