

Lecture Book

B.Sc Mobile Robotics

D. T. McGuiness, PhD

Version: 2025.SS

(2025, D. T. McGuines, Ph.D)

Current version is 2025.SS.

This document includes the contents of Drive Systems, official name being *Mobile Robotics*, taught at MCI in the Mechatronik Design Innovation. This document is the part of the module MECH-B-4-MRV-MRO-ILV taught in the B.Sc degree.

All relevant code of the document is done using *SageMath* where stated using v10.3 and Python v3.13.1.

This document was compiled with *LuaT_EX* v1.18.0, and all editing were done using GNU Emacs v29.4 using AUCT_EX and org-mode package.

This document is based on the books and resources:

The current maintainer of this work along with the primary lecturer
is D. T. McGuines, Ph.D. (dtm@mci4me.at).

Contents

I Mechanics of Mobile Robotics	3
1 Locomotion	5
1.1 Introduction	5
1.1.1 Key Issues for Locomotion	8
1.2 Legged Mobile Robots	10
1.2.1 Examples of Legged Robot Locomotion	13
1.3 Wheeled Mobile Robots	17
1.3.1 Design	17
1.3.2 Stability	19
1.3.3 Manoeuvrability	20
1.3.4 Controllability	21
1.3.5 Case Studies for Wheeled Motion	22
1.3.6 Walking Wheels	24
2 Perception	27
2.1 Introduction	27
2.1.1 Sensors for Mobile Robotics	27
2.1.2 Sensor Classification	28
2.1.3 Characterising Sensor Performance	29
2.1.4 Wheel and Motor Sensors	34
2.2 Active Ranging	40
2.2.1 The Ultrasonic Sensor	41
2.2.2 Motion and Speed Sensors	48
2.3 Vision Based Sensors	51
2.3.1 CMOS Technology	55
2.3.2 Visual Ranging Sensors	56
2.3.3 Depth from Focus	57
2.4 Feature Extraction	61
2.4.1 Defining Feature	61
2.4.2 Using Range Data	63
II Probability and Statistics	65
3 Theory of Probability	67
3.1 Introduction	67

3.2	Experiments & Outcomes	71
3.3	Probability	72
3.4	Permutations & Combinations	77
3.4.1	Permutations	77
3.4.2	Combinations	78
3.4.3	Factorial Function	79
3.4.4	Binomial Coefficients	80
3.5	Random Variables and Probability Distributions	81
3.5.1	Discrete Random Variables and Distributions	82
3.5.2	Continuous Random Variables and Distributions	83
3.6	Mean and Variance of a Distribution	85
3.7	Binomial, Poisson, and Hyper-geometric Distributions	88
3.7.1	Sampling with Replacement	91
3.7.2	Sampling without Replacement: Hyper-geometric Distribution	91
3.8	Normal Distribution	93
3.8.1	Distribution Function	94
3.8.2	Numeric Values	94
3.8.3	Normal Approximation of the Binomial Distribution	95
3.9	Distribution of Several Random Variables	96
3.9.1	Discrete Two-Dimensional Distribution	96
3.9.2	Continuous Two-Dimensional Distribution	97
3.9.3	Marginal Distributions of a Discrete Distribution	97
3.9.4	Marginal Distributions of a Continuous Distribution	98
3.9.5	Independence of Random Variables	99
3.9.6	Functions of Random Variables	100
3.9.7	Addition of Means	100
3.9.8	Addition of Variances	101
4	Statistical Methods	103
4.1	Introduction	103
4.2	Point Estimation of Parameters	107
4.2.1	Maximum Likelihood Method	108
4.3	Confidence Intervals	111
4.4	Testing of Hypotheses and Making Decisions	116
4.4.1	Errors in Tests	118
4.5	Goodness of Fit	121
4.6	Regression and Correlation	124
4.6.1	Regression Analysis	124
4.6.2	Confidence Intervals	127
4.6.3	Correlation Analysis	129
4.6.4	Test for the Correlation Coefficient	131
4.7	Bayesian Statistics	133
4.7.1	Subjective Probability	133

III Localisation and Mapping	135
5 Mobile Robot Localisation	137
5.1 Introduction	137
5.2 The problems of Noise and Aliasing	139
5.2.1 Sensor Noise	139
5.2.2 Sensor Aliasing	140
5.2.3 Effector Noise	141
5.2.4 An Error Model for Odometric Position Estimation	143
5.3 Localisation v. Hard-Coded Navigation	145
5.4 Representing Belief	148
5.4.1 Single Hypothesis Belief	148
5.4.2 Multiple Hypothesis Belief	150
5.5 Representing Maps	152
5.5.1 Continuous Representation	153
5.5.2 Decomposition Methods	155
5.5.3 Current Challenges	158
5.6 Probabilistic Map-Based Localisation	160
5.6.1 Introduction	160
IV GNU/Linux Operating System	163
6 Welcome to Linux	165
6.1 Learning the Linux Command Line	165
6.1.1 A History of Command Line Interface	166
6.1.2 Linux is a Nutshell	168
6.1.3 Linux Distributions	169
6.2 Installation	172
6.2.1 Docker	172
V Robot Operating System	175
7 Installation	177
7.1 Introduction	177
7.2 Installing ROS Humble Hawksbill	178
7.2.1 Set locale	178
7.2.2 Setup Sources	178
7.2.3 Install ROS 2 packages	179
7.2.4 Setting up the Environment	179
Bibliography	185

List of Figures

1.1	Types of locomotion mechanisms used in biological systems [1].	6
1.2	Bipedal motion is not unique to only humans as a wide variety of animals show bipedal motion [4].	7
1.3	Specific power versus attainable speed of various locomotion mechanisms (Adapted from [1]).	7
1.4	RoboTrac, a hybrid wheel-leg vehicle for rough terrain.	8
1.6	Types of motions used by different animals.	10
1.5	Dug-beetle are a great example for legged mobile robotics [7] as not only they can manoeuvre in their environment using legged motion, they can also manipulate their environment and generate rotational motion [8].	10
1.7	Main locomotory gaits in <i>Pleurotya</i> caterpillar [15].	11
1.8	The Degrees of Freedom (DoF) a human leg has [16].	11
1.9	An example of a leg possessing three (3) DoF [17].	11
1.10	The Raibert hopper [22].	13
1.11	The 2D single Bow Leg Hopper.	13
1.12	The New ASIMO introduced in 2005 [23].	13
1.13	Spring Flamingo is a planar bipedal walking robot [28].	15
1.14	Genghis, one of the most famous walking robots from MIT uses hobby servomotors as its actuators.	16
1.15	Genghis, one of the most famous walking robots from MIT uses hobby servomotors as its actuators.	17
1.16	The four basic wheel types a)Standard wheel: Two degrees of freedom; rotation around the (motorized) wheel axle and the contact point b)castor wheel: Two degrees of freedom; rotation around an offset steering joint c)Swedish wheel: Three degrees of freedom; rotation around the (motorized) wheel axle, around the rollers and around the contact point	18
1.17	NAVLAB I, the first autonomous highway vehicle that steers and controls the throttle using vision and radar sensors [30].	20
1.18	Example of an Ackerman drive used mostly in automotive industry [31].	21
2.1	An example of a rotary encoder. [32]	34
2.2	An example of an electronic compass [33].	35
2.3	Optical Gyroscopes have no moving parts, (unlike mechanical gyroscopes) making them extremely reliable [34].	37
2.4	37
2.5	Signals of an ultrasonic sensor.	41

2.6	An example of an ultrasonic sensor used in Raspberry Pi applications [35].	42
2.8	Schematic of laser rangefinding by phase-shift measurement.	44
2.7	A laser range finder used in robotics applications	44
2.9	Range estimation by measuring the phase shift between transmitted and received signals.	45
2.10	Principle of 1D laser triangulation.	46
2.11	Structured light sources on display at the 2014 Machine Vision Show in Boston [36].	47
2.12	a) Principle of active two dimensional triangulation b) Other possible light structures c) One-dimensional schematic of the principle	47
2.13	Doppler effect between two moving objects (a) or a moving and a stationary object(b)	49
2.14	Sony ICX493AQA 10.14-megapixel APS-C (23.4 × 15.6 mm) Charge Coupled Device (CCD) from digital camera Sony DSLR-A200 or DSLR-A300, sensor side [37].	51
2.15	Normalized Spectral Response of a Typical Monochrome CCD.	52
2.16	Types of colour filter used in commercial and industrial applications	52
2.17	Example of white balance. Here the same scene is emulated to be shot under different light conditions [40].	54
2.18	A close-up view of a Complimentary MOS (CMOS) sensor and its circuitry [44]. .	55
2.19	Photon noise simulation. Number of photons per pixel increases from left to right and from upper row to bottom row [47].	56
2.20	Depiction of the camera optics and its impact on the image. To get a sharp image, the image plane must coincide with the focal plane. Otherwise the image of the point (x, y, z) will be blurred in the image as can be seen in the drawing above. .	57
2.21	Three images of the same scene taken with a camera at three different focusing positions. Note the significant change in texture sharpness between the near surface and far surface [48].	58
3.1	The histogram of the data given in Exercise 1	69
3.2	A visual comparison of the Stirling formula and the actual values of the factorial function.	79
3.3	A visual representation of the Eq. (3.42).	84
3.4	Binomial distribution with different values of probability with a sample size of 50. .	89
3.5	The Poisson distribution with different mean (μ) values.	90
3.6	The probability density distribution of hyper-geometric distribution with different parameters.	92
3.7	The poster child of probability and statistics, the normal distribution.	93
3.8	A visual representation between the relationship of PDF and CDF.	94
3.9	Many samples from a bivariate normal distribution. The marginal distributions are shown on the z-axis. The marginal distribution of X is also approximated by creating a histogram of the X coordinates without consideration of the Y coordinates. . .	97
4.1	Original map by John Snow showing the clusters of cholera cases in the London epidemic of 1854, drawn and lithographed by Charles Cheffins.	105
4.2	The student-t distribution with different degrees of freedom m	114
4.3	Chi-square distribution with different degrees of freedom.	115

4.4	The t -distribution used in example. As can be seen, anything left of the critical line would tell us to reject the hypothesis, whereas if the t value lies on the Right Hand Side (RHS), then the test would tell us the null hypothesis is true.	117
4.5	Illustration of Type I and II errors in testing a hypothesis $\theta = \theta_0$ against an alternative $\theta = \theta_0$	120
4.6	Samples with various values of the correlation coefficient r	130
5.1	Navigation is one if not the most demanding and complicated task in Autonomous Mobile Robotics (AMR). However a successful implementation will result in a versatile AMR which can find its way in unknown environments such as exploring other planets [59].	137
5.2	General schematic for mobile robot localisation.	138
5.3	A sample environment.	145
5.4	An Architecture for Behavior-based Navigation	146
5.5	An Architecture for Map-based (or model-based) Navigation	146
5.6	The three (3) examples of single hypotheses of position using different map representation. a) real map with walls, doors and furniture b) line-based map -> around 100 lines with two parameters c) occupancy grid based map -> around 3000 grid cells sizing 50x50 cm d) topological map using line features (Z/S-lines) and doors -> around 50 features and 18 nodes	149
5.7	The presented robot-centric mapping framework enables mobile robots to create consistent elevation maps of the terrain. Mapping does not necessarily need to be done only in 2D as robots which will be used in outdoor environment would need the height of the map as well [75].	152
5.8	A continuous representation using polygons as environmental obstacles.	153
5.9	Example of a continuous-valued line representation of EPFL. left: real map right: representation with a set of infinite lines.	154
6.1	Hughes telegraph, an early (1855) teleprinter built by Siemens and Halske. The centrifugal governor to achieve synchronicity with the other end can be seen (Original)	166
6.2	Nokia Bell Labs Murray Hill, NJ (Original)	166
6.3	Bourne shell interaction on Version 7 Unix (Original).	167
6.4	The kernel mapping of the Linux operating system.	168
6.5	A Family tree of the debian branch of linux [79].	171
6.6	The docker logo	172

List of Tables

4.1	Useful c values based on a given confidence (γ) value.	112
4.2	Type I and Type II errors in testing a hypothesis $\theta = \theta_0$ against an alternative $\theta = \theta_1$.	119
4.3	Dataset	127
4.4	the given dataset of measurement.	129
6.1	Most popular distributions used according to distrowatch.com	170

List of Examples

3.1 Recording Data	68
3.2 Leaf Plots	68
3.3 Histogram	68
3.4 Empirical Rule Outliers and z-Score	70
3.5 Sample Spaces of Random Experiments & Events	71
3.6 Fair Die	72
3.7 Coin Tossing	73
3.8 Mutually Exclusive Events	74
3.9 Union of Arbitrary Events	74
3.10 Multiplication Rule	75
3.11 Sampling w/o Replacement	76
3.12 An Encrypted Message	78
3.13 Sampling Light-bulbs	79
3.14 Waiting Time Problem	83
3.15 Continuous Distribution	84
3.16 Mean and Variance	85
3.17 Binomial Distribution	89
3.18 Poisson Distribution	90
3.19 The Parking Problem	91
3.20 Marginal Distributions of a Discrete Two-Dimensional Random Variable	98
3.21 Independence and Dependence	99
4.1 Maximum Likelihood of Gaussian Distribution	109
4.2 Maximum Likelihood of Poisson Distribution	109
4.3 For Science	110
4.4 Sampling the Population	110
4.5 Confidence Interval for Mean with known Variance in Normal Distribution	112
4.6 Sample Size Needed for a Confidence Interval of Prescribed Length	113
4.7 Confidence Interval for Mean of Normal Distribution with Unknown Variance	114
4.8 Test for the Mean of the Normal Distribution with Known Variance	120
4.9 Printed Circuit Boards	122
4.10 Regression Line	127
4.11 Confidence Interval for the Regression Coefficient	128
4.12 Uncorrelated but Dependent Random Variables	131

4.13 Test for the Correlation Coefficient	131
---	-----

List of Theorems

3.1	First Definition of Probability	72
3.2	General Definition of Probability	73
3.3	Complementation Rule	73
3.4	Addition Rule for Mutually Exclusive Events	74
3.5	Addition Rule for Arbitrary Events	74
3.6	Multiplication Rule	75
3.7	Permutations	77
3.8	Permutations	78
3.9	Combinations	79
3.10	Random Variable	81
3.11	Mean of a Symmetric Distribution	86
3.12	Transformation of Mean and Variance	86
3.13	Relationship between PDF and CDF	94
3.14	Normal Probabilities for Intervals	94
3.15	Limit Theorem of De Moivre and Laplace	95
3.16	Addition of Means	101
3.17	Multiplication of Means	101
3.18	Addition of Variances	102
4.1	Sum of Independent Normal Random Variables	112
4.2	Student's t-Distribution	114
4.3	Chi-Square Distribution	115
4.4	Chi-square Test for $F(x)$ being the Distribution Function of a Population	122
4.5	Least Square Principle	125
4.6	Assumption A1	125
4.7	Assumption A2	127
4.8	Assumption A3	128
4.9	Sample Correlation Coefficient	129
4.10	Correlation Coefficient	130
4.11	Independence and Relation to Normal Distribution	130

Part I

Mechanics of Mobile Robotics

Chapter 1

Locomotion

Table of Contents

1.1	Introduction	5
1.2	Legged Mobile Robots	10
1.3	Wheeled Mobile Robots	17

1.1 Introduction

A mobile robot needs locomotion mechanisms which enable it to move **unbounded** throughout its environment. However, as with everything in engineering our solution comes with options, and so the selection of a robot's approach to locomotion is an important aspect of mobile robot design. In laboratory settings, there are robots that can walk, jump, run, slide, swim, fly and of course roll.

Most locomotion mechanisms have been inspired by biological counterparts, shown in **Fig. 1.1.**

There is, however, one (1) exception where there is, practically, **NO** natural equivalent:

Actively powered wheel is a human invention achieving high efficiency on flat ground.

This mechanism is **NOT** completely foreign to biological systems¹. Our bi-pedal walking system can be approximated by a rolling polygon, with sides equal in length to the span of the step. As the step size decreases, the polygon approaches a circle or wheel. But nature did not develop a fully rotating, actively powered joint, which is the technology necessary for wheeled locomotion.

Biological systems succeed in moving through a wide variety of harsh environments. Therefore it can be desirable to copy their selection of locomotion mechanisms². Replicating nature in this regard, however, is extremely difficult for several reasons.

¹While this statement is practically true, single cell organism use a similar locomotion to what we call wheel [2].

²Scientifically, this is called **Biomimetics** [3]

Type of motion	Resistance to motion	Basic kinematics of motion
Flow in a Channel	Hydrodynamic forces	Eddies
Crawl	Friction forces	Longitudinal vibration
Sliding	Friction forces	Transverse vibration
Running	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum
Jumping	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum
Walking	Gravitational forces	Rolling of a polygon (see figure 2.2)

Figure 1.1: Types of locomotion mechanisms used in biological systems [1].

- Mechanical complexity is easily achieved in biological systems through structural replication.

Cell division, in combination with specialisation, can readily produce a millipede with several hundred legs and several tens of thousands of individually sensed cilia³. In man-made structures, each part must be fabricated individually, and therefore, no such economies of scale exist.

- Cell is a microscopic building block that enables extreme miniaturisation. With very small size and weight, insects achieve a level of robustness that we have not been able to match with human fabrication techniques.
- The biological energy storage system and the muscular and hydraulic activation systems used in animals and insects achieve torque, response time and conversion efficiencies that far exceed similarly scaled man-made systems.

Based on these aforementioned limitations, mobile robots generally generate motion, either using wheeled mechanisms, a well-known human technology for vehicles, or using a small number of articulated legs, the simplest of the biological approaches to locomotion (shown in Fig. 1.2).

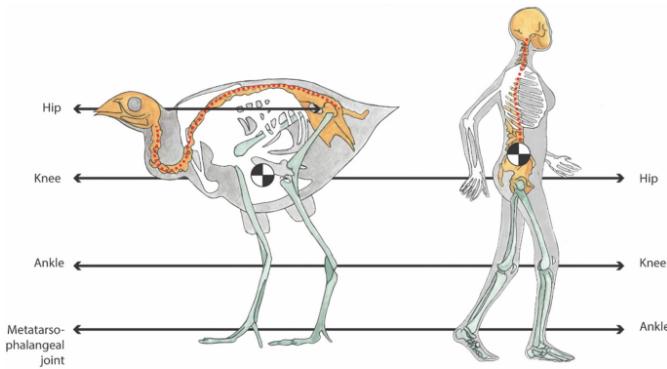


Figure 1.2: Bipedal motion is not unique to only humans as a wide variety of animals show bipedal motion [4].

In general, legged locomotion requires higher DoF and therefore greater mechanical complexity than wheeled locomotion [5]. Wheels, in addition to being simple, are extremely well suited to flat ground. As **Fig. 1.3** depicts, on flat surfaces wheeled locomotion is one to two orders of magnitude more efficient than legged locomotion.

The railway is ideally engineered for wheeled locomotion because rolling friction is minimised using a hard and flat steel surface.

But as the surface becomes soft, wheeled locomotion accumulates inefficiencies due to **rolling friction** while legged locomotion suffers much less because it consists only of **point contacts** with the ground. This is demonstrated in figure 2.3 by the dramatic loss of efficiency in the case of a tire on soft ground.

the efficiency of wheeled locomotion depends greatly on environmental qualities, particularly the flatness and hardness of the ground, while the efficiency of legged locomotion depends on the leg mass and body mass, both of which the robot must support at various points in a legged gait.

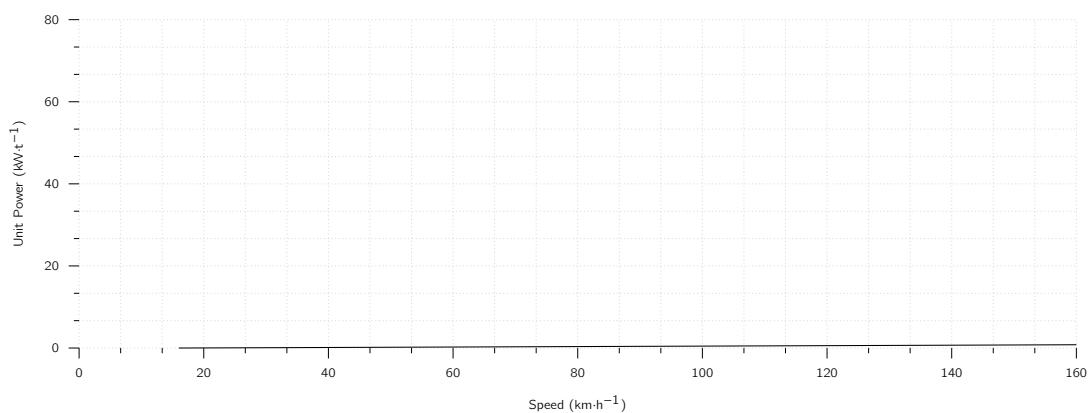


Figure 1.3: Specific power versus attainable speed of various locomotion mechanisms (Adapted from [1]).

It is understandable therefore nature favours legged locomotion, as locomotion systems in nature must operate on rough and unstructured terrain. For example, in the case of insects in a forest the vertical variation in ground height is often an order of magnitude greater than the total height of the insect.

By the same token, the human environment frequently consists of engineered, smooth surfaces both indoors and outdoors. Therefore, it is also understandable that virtually all industrial applications of mobile robotics utilise some form of wheeled locomotion. Recently, for more natural outdoor environments, there has been some progress toward hybrid and legged industrial robots such as the forestry robot [6] shown in **Fig. 1.4**.

In the next section, we present general considerations that concern all forms of mobile robot locomotion. Following this will be overviews of legged locomotion and wheeled locomotion techniques for mobile robots.

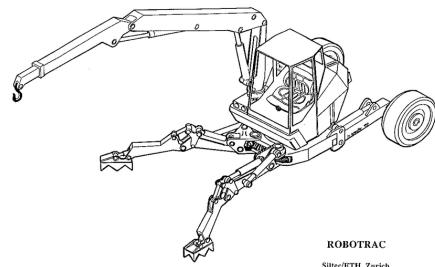


Figure 1.4: RoboTrac, a hybrid wheel-leg vehicle for rough terrain.

1.1.1 Key Issues for Locomotion

Locomotion is the **complement of manipulation**:

- In manipulation, the robot arm is fixed but moves objects in the workspace by imparting force to them.
- In locomotion, the environment is fixed and the robot moves by imparting force to the environment.

For both cases, the scientific basis is the **study of actuators** which generate interaction forces, and mechanisms that implement desired kinematic and dynamic properties. Locomotion and manipulation therefore share the same core issues of stability, contact characteristics and environmental type:

- | | |
|---|---|
| <ul style="list-style-type: none"> ■ Stability <ul style="list-style-type: none"> – number and geometry of contact points – centre of gravity – static/dynamic stability – inclination of terrain
 ■ characteristics of contact | <ul style="list-style-type: none"> – contact point/path size and shape – angle of contact – friction
 ■ type of environment <ul style="list-style-type: none"> – structure – medium (e.g., water, air, soft or hard ground) |
|---|---|

A theoretical analysis of locomotion begins with mechanics and physics. From this starting point, we can formally define and analyse all manner of mobile robot locomotion systems. However, this Lecture Book puts more emphasis on the mobile robot navigation problem, particularly on the topics of perception, localisation and cognition. Therefore, we will not delve deeply into the physical basis of locomotion. Nevertheless, two remaining sections in this chapter present overviews of issues in legged locomotion and wheeled locomotion.



(a) Bipedal motion [9].



(b) Quadpedal motion [10].



(c) Hexapedal motion [11]

Figure 1.6: Types of motions used by different animals.

1.2 Legged Mobile Robots

Legged locomotion is characterised by a **series of point contacts between the robot and the ground**. The primary advantages include adaptability and manoeuvrability in rough terrain. Because only a set of point contacts is required, the quality of the ground between those points does not matter, so long as the robot can maintain adequate ground clearance. In addition, a walking robot is capable of crossing a hole or chasm so long as its reach exceeds the width of the hole. A final advantage of legged locomotion is the potential to manipulate objects in the environment with great skill.

The dung beetle, is capable of rolling a ball while locomotion as a result of its dexterous front legs shown in **Fig. 1.5**.

The main disadvantages of legged locomotion include **power and mechanical complexity**. The leg, which may include several DoF, must be capable of sustaining part of the robot's total weight, and in many robots must be capable of lifting and lowering the robot. Additionally, high manoeuvrability will only be achieved if the legs have a sufficient number of DoF to impart forces in a number of different directions.



Figure 1.5: Dung-beetle are a great example for legged mobile robotics [7] as not only they can manoeuvre in their environment using legged motion, they can also manipulate their environment and generate rotational motion [8].

Leg Configurations and Stability

Because legged robots are biologically inspired, it is instructive to examine biologically successful legged systems. A number of different leg configurations have been successful in a variety of organisms seen in **Fig. 1.6**.

Large animals such as mammals and reptiles have four (4) legs whereas insects have six (6) or more legs. In some mammals, the ability to walk on only two (2) legs has been perfected. Especially in the case of humans, balance has progressed to the point that we can even jump with one leg⁴. This

⁴In child development, one of the tests used to determine if the child is acquiring advanced locomotion skills is the ability to jump on one leg [12].

exceptional manoeuvrability comes at a price:

Bipedal motion is much more complex active control to maintain balance.

In contrast, a creature with three (3) legs can exhibit a static, stable pose provided that it can ensure that its centre of gravity is within the tripod of ground contact. Static stability, demonstrated by a three-legged stool, means that balance is maintained with no need for motion. A small deviation from stability⁵ is passively corrected towards the stable pose when the upsetting force stops. But a robot must be able to lift its legs in order to walk. To achieve static walking, a robot **must** have at least six (6) legs [13]. In such a configuration, it is possible to design a gait⁶ in which a statically stable tripod of legs is in contact with the ground at all times.

Insects⁷ are immediately able to walk when born. For them, the problem of balance during walking is relatively simple. Mammals, with four (4) legs, cannot achieve static walking, but are able to stand easily on four (4) legs. Fauns⁸, for example, spend several minutes attempting to stand before they are able to do so, then spend several more minutes learning to walk without falling [14]. Humans, with two (2) legs, cannot even stand in one place with static stability. Infants require months to stand and walk, and even longer to learn to jump, run and stand on one leg.

There is also the potential for great variety in the complexity of each individual leg. Once again, the biological world provides ample examples at both extremes. For instance, in the case of the caterpillar, each leg is extended using hydraulic pressure by constricting the body cavity and forcing an increase in pressure, and each leg is retracted longitudinally by relaxing the hydraulic pressure, then activating a single tensile muscle that pulls the leg in towards the body, seen in **Fig. 1.7**. Each leg has only a single DoF, which is oriented longitudinally along the leg.

Forward locomotion depends on the hydraulic pressure in the body, which extends the distance between pairs of legs. The caterpillar leg is therefore mechanically very simple, using a minimal number of extrinsic muscles to achieve complex overall locomotion.

At the other extreme, the human leg has more than six (6) major degrees of freedom, combined with further actuation at the toes, shown in **Fig. 1.8**. There are more than 50 muscles in each lower limb and at least half of them participate actively in the control of leg

⁵e.g., such as gently pushing the stool

⁶the pattern of steps of an animal at a particular speed.

⁷such as spiders, ant, beetles, ...

⁸a baby deer

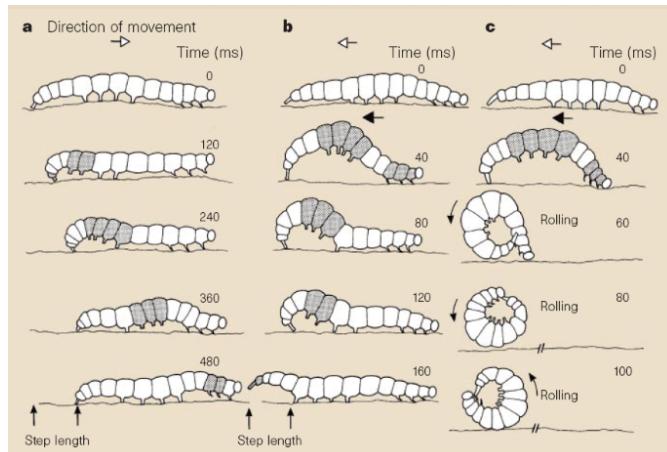
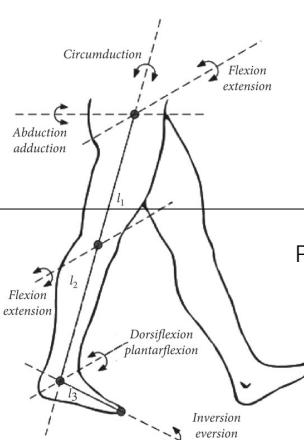


Figure 1.7: Main locomotory gaits in *Pleurotya* caterpillar [15].



motion [18, 19].

In the case of legged mobile robots, a minimum of two (2) DoF is generally required to move a leg forward by lifting the left and swinging it forward. More common is the addition of a 3rd DoF for more complex manoeuvres, resulting in legs such as those shown in **Fig. 1.9**. Recent successes in the creation of bipedal walking robots have added a fourth DOF at the ankle joint [20]. The ankle enables more consistent ground contact by actuating the pose of the sole of the foot. In general, adding DoF to a robot leg increases the manoeuvrability of the robot, both augmenting the range of terrains on which it can travel and the ability of the robot to travel with a variety of gaits. The primary disadvantages of additional joints and actuators is, of course, energy, control and mass. Additional actuators require energy and control, and they also add to leg mass, further increasing power and load requirements on existing actuators.

In the case of a multi-legged mobile robot, there is the issue of leg coordination for locomotion, or gait control.

The number of possible gaits depends on the number of legs [21].

The gait is a sequence of lift and release events for the individual legs. For a mobile robot with k legs, the total number of possible events N for a walking machine is:

$$N = (2k - 1)! \quad (1.1)$$

For a bipedal walker ($k=2$) legs the number of possible events N is:

$$N = (2k - 1)! = 3! = 3 \cdot 2 \cdot 1 = 6 \quad (1.2)$$

The six (6) different events are:

- lift right leg
- lift left leg
- release right leg
- release left leg
- lift both legs together
- release both legs together

As can we see, this list of possible events quickly grows quite large. For example, a robot with six

(6) legs has far more gaits theoretically:

$$N = 11! = 39\,916\,800 \quad (1.3)$$

1.2.1 Examples of Legged Robot Locomotion

Although there are no high-volume industrial applications to date, legged locomotion is an important area of long-term research. Several interesting designs are presented below, beginning with the one-legged robot and finishing with six-legged robots.

Single Leg

The minimum number of legs a legged robot can have is, of course, one. Minimising the number of legs is beneficial for several reasons.

- Body mass is particularly important to walking machines, and the single leg minimises cumulative leg mass.
- Leg coordination is required when a robot has several legs, but with one leg no such coordination is needed.
- The one-legged robot maximises the basic advantage of legged locomotion: legs have single points of contact with the ground in lieu of an entire track as with wheels.

A single legged robot requires only a sequence of single contacts, making it useful in rough terrain.

Perhaps most importantly, a hopping robot can dynamically cross a gap that is larger than its stride by taking a running start, whereas a multi-legged walking robot that cannot run is limited to crossing gaps that are as large as its reach.

The major challenge of creating a single-leg robot is **balance**. For a robot with one leg, static walking is not only impossible, but static stability when stationary is also impossible. The robot must actively balance itself by either changing its centre of gravity or by imparting corrective forces. Thus, the successful single-leg robot **must be dynamically stable**.

Fig. 1.10 shows the Raibert Hopper [24, 25], one of the most well-known single-leg hopping robots created. This robot makes continuous corrections to body attitude and to robot velocity by adjusting the leg angle with respect

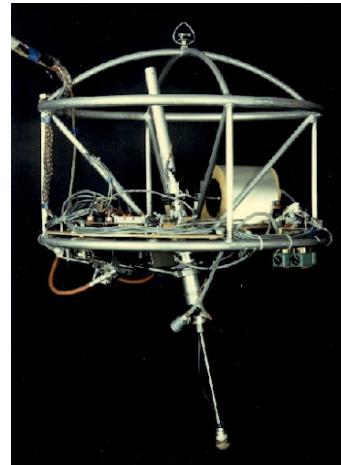


Figure 1.10: The Raibert hopper [22].

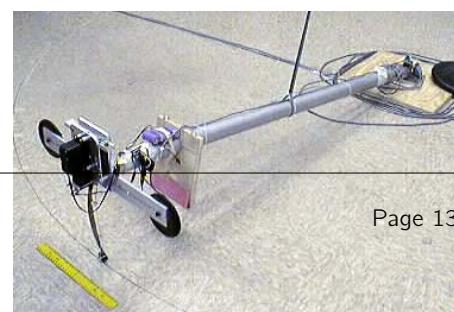


Figure 1.11: The 2D single Bow Leg Hopper.

to the body. The actuation is hydraulic, including high-power longitudinal extension of the leg during stance to hop back into the air. Although powerful, these actuators require a large, off-board hydraulic pump to be connected to the robot at all times. **Fig.** 1.11 shows a more energy efficient design developed [26]. Instead of supplying power by means of an off-board hydraulic pump, the Bow Leg Hopper is designed to capture the kinetic energy of the robot as it lands using an efficient bow spring leg. This spring returns approximately 85% of the energy, meaning that stable hopping requires only the addition of 15% of the required energy on each hop. This robot, which is constrained along one axis by a boom, has demonstrated continuous hopping for 20 minutes using a single set of batteries carried on board the robot. As with the Raibert Hopper, the Bow Leg Hopper controls velocity by changing the angle of the leg to the body at the hip joint. The paper of Ringrose [27] demonstrates the very important duality of mechanics and controls as applied to a single leg hopping machine. Often clever mechanical design can perform the same operations as complex active control circuitry. In this robot, the physical shape of the foot is exactly the right curve so that when the robot lands without being perfectly vertical, the proper corrective force is provided from the impact, making the robot vertical by the next landing. This robot is dynamically stable, and is furthermore passive.

The correction is provided by physical interactions between the robot and its environment, with no computer nor any active control in the loop.

Two Legs (Bipedal)

A variety of successful bipedal robots have been demonstrated. Two-legged robots have been shown to run, jump, travel up and down stairs and even do aerial tricks such as somersaults. **Fig.** 1.12 shows the Honda P2 bipedal robot, which is the product of tens of millions of research dollars and more than a decade of work. This biped can walk on slopes, climb and descend stairs, and push shopping carts. The crucial technology that enables this robot is Honda's research into the fabrication of extremely high torque, low mass motors that serve as the robot's joints. In the case of P2, the most significant obstacle that remains is energy capacity, efficiency and autonomous navigation. This robot can operate for only about 20 minutes with on-board power. An important feature of bipedal robots is their anthropomorphic shape. They can be built to have the same approximate dimensions

as humans, and this makes them excellent vehicles for research in human-robot interaction.

Bipedal robots can only be statically stable within some limits, and so robots such as P2 and Wabian generally must perform continuous balance-correcting servoing even when standing still. Furthermore, each leg must have sufficient capacity to support the full weight of the robot. In the case of four-legged robots, the balance problem is facilitated along with the load requirements of each leg. An elegant design of a biped robot is the Spring Flamingo of MIT seen in **Fig. 1.13**. This robot inserts springs in series with the leg actuators to achieve a more elastic gait. Combined with "kneecaps" that limit knee joint angles, the Flamingo achieves surprisingly biomimetic motion.

Four Legs (Quadruped)

Although standing still on four legs is passively stable, walking remains challenging because to remain stable the robot's center of gravity must be actively shifted during the gait. Sony recently invested several million dollars to develop a four-legged robot (figure 2.14). To create this robot, Sony created both a new robot operating system that is near real-time and invented new geared servomotors that are sufficiently high torque to support the robot, yet backdriveable for safety. In addition to developing custom motors and software, Sony incorporated a color vision system that enables Aibo to chase a brightly colored ball. The robot is able to function for at most one hour before requiring recharging. Early sales of the robot have been very strong, with more than 60,000 units sold in the first year. Nevertheless, the number of motors and the technology investment behind this robot dog have resulted in a very high price of approximately 1500. Four legged robots have the potential to serve as effective artifacts for research in human-robot interaction (fig. 2.15). Humans can treat the Sony robot, for example, as a pet and might develop an emotional relationship similar to that between man and dog. Furthermore, Sony has designed Aibo's walking style and general behavior to emulate learning and maturation, resulting in dynamic behavior over time that is more interesting for the owner who can track the changing behavior. As the challenges of high energy storage and motor technology are solved, it is likely that quadruped robots much more capable than Aibo will become common throughout the human environment.

Six Legs (Hexapod)

Six legged configurations have been extremely popular in mobile robotics because of their static stability during walking, thus reducing the control complexity (figure 2.16 and 2.17). In most cases,

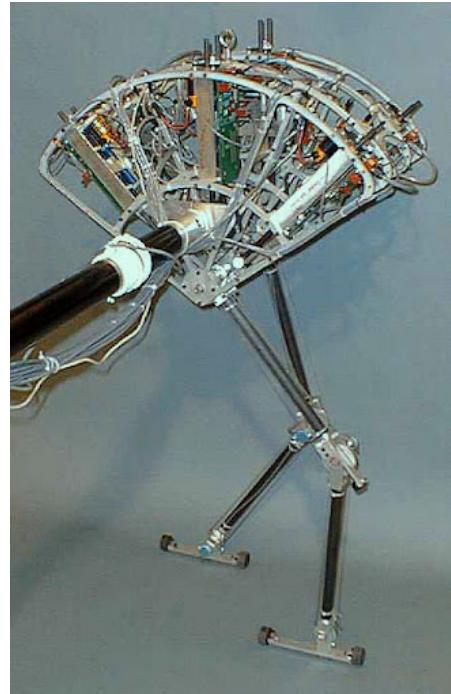


Figure 1.13: Spring Flamingo is a planar bipedal walking robot [28].

each leg has 3 DOF, including hip flexion, knee flexion and hip abduction (figure 2.6).

Genghis is a commercially available hobby robot that has six legs, each of which has 2 DOF provided by hobby servos (figure 2.18). Such a robot, which consists only of hip flexion and hip abduction, has less maneuverability in rough terrain but performs quite well on flat ground. Because it consists of a straightforward arrangement of servo motors and straight legs, such robots can be readily built by a robot hobbyist. Insects, which are arguably the most successful locomoting creatures on earth, excel at traversing all forms of terrain with six legs, even upside down. Currently, the gap between the capabilities of six-legged insects and artificial six-legged robots is still quite large. Interestingly, this is not due to a lack of sufficient numbers of degrees of freedom on the robots. Rather, insects combine a small number of active degrees of freedom with passive structures, such as microscopic barbs and textured pads, that increase the gripping strength of each leg significantly. Robotic research into such passive tip structures has only recently begun. For example, a research group is attempting to recreate the complete mechanical function of the cockroach leg (Roland, reference in notes (Espenschied et al.)). It is clear from all of the above examples that legged robots have much progress to make before they are competitive with the 24 Autonomous Mobile Robots have been realised recently, primarily due to advances in motor design. Creating actuation systems that approach the efficiency of animal muscle remains far from the reach of robotics, as does energy storage with the energy densities found in organic life forms



Figure 1.14: Genghis, one of the most famous walking robots from MIT uses hobby servomotors as its actuators.

1.3 Wheeled Mobile Robots

The wheel has been by far the most popular locomotion mechanism in mobile robotics and in man-made vehicles in general.⁹ It can achieve high efficiencies, as demonstrated in figure 2.3, and does so with a relatively simple mechanical implementation. In addition, balance is not usually a research problem in wheeled robot designs, because wheeled robots are almost always designed so that all wheels are in ground contact at all times.

⁹This should be clear as wheel motion is one of the most efficient method of converting energy to motion.

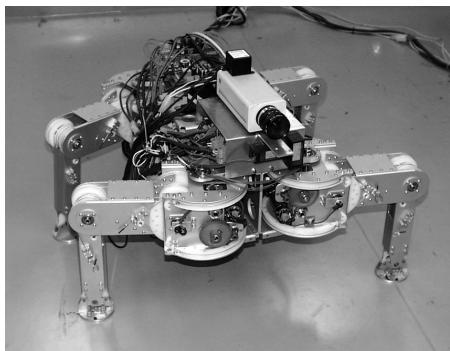


Figure 1.15: Genghis, one of the most famous walking robots from MIT uses hobby servomotors as its actuators.

Therefore, **three wheels are sufficient to guarantee stable balance**, although as we will see below, two-wheeled robots can also be stable.¹⁰ When more than three wheels are used, a suspension system is required in order to allow all wheels to maintain ground contact when the robot encounters uneven terrain. Instead of worrying about balance, researchers in wheeled robots tends to focus on the problems of **traction** and **stability**, maneuverability and control:

can the robot wheels provide sufficient traction and stability for the robot to cover all of the desired terrain, and does the robot's wheeled configuration enable sufficient control over the velocity of the robot?

¹⁰Of course, with clever implementation.

1.3.1 Design

As we will see, there is a very large space of possible wheel configurations when we consider possible techniques for mobile robot locomotion. We will begin by discussing the wheel in detail, as there are a number of different wheel types with specific strengths and weaknesses. Then, we will examine complete wheel configurations that deliver particular forms of locomotion for a mobile robot.

Wheel Design

There are four (4) major wheel classes, as shown in **Fig. 1.16**. They differ widely in their kinematics, and therefore the choice of wheel type has a large effect on the overall kinematics of the mobile robot.

The standard wheel and the castor wheel have a **primary axis of rotation** and therefore are highly directional. To move in a different direction, the wheel must be steered first along a vertical axis. The key difference between these two (2) wheels is that the standard wheel can accomplish this steering motion with no side effects, as the centre of rotation passes through the contact patch

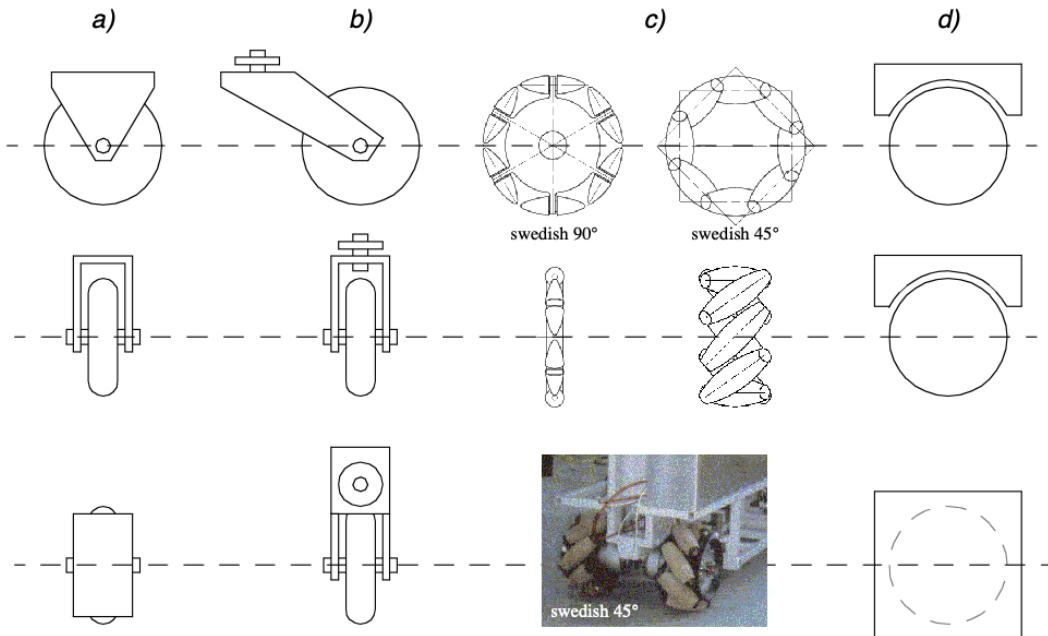


Figure 1.16: The four basic wheel types a)Standard wheel: Two degrees of freedom; rotation around the (motorized) wheel axle and the contact point b)castor wheel: Two degrees of freedom; rotation around an offset steering joint c)Swedish wheel: Three degrees of freedom; rotation around the (motorized) wheel axle, around the rollers and around the contact point

with the ground, while the castor wheel rotates around an offset axis, causing a force to be imparted to the robot chassis during steering.

¹¹Sometimes known as Swedish wheel or Ilon wheel after its inventor Bengt Erland Ilon [29].

The mecanum wheel¹¹ and the spherical wheel are both designs that are less constrained by directionality than the conventional standard wheel. The swedish wheel functions as a normal wheel, but provides low resistance in another direction as well, sometimes perpendicular to the conventional direction as in the swedish 90 and sometimes at an intermediate angle as in the swedish 45. The small rollers attached around the circumference of the wheel are passive and the wheel's primary axis serves as the only actively powered joint. The key advantage of this design is that, although the wheel rotation is powered only along the one principal axis (through the axle), the wheel can kinematically move with very little friction along many possible trajectories, not just forward and backward.

The spherical wheel is a truly omnidirectional wheel, often designed so that it may be actively powered to spin along any direction. One mechanism for implementing this spherical design imitates the computer mouse, providing actively powered rollers that rest against the top surface of the sphere and impart rotational force.

Regardless of what wheel is used, in robots designed for all-terrain environments and in robots with more than three (3) wheels, a suspension system is normally required to maintain wheel contact with the ground. One of the simplest approaches to suspension is to design flexibility into the wheel itself. For instance, in the case of some four-wheeled indoor robots that use castor wheels, manufacturers have applied a deformable tire of soft rubber to the wheel in order to create a

primitive suspension. Of course, this limited solution cannot compete with a sophisticated suspension system in applications where the robot needs a more dynamic suspension for significantly non-flat terrain.

Wheel Geometry

The choice of wheel types for a mobile robot is strongly linked to the choice of wheel arrangement, or wheel geometry. When designing a mobile robot locomotion we must consider these two (2) issues simultaneously. Why does wheel type and wheel geometry matter? Three fundamental characteristics of a robot are governed by these choices:

- maneuverability,
- controllability
- stability.

Unlike automobiles, which are largely designed for a highly standardised environment¹², mobile robots are designed for applications in a wide variety of situations. Automobiles all share similar wheel configurations as there is one region in the design space that maximises maneuverability, controllability and stability for their standard environment:

the paved road.

However, there is no single wheel configuration that maximises these qualities for the variety of environments faced by different mobile robots. So, we will see great variety in the wheel configurations of mobile robots. In fact, few robots use the Ackerman wheel configuration of the automobile because of its poor maneuverability, with the exception of mobile robots designed for the road system (figure 2.20).

Table 2.1 gives an overview of wheel configurations ordered by the number of wheels. This table shows both the selection of particular wheel types and their geometric configuration on the robot chassis. Note that some of the configurations shown are of little use in mobile robot applications. For instance, the 2-wheeled bicycle arrangement has moderate maneuverability and poor controllability. Like a single-leg hopping machine, it can never stand still. Nevertheless, this table provides an indication of the large variety of wheel configurations that are possible in mobile robot design.

1.3.2 Stability

Surprisingly, the minimum number of wheels required for static stability is two (2). As shown above, a two-wheel differential drive robot can achieve static stability if the center of mass is below the wheel axle. Cye is a commercial mobile robot that uses this wheel configuration

¹²such as the road network



Figure 1.17: NAVLAB I, the first autonomous highway vehicle that steers and controls the throttle using vision and radar sensors [30].

However, under ordinary circumstances such a solution requires wheel diameters that are impractically large. Dynamics can also cause a two-wheeled robot to strike the floor with a third point of contact, for instance with sufficiently high motor torques from standstill. Conventionally, static stability requires a minimum of three (3) wheels, with the additional caveat that the center of gravity must be contained within the triangle formed by the ground contact points of the wheels. Stability can be further improved by adding more wheels, although once the number of contact points exceeds three, the hyperstatic nature of the geometry will require some form of flexible suspension on uneven terrain.

1.3.3 Manoeuvrability

Some robots are omnidirectional, meaning that they can move at any time in any direction along the ground plane (X, Y) regardless of the orientation of the robot around its vertical axis. This level of maneuverability requires wheels that can move in more than just a single direction, and so omnidirectional robots usually employ swedish or spherical wheels that are powered. A good example is Uranus, shown in figure 2.24. This robot uses four swedish wheels to rotate and translate independently and without constraints. In general, the ground clearance of robots with swedish and spherical wheels is somewhat limited, due to the mechanical constraints of constructing omnidirectional wheels. An interesting recent solution to the problem of omnidirectional navigation while solving this ground clearance problem is the four castor-wheeled configuration in which each castor wheel is actively steered and actively translated. In this configuration, the robot is truly omnidirectional because, even if the castor wheels are facing a direction perpendicular to the desired direction of travel, the robot can still move in the desired direction by steering these wheels. Because the vertical axis is offset from the ground contact path, the result of this steering motion is robot motion.

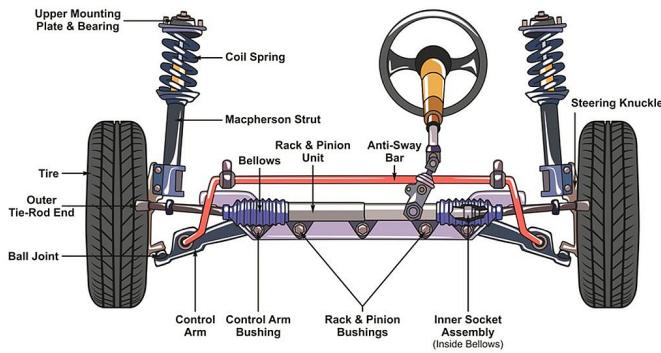


Figure 1.18: Example of an Ackerman drive used mostly in automotive industry [31].

In the research community, another classes of mobile robots are popular which achieve high maneuverability, only slightly inferior to that of the omnidirectional configurations. In such robots, motion in a particular direction may initially require a rotational motion. With a circular chassis and an axis of rotation at the center of the robot, such a robot can spin without changing its ground footprint. The most popular such robot is the two-wheel differential drive robot where the two wheels rotate around the center point of the robot. One or two additional ground contact points may be used for stability, based on the application specifics.

In contrast to the above configurations, consider the Ackerman steering configuration common in automobiles. Such a vehicle typically has a turning diameter that is larger than the car. Furthermore, for such a vehicle to move sideways requires a parking maneuver consisting of repeated changes in direction forward and backward. Nevertheless, Ackerman steering geometries have been especially popular in the hobby robotics market, where a robot can be built by starting with a remote-control race car kit and adding sensing and autonomy to the existing mechanism. In addition, the limited maneuverability of Ackerman steering has an important advantage: its directionality and steering geometry provide it with very good lateral stability in high speed turns.

1.3.4 Controllability

There is generally an **inverse correlation** between controllability and maneuverability. For example, the omni-directional designs such as the four castor-wheeled configuration require significant processing to convert desired rotational and translational velocities to individual wheel commands. Furthermore, such omni-directional designs often have greater degrees of freedom at the wheel. For instance, the swedish wheel has a set of free rollers along the wheel perimeter. These degrees of freedom cause an **accumulation of slippage**, tend to reduce dead-reckoning accuracy and increase the design complexity.

Controlling an omnidirectional robot for a specific direction of travel is also more difficult and often less accurate when compared to less manoeuvrable designs.

For example, an Ackerman steering vehicle can go straight simply by locking the steerable wheels and driving the drive wheels, which can be seen in **Fig.** 1.18.

In a differential drive vehicle, the two (2) motors attached to the two wheels must be driven along exactly the same velocity profile, which can be challenging considering variations between wheels, motors and environmental differences. With four-wheel omnidrive, such as the Uranus robot which has four swedish wheels, the problem is even harder because all four wheels must be driven at exactly the same speed for the robot to travel in a perfectly straight line.

In summary, there is **NO** “ideal” drive configuration that simultaneously maximises stability, manoeuvrability and controllability. Each mobile robot application places unique constraints on the robot design problem, and the designer’s task is to choose the most appropriate drive configuration possible from among this space of compromises.

1.3.5 Case Studies for Wheeled Motion

Now let’s describe four (4) specific wheel configurations, in order to demonstrate concrete applications of the concepts above to mobile robots built for real-world activities.

Synchro Drive

The synchro drive configuration (figure 2.22) is a popular arrangement of wheels in indoor mobile robot applications. It is an interesting configuration because, although there are three driven and steered wheels, only two motors are used in total. The one translation motor sets the speed of all three wheels together, and the one steering motor spins all the wheels together about each of their individual vertical steering axes. But note that the wheels are being steered with respect to the robot chassis, and therefore there is no direct way of re-orienting the robot chassis. In fact, the chassis orientation does drift over time due to uneven tire slippage, causing rotational dead-reckoning error.

Synchro drive is particularly advantageous in cases where omnidirectionality is needed. So long as each vertical steering axis is aligned with the contact path of each tire, the robot can always re-orient its wheels and move along a new trajectory without changing its footprint. Of course, if the robot chassis has directionality and the designers intend to re-orient the chassis purposefully, then synchro drive is only appropriate when combined with an independently rotating turret that attaches to the wheel chassis. Commercial research robots such as the Nomadics 150 or the RWI B21r have been sold with this configuration (figure 1.12). In terms of dead-reckoning, synchro drive systems are generally superior to true omni-directional configurations but inferior to differential drive and Ackerman steering systems. There are two main reasons for this. First and foremost, the translation motor generally drives the three wheels using a single belt. Due to slop and backlash in the drivetrain, whenever the drive motor engages, the closest wheel begins spinning before the furthest wheel, causing a small change in the orientation of the chassis. With additional changes in motor speed, these small angular shifts accumulate to create a large error in orientation during dead-reckoning.

Second, the mobile robot has no direct control over the orientation of the chassis. Depending on the orientation of the chassis, the wheel thrust can be highly asymmetric, with two wheels on one side and the third wheel alone, or symmetric, with one wheel on each side and one wheel straight ahead or behind, as shown in (2.22). The asymmetric cases results in a variety of errors when tire-ground slippage can occur, again causing errors in dead-reckoning of robot orientation.

Omnidirectional Drive

As we will see later in chapter 3.4.2, omnidirectional movement is of great interest for complete maneuverability. Omnidirectional robots that are able to move in any direction () at any time are also holonomic (see chapter 3.4.2). They can be realized by either using spheric, castor or swedish wheels. Three examples of such holonomic robots are presented below.

Omnidirectional locomotion with three spheric wheels

The omnidirectional robot depicted in figure 2.23 is based on three spheric wheels, each actuated by one motor. In this design, the spheric wheels are suspended by three contact points, two given by spherical bearings and one by the a wheel connected to the motor axle. This concept provides excellent maneuverability and is simple in design. However, it is limited to flat surfaces and small loads, and it is quite difficult to find round wheels with high friction coefficients

Omnidirectional locomotion with four swedish wheels

The omnidirectional arrangement depicted in figure 2.24 has been used successfully on several research robots, including the CMU Uranus. This configuration consists of four swedish 45 degree wheels, each driven by a separate motor. By varying the direction of rotation and relative speeds of the four wheels, the robot can be moved along any trajectory in the plane

and, even more impressively, can simultaneously spin around its vertical axis. For example, when all four wheels spin "forward" or "backward", the robot as a whole moves in a straight line forward and backward, respectively. However, when one diagonal pair of wheels is spun in the same direction and the other diagonal pair is spun in the opposite direction, the robot moves laterally. This four-wheel arrangement of swedish wheels is not minimal in terms of control motors. Because there are only 3 degrees of freedom in the plane, one can build a three-wheeled omnidirectional robot chassis using three swedish 90 degree wheels as shown in Table 2.1. However, existing examples such as Uranus have been designed with four wheels due to capacity and stability considerations. One application for which such omnidirectional designs are particular amenable is mobile manipulation. In this case, it is desirable to reduce the degrees of freedom of the manipulator arm to save arm mass by using the mobile robot chassis motion for gross motion. As with humans, it would be ideal if the base could move omnidirectionally without greatly impact-

Omnidirectional locomotion with four castor wheels and eight motors

Another solution for omnidirectionality is to use castor wheels. This is done for the Nomad XR4000 from Nomadics (fig. 2.25) giving it an excellent maneuverability. Unfortunately Nomadics Technology has ceased the production of mobile robots. The above two examples are drawn from Table 2.1, but this is not an exhaustive list of all wheeled locomotion techniques. Hybrid approaches that combine legged and wheeled locomotion, or tracked and wheeled locomotion, can also offer particular advantages. Below are two unique designs created for specialized applications.

Tracked Slip/Skid Locomotion

In the wheel configurations discussed above, we have made the assumption that wheels are not allowed to skid against the surface. An alternative form of steering, termed slip/skid, may be used to re-orient the robot by spinning wheels that are facing the same direction at different speeds or in opposite directions. The army tank operates this way, and Nanokhod, pictured below (figure 2.26) is an example of a mobile robot based on the same concept. Robots that make use of tread have much larger ground contact patches, and this can significantly improve their maneuverability in loose terrain compared to conventional wheeled designs. However, due to this large ground contact patch, changing the orientation of the robot usually requires a skidding turn, wherein a large portion of the track must slide against the terrain. The disadvantage of such configurations is coupled to the slip/skid steering. Because of the large amount of skidding during a turn, the exact center of rotation of the robot is hard to predict and the exact change in position and orientation is also subject to variations depending on the ground friction. Therefore, dead-reckoning on such robots is highly inaccurate. This is the trade-off that is made in return for extremely good maneuverability and traction over rough and loose terrain. Furthermore, a slip/skid approach on a high-friction surface can quickly overcome the torque capabilities of the motors being used. In terms of power efficiency, this approach is reasonably efficient on loose terrain but extremely inefficient otherwise.

1.3.6 Walking Wheels

Walking robots might offer the best maneuverability in rough terrain. However, they are inefficient on flat ground and need sophisticated control. Hybrid solutions, combining the adaptability of legs with the efficiency of wheels offer an interesting compromise. Solutions that passively adapt to the terrain are of particular interest for field and space robotics. The Sojourner robot of NASA/JPL (fig. 1.2) represents such a hybrid solution, able to overcome objects up to the size of the wheels. A more advanced mobile robot design for similar applications has recently been produced by EPFL (fig. 2.27). This robot, called Shrimp2, has 6 motorized wheels and is capable of climbing objects up to two times its wheel diameter [84,85]. This enables it to climb regular stairs though the robot is even smaller than the Sojourner. Using a rhombus configuration, the Shrimp has a steering wheel in the front and the rear, and two wheels arranged on a bogie on each side. The front wheel has a spring suspension to guarantee optimal ground contact of all wheels at any time. The

steering of the rover is realized by synchronizing the steering of the front and rear wheels and the speed difference of the bogie wheels. This allows for high precision maneuvers and turning on the spot with minimum slip/skid of the four center wheels. The use of parallel articulations for the front wheel and the bogies creates a virtual center of rotation at the level of the wheel axis. This ensures maximum stability and climbing abilities even for very low friction coefficients between the wheel and the ground. As mobile robotics research matures we find ourselves able to design more intricate mechanical systems. At the same time, the control problems of inverse kinematics and dynam2 Locomotion 37 R. Siegwart, EPFL, Illah Nourbakhsh, CMU ics are now so readily conquered that these complex mechanics can in general be controlled. So, in the near future, you should expect to see a great number of unique, hybrid mobile robots that draw together advantages from several of the underlying locomotion mechanisms that we have discussed in this chapter. They will each be technologically impressive, and each will be designed as the expert robot for its particular environmental niche.

Chapter 2

Perception

Table of Contents

2.1	Introduction	27
2.2	Active Ranging	40
2.3	Vision Based Sensors	51
2.4	Feature Extraction	61

2.1 Introduction

One of the most important tasks of an AMR is to acquire knowledge about its environment.¹ This is achieved by taking measurements using various sensors and then extracting meaningful information from those measurements.

In this chapter we present the most common sensors used in AMR and then discuss strategies for extracting information from the sensors.

¹One could even argue it is the definition of life, if you ask a biologist as the ability to feel and act on its environment is the bare necessity.

2.1.1 Sensors for Mobile Robotics

There is a wide variety of sensors used in AMRs (Fig. 4.1). Some are used to measure simple values like the internal temperature of a robot's electronics or the rotational speed of the motors in its wheels or actuators. Other, more sophisticated sensors can be used to acquire information about the robot's environment or even to directly measure a robot's global position. Here, we focus primarily on sensors used to extract information about the robot's environment. Because a AMR moves around, it will frequently encounter **unforeseen** environmental characteristics, and therefore such sensing is particularly critical. We begin with a functional classification of sensors. Then, after

presenting basic tools for describing a sensor's performance, we proceed to describe selected sensors in detail.

2.1.2 Sensor Classification

We classify sensors using two (2) important functional axes. Let's define these terms for clarity;

Proprioceptive sensors which measure values **internal** to the robot.

e.g., motor speed, wheel load, robot arm joint angles, battery voltage.

Exteroceptive sensors which measure information from the **robot's environment**;

e.g., distance measurements, light intensity, sound amplitude.

exteroceptive sensor measurements are interpreted by the robot to extract meaningful environmental features.

Passive sensors measure ambient environmental energy entering the sensor.

e.g., temperature probes, microphones and CCD or CMOS cameras.

Active sensors emit energy into the environment, then measure the environmental reaction. Because active sensors can manage more controlled interactions with the environment, they often achieve superior performance. However, active sensing introduces several risks: the outbound energy may affect the very characteristics that the sensor is attempting to measure. Furthermore, an active sensor may suffer from interference between its signal and those beyond its control. For example, signals emitted by other nearby robots, or similar sensors on the same robot may influence the resulting measurements. Examples of active sensors include wheel quadrature encoders, ultrasonic sensors and laser rangefinders.

The sensor classes in Table (4.1) are arranged in ascending order of complexity and descending order of technological maturity. Tactile sensors and proprioceptive sensors are critical to virtually all mobile robots, and are well understood and easily implemented. Commercial quadrature encoders, for example, may be purchased as part of a gear-motor assembly used in a AMR. At the other extreme, visual interpretation by means of one or more CCD/CMOS cameras provides a broad array of potential functionalities, from obstacle avoidance and localisation to human face recognition. However, commercially available sensor units that provide visual functionalities are only now beginning to emerge

2.1.3 Characterising Sensor Performance

The sensors we describe in this chapter vary greatly in their performance characteristics. Some sensors provide extreme accuracy in well-controlled laboratory settings, but are overcome with error when subjected to real-world environmental variations. Other sensors provide narrow, high precision data in a wide variety settings. To quantify such performance characteristics, first we formally define the sensor performance terminology that will be valuable throughout the rest of this chapter.

Basic Sensor Response Ratings

A number of sensor characteristics can be rated **quantitatively** in a laboratory setting. Such performance ratings will necessarily be best-case scenarios when the sensor is placed on a real-world robot, but are nevertheless useful.

Dynamic Range Used to measure the spread between the lower and upper limits of inputs values to the sensor while maintaining normal sensor operation. Formally, the dynamic range is the ratio of the maximum input value to the minimum measurable input value. Because this raw ratio can be unwieldy, it is usually measured in Decibels, which is computed as ten times the common logarithm of the dynamic range. However, there is potential confusion in the calculation of Decibels, which are meant to measure the ratio between powers, such as Watts or Horsepower.

Suppose your sensor measures motor current and can register values from a minimum of 1 mA to 20 A. The dynamic range of this current sensor is defined as:

$$10 \cdot \log \left[\frac{20}{0.001} \right] = 43 \text{ dB} \quad (2.1)$$

Now suppose you have a voltage sensor that measures the voltage of your robot's battery, measuring any value from 1 mV to 20 V. Voltage is **NOT** a unit of power, but the square of voltage is proportional to power. Therefore, we use 20 instead of 10:

$$20 \cdot \log \left[\frac{20}{0.001} \right] = 86 \text{ dB} \quad (2.2)$$

Range An important rating in AMR because often robot sensors operate in environments where they are frequently exposed to input values beyond their working range. In such cases, it is critical to understand how the sensor will respond. For example, an optical rangefinder will have a minimum operating range and can thus provide spurious data when measurements are taken with object closer than that minimum.

Resolution The minimum difference between two (2) values that can be detected by a sensor. Usually, the lower limit of the dynamic range of a sensor is equal to its resolution. However, in the case of digital sensors, this is not necessarily so. For example, suppose that you have a sensor that measures voltage, performs an analogue-to-digital conversion and outputs the

converted value as an 8-bit number linearly corresponding to between 0 and 5 Volts. If this sensor is truly linear, then it has $2^8 - 1$ total output values or a resolution of:

$$\frac{5}{255} = 20 \text{ mV}$$

Linearity is an important measure governing the behaviour of the sensor's output signal as the input signal varies. A linear response indicates that if two (2) inputs, say x and y result in the two outputs $f(x)$ and $f(y)$, then for any values a and b , the following relation can be derived:

$$f(x + y) = f(x) + f(y).$$

This means that a plot of the sensor's input/output response is simply a straight line.

Bandwidth or Frequency is used to measure the speed with which a sensor can provide a stream of readings. Formally, the number of measurements per second is defined as the sensor's frequency in Hz. Because of the dynamics of moving through their environment, mobile robots often are limited in maximum speed by the bandwidth of their obstacle detection sensors. Thus increasing the bandwidth of ranging and vision-based sensors has been a high-priority goal in the robotics community.

In Situ Sensor Performance

The above sensor characteristics can be reasonably measured in a laboratory environment, with confident extrapolation to performance in real-world deployment. However, a number of important measures cannot be reliably acquired without deep understanding of the complex interaction between all environmental characteristics and the sensors in question. This is most relevant to the most sophisticated sensors, including active ranging sensors and visual interpretation sensors.

Sensitivity A measure of the degree to which an incremental change in the target input signal changes the output signal. Formally, sensitivity is the ratio of output change to input change. Unfortunately, however, the sensitivity of exteroceptive sensors is often confounded by undesirable sensitivity and performance coupling to other environmental parameters.

Cross-Sensitivity is the technical term for sensitivity to environmental parameters that are orthogonal to the target parameters for the sensor. For example, a flux-gate compass can demonstrate high sensitivity to magnetic north and is therefore of use for AMR navigation. However, the compass will also demonstrate high sensitivity to ferrous building materials, so much so that its cross-sensitivity often makes the sensor useless in some indoor environments. High cross-sensitivity of a sensor is generally undesirable, especially so when it cannot be modelled.

Error of a sensor is defined as the difference between the sensor's output measurements and the true values being measured, within some specific operating context.

As an example, given a true value v and a measured value m , we can define error as:

$$\text{Error} = m - v.$$

Accuracy defined as the degree of conformity between the sensor's measurement and the true value, and is often expressed as a proportion of the true value (e.g. 97.5% accuracy):

$$\text{Accuracy} = 1 - \frac{|m - v|}{v}.$$

Of course, obtaining the ground truth (v), can be difficult or impossible, and so establishing a confident characterisation of sensor accuracy can be problematic. Further, it is important to distinguish between two different sources of error:

- Systematic errors are caused by factors or processes that can in theory be modelled. These errors are, therefore, deterministic.²

Poor calibration of a laser rangefinder, un-modelled slope of a hallway floor and a bent stereo camera head due to an earlier collision are all possible causes of systematic sensor errors

²Meaning, its value is not determined by a random process and therefore should, in theory, be predictable.

- Random errors cannot be predicted using a sophisticated model nor can they be mitigated with more precise sensor machinery. These errors can only be described in probabilistic terms (i.e. stochastic). Hue instability in a colour camera, spurious range-finding errors and black level noise in a camera are all examples of random errors.

Precision is often confused with accuracy, and now we have the tools to clearly distinguish these two terms. Intuitively, high precision relates to reproducibility of the sensor results. For example, one sensor taking multiple readings of the same environmental state has high precision if it produces the same output. In another example, multiple copies of this sensors taking readings of the same environmental state have high precision if their outputs agree. Precision does not, however, have any bearing on the accuracy of the sensor's output with respect to the true value being measured. Suppose that the random error of a sensor is characterised by some mean value (μ) and a standard deviation (σ). The formal definition of precision is the ratio of the sensor's output range to the standard deviation:

$$\text{Precision} = \frac{\text{Range}}{\sigma}.$$

Only σ and **NOT** μ has impact on precision. In contrast mean error is directly proportional to overall sensor error and inversely proportional to sensor accuracy.

Characterising Error

Mobile robots depend heavily on **exteroceptive** sensors. Many of these sensors concentrate on a central task for the robot:

acquiring information on objects in the robot's immediate vicinity so that it may interpret the state of its surroundings.

³In this case we are referring to the robot reference frame.

Of course, these “objects” surrounding the robot are all detected from the viewpoint of its local reference frame.³ Since the systems we study are **mobile**, their ever-changing position and their motion has a significant impact on overall sensor behaviour.

Now that we have the necessary knowledge on the fundamental concepts and terminology, we can now describe how dramatically the sensor error of an AMR **disagrees** with the ideal picture drawn in the previous section.

Blurring of Systematical and Random Errors

Active ranging sensors tend to have failure modes which are triggered largely by specific relative positions of the sensor and environment targets.

⁴The incident light is reflected into a single outgoing direction.

For example, a sonar sensor will product specular reflections⁴ producing grossly inaccurate measurements of range, at specific angles to a smooth sheet-rock wall.

During motion of the robot, such relative angles occur at stochastic intervals. This is especially true in a AMR outfitted with a ring of multiple sonars. The chances of one sonar entering this error mode during robot motion is high. From the perspective of the moving robot, the sonar measurement error is a **random error** in this case. However, if the robot were to stop, becoming motionless, then a very different error modality is possible.

If the robot's static position causes a particular sonar to fail in this manner, the sonar will fail consistently and will tend to return precisely the same (and incorrect!) reading time after time. Once the robot is motionless, the error appears to be systematic and high precision.

The fundamental mechanism at work here is the cross-sensitivity of AMR sensors to robot pose and robot-environment dynamics.

The models for such cross-sensitivity are **NOT**, in an underlying sense, truly random. However, these physical interrelationships are rarely modelled and therefore, from the point of view of an incomplete model, the errors appear random during motion and systematic when the robot is at rest. Sonar is not the only sensor subject to this blurring of systematic and random error modality. Visual interpretation through the use of a CCD camera is also highly susceptible to robot motion and position because of camera dependency on lighting.⁵

⁵such as glare and reflections.

The important point is to realise that, while systematic error and random error are well-defined in a controlled setting, the AMR can exhibit error characteristics that bridge the gap between deterministic and stochastic error mechanisms.

Multi-Modal Error Distributions

It is common to characterise the behaviour of a sensor's random error in terms of a probability distribution over various output values. In general, one knows very little about the causes of random error and therefore several simplifying assumptions are commonly used. For example, we can assume that the error is zero-mean ($\mu = 0$), in that it symmetrically generates both positive and negative measurement error. We can go even further and assume that the probability density curve is Gaussian. Although we discuss the mathematics of this in detail later, it is important for now to recognise the fact that one frequently assumes symmetry as well as unimodal distribution. This means that measuring the correct value is most probable, and any measurement that is further away from the correct value is less likely than any measurement that is closer to the correct value. These are strong assumptions that enable powerful mathematical principles to be applied to AMR problems, but it is important to realise how wrong these assumptions usually are.

Consider, for example, the sonar sensor once again. When ranging an object that reflects the sound signal well, the sonar will exhibit high accuracy, and will induce random error based on noise, for example, in the timing circuitry. This portion of its sensor behaviour will exhibit error characteristics that are fairly **symmetric** and **unimodal**. However, when the sonar sensor is moving through an environment and is sometimes faced with materials that cause coherent reflection rather than returning the sound signal to the sonar sensor, then the sonar will grossly overestimate distance to the object. In such cases, the error will be biased toward positive measurement error and will be far from the correct value. The error is not strictly systematic, and so we are left modelling it as a probability distribution of random error. So the sonar sensor has two (2) separate types of operational modes, one in which the signal does return and some random error is possible, and the second in which the signal returns after a multi-path reflection, and gross overestimation error occurs. The probability distribution could easily be at least bimodal in this case, and since overestimation is more common than underestimation it will also be asymmetric.

As a second example, consider ranging via stereo vision. Once again, we can identify two (2) modes of operation. If the stereo vision system correctly correlates two images, then the resulting random error will be caused by camera noise and will limit the measurement accuracy. But the stereo vision system can also correlate two images incorrectly, matching two fence posts for example that are not the same post in the real world. In such a case stereo vision will exhibit gross measurement error, and one can easily imagine such behaviour violating both the unimodal and the symmetric assumptions. The thesis of this section is that sensors in a AMR may be subject to multiple modes of operation and, when the sensor error is characterised, uni modality and symmetry may be grossly violated. Nonetheless, as you will see, many successful AMR systems make use of these simplifying assumptions and the resulting mathematical techniques with great empirical success. The above sections have presented a terminology with which we can characterise the advantages and disadvantages of various mobile robot sensors. In the following sections, we do the same for a sampling of the most commonly used AMR sensors today.

2.1.4 Wheel and Motor Sensors

Wheel/motor sensors are devices used to measure the internal state and dynamics of a mobile robot. These sensors have vast applications outside of AMR and, as a result, AMR has enjoyed the benefits of high-quality, low-cost wheel and motor sensors which offer excellent resolution.

In the next part, we sample just one such sensor, the optical incremental encoder.

Optical Encoders

Optical incremental encoders have become the most popular device for measuring angular speed and position within a motor drive or at the shaft of a wheel or steering mechanism. In mobile robotics, encoders are used to control the position or speed of wheels and other motor-driven joints. Because these sensors are proprioceptive, their estimate of position is best in the reference frame of the robot and, when applied to the problem of robot localisation, significant corrections are required as discussed in Chapter 5.

An optical encoder is basically a mechanical light chopper that produces a certain number of sine or square wave pulses for each shaft revolution. It consists of an illumination source, a fixed grating that masks the light, a rotor disc with a fine optical grid that rotates with the shaft, and fixed optical detectors. As the rotor moves, the amount of light striking the optical detectors varies based on the alignment of the fixed and moving gratings. In robotics, the resulting sine wave is transformed into a discrete square wave using a threshold to choose between light and dark states. Resolution is measured in Cycles Per Revolution (CPR). The minimum angular resolution can be readily computed from an encoder's CPR rating. A typical encoder in AMR may have 2,000 CPR while the optical encoder industry can readily manufacture encoders with 10,000 CPR. In terms of required bandwidth, it is of course critical that the encoder be sufficiently fast to count at the shaft spin speeds that are expected. Industrial optical encoders present no bandwidth limitation to AMR applications. Usually in AMR the quadrature encoder is used. In this case, a second illumination and detector pair is placed 90° shifted with respect to the original in terms of the rotor disc. The resulting twin square waves, shown in Fig. 4.2, provide significantly more information. The ordering of which square wave produces a rising edge first identifies the direction of rotation. Furthermore, the four detectability different states improve the resolution by a factor of four with no change to the rotor disc. Thus, a 2,000 CPR encoder in quadrature yields 8,000 counts. Further improvement is possible by retaining the sinusoidal wave measured by the optical detectors and performing sophisticated interpolation. Such methods, although rare in AMR, can yield 1000-fold improvements in resolution. As with most proprioceptive sensors, encoders are generally in the controlled environment of a AMR's internal structure, and so systematic error and cross-sensitivity can be engineered away. The accuracy of optical encoders is often assumed to be



Figure 2.1: An example of a rotary encoder. [32]

100% and, although this may not entirely correct, any errors at the level of an optical encoder are dwarfed by errors downstream of the motor shaft.

Heading Sensors

Heading sensors can be proprioceptive (gyroscope, inclinometer) or exteroceptive (compass). They are used to determine the robot's orientation and inclination. They allow us, together with appropriate velocity information, to integrate the movement to a position estimate. This procedure, which has its roots in vessel and ship navigation, is called dead reckoning.

Compasses

The two most common modern sensors for measuring the direction of a magnetic field are the Hall Effect and Flux Gate compasses. Each has advantages and disadvantages, as described below. The Hall Effect describes the behaviour of electric potential in a semiconductor when in the presence of a magnetic field. When a constant current is applied across the length of a semiconductor, there will be a voltage difference in the perpendicular direction, across the semiconductor's width, based on the relative orientation of the semiconductor to magnetic flux

lines. In addition, the sign of the voltage potential identifies the direction of the magnetic field. Thus, a single semiconductor provides a measurement of flux and direction along one dimension. Hall Effect digital compasses are popular in AMR, and contain two such semiconductors at right angles, providing two axes of magnetic field (thresholded) direction, thereby yielding one of 8 possible compass directions. The instruments are inexpensive but also suffer from a range of disadvantages. Resolution of a digital hall effect compass is poor. Internal sources of error include the nonlinearity of the basic sensor and systematic bias errors at the semiconductor level. The resulting circuitry must perform significant filtering, and this lowers the bandwidth of hall effect

compasses to values that are slow in AMR terms. For example the hall effect compasses pictured in figure 4.3 needs 2.5 seconds to settle after a 90° spin. The Flux Gate compass operates on a different principle. Two small coils are wound on ferrite cores and are fixed perpendicular to one-another. When alternating current is activated in both coils, the magnetic field causes shifts in the phase depending upon its relative alignment with each coil. By measuring both phase shifts, the direction of the magnetic field in two dimensions can be computed. The flux-gate compass can accurately measure the strength of a magnetic field and has improved resolution and accuracy; however it is both larger and more expensive than a Hall Effect compass. Regardless of the type of compass used, a major drawback concerning the use of the Earth's magnetic field for AMR applications involves disturbance of that magnetic field by other magnetic objects and man-made structures, as well as the bandwidth limitations of electronic compasses and their susceptibility

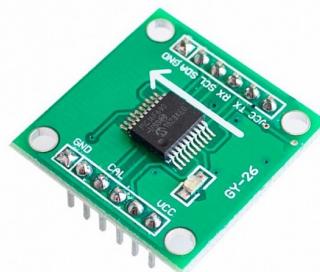


Figure 2.2: An example of an electronic compass [33].

to vibration. Particularly in indoor environments AMR applications have often avoided the use of compasses, although a compass can conceivably provide useful local orientation information indoors, even in the presence of steel structures.

Gyroscope

Gyroscopes are heading sensors which preserve their orientation in relation to a fixed reference frame. Thus they provide an absolute measure for the heading of a mobile system. Gyroscopes can be classified in two categories, mechanical gyroscopes and optical gyroscopes.

Mechanical Gyroscopes

The concept of a mechanical gyroscope relies on the inertial properties of a fast spinning rotor. The property of interest is known as the gyroscopic precession. If you try to rotate a fast spinning wheel around its vertical axis, you will feel a harsh reaction in the horizontal axis. This is due to the angular momentum associated with a spinning wheel and will keep the axis of the gyroscope inertially stable. The reactive torque τ and thus the tracking stability with the inertial frame are proportional to the spinning speed ω , the precession speed Ω and the wheel's inertia I .

$$\tau = I\omega\Omega$$

By arranging a spinning wheel as seen in Figure 4.4, no torque can be transmitted from the outer pivot to the wheel axis. The spinning axis will therefore be space-stable (i.e. fixed in an inertial reference frame). Nevertheless, the remaining friction in the bearings of the gyro-axis introduce small torques, thus limiting the long term space stability and introducing small errors over time. A high quality mechanical gyroscope can cost up to \$100,000 and has an angular drift of about 0.1̄ in 6 hours. For navigation, the spinning axis has to be initially selected. If the spinning axis is aligned with the north-south meridian, the earth's rotation has no effect on the gyro's horizontal axis. If it points east-west, the horizontal axis reads the earth rotation. Rate gyros have the same basic arrangement as shown in Figure 4.4 but with a slight modification. The gimbals are restrained by a torsional spring with additional viscous damping. This enables the sensor to measure angular speeds instead of absolute orientation.

Optical Gyroscopes

Optical gyroscopes are a relatively new innovation. Commercial use began in the early 1980's when they were first installed in aircraft. Optical gyroscopes are angular speed sensors that use two monochromatic light beams, or lasers, emitted from the same source instead of moving, mechanical parts. They work on the principle that the speed of light remains unchanged and, therefore, geometric change can cause light to take a varying amount of time to reach its destination. One laser beam is sent traveling clockwise through a fiber while the other travels counterclockwise. Because the laser

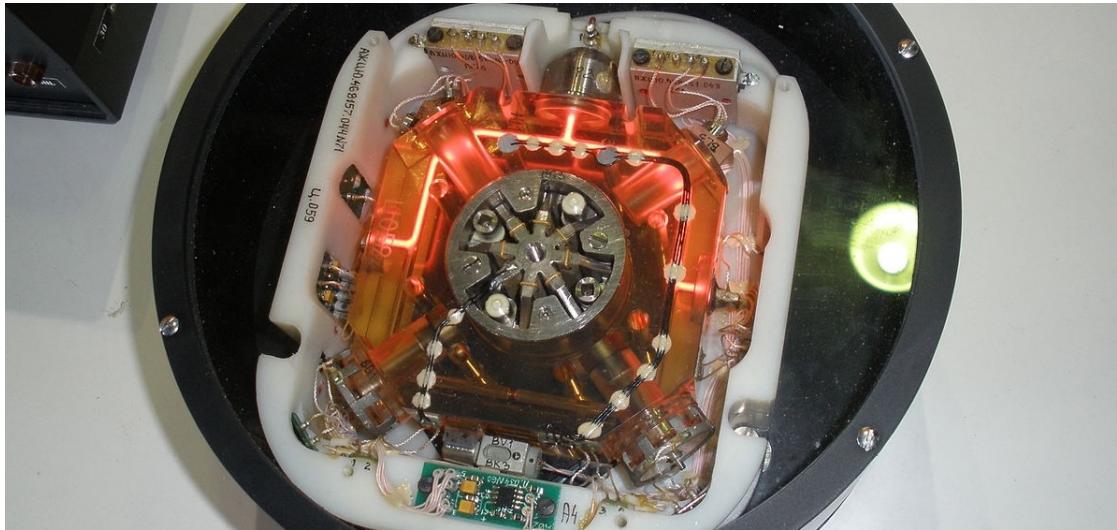


Figure 2.3: Optical Gyroscopes have no moving parts, (unlike mechanical gyroscopes) making them extremely reliable [34].

traveling in the direction of rotation has a slightly shorter path, it will have a higher frequency. The difference in frequency of the two beams is proportional to the angular velocity of the cylinder. New solid-state optical gyroscopes based on the same principle are built using microfabrication technology, thereby providing heading information with resolution and bandwidth far beyond the needs of mobile robotic applications. Bandwidth, for instance, can easily exceed 100KHz while resolution can be smaller than 0.0001°/hr.

Ground Based Beacons



Figure 2.4

One elegant approach to solving the localization problem in AMR is to use active or passive beacons. Using the interaction of on-board sensors and the environmental beacons, the robot can identify its position precisely. Although the general intuition is identical to that of early human navigation beacons, such as stars, mountains and lighthouses, modern technology has enabled sensors to localize

an outdoor robot with accuracies of better than 5 cm within areas that are kilometres in size.

In the following subsection, we describe one such beacon system, the Global Positioning System (GPS), which is extremely effective for outdoor ground-based and flying robots. In-door beacon systems have been generally less successful for a number of reasons. The expense of environmental modification in an indoor setting is not amortized over an extremely large useful area, as it is for example in the case of GPS. Furthermore, indoor environments offer significant challenges not seen outdoors, including multipath and environment dynamics. A laser-based indoor beacon system, for example, must disambiguate the one true laser signal from possibly tens of other powerful signals that have reflected off of walls, smooth floors and doors. Confounding this, humans and other obstacles may be constantly changing the environment, for example occluding the one true path from the beacon to the robot. In commercial applications such as manufacturing plants, the environment can be carefully controlled to ensure success. In less structured indoor settings, beacons have nonetheless been used, and the problems are mitigated by careful beacon placement and the use of passive sensing modalities.

Global Positioning System

The Global Positioning System (GPS) was initially developed for military use but is now freely available for civilian navigation. There are at least 24 operational GPS satellites at all times. The satellites orbit every 12 hours at a height of 20.190km. There are four (4) satellites located in each of six planes inclined 55° with respect to the plane of the earth's equator (figure 4.5).

Each satellite continuously transmits data which indicates its location and the current time. Therefore, GPS receivers are **completely passive** but **exteroceptive** sensors. The GPS satellites synchronise their transmissions to allow their signals to be sent at the same time. When a GPS receiver reads the transmission of two (2) or more satellites, the arrival time differences inform the receiver as to its relative distance to each satellite.

By combining information regarding the arrival time and instantaneous location of four (4) satellites, the receiver can infer its own position.

In theory, such triangulation requires only three (3) data points. However, timing is extremely critical in the GPS application because the time intervals being measured are in ns.

It is, of course, mandatory the satellites to be well synchronised. To this end, they are updated by ground stations regularly and each satellite carries on-board atomic clocks⁶ for timing. The GPS receiver clock is also important so that the travel time of each satellite's transmission can be accurately measured. But GPS receivers have a simple quartz clock. So, although 3 satellites would ideally provide position in three axes, the GPS receiver requires 4 satellites, using the additional information to solve for 4 variables: three position axes plus a time correction. The fact that the GPS receiver must read the transmission of 4 satellites simultaneously is a significant limitation. GPS satellite transmissions are extremely low-power, and reading them successfully requires direct



⁶An example of a cesium clock for use in GPS.

line-of-sight communication with the satellite. Thus, in confined spaces such as city blocks with tall buildings or dense forests, one is unlikely to receive 4 satellites reliably. Of course, most indoor spaces will also fail to provide sufficient visibility of the sky for a GPS receiver to function. For these reasons, GPS has been a popular sensor in AMR, but has been relegated to projects involving AMR traversal of wide-open spaces and autonomous flying machines. A number of factors affect the performance of a localization sensor that makes use of GPS. First, it is important to understand that, because of the specific orbital paths of the GPS satellites, coverage is not geometrically identical in different portions of the Earth and therefore resolution is not uniform. Specifically, at the North and South poles, the satellites are very close to the horizon and, thus, while resolution in the latitude and longitude directions is good, resolution of altitude is relatively poor as compared to more equatorial locations.

The second point is that GPS satellites are merely an information source. They can be employed with various strategies in order to achieve dramatically different levels of localisation resolution. The basic strategy for GPS use, called pseudorange and described above, generally performs at a resolution of 15m. An extension of this method is differential GPS, which makes use of a second receiver that is static and at a known exact position. A number of errors can be corrected using this reference, and so resolution improves to the order of 1m or less. A disadvantage of this technique is that the stationary receiver must be installed, its location must be measured very carefully and of course the moving robot must be within kilometers of this static unit in order to benefit from the DGPS technique. A further improved strategy is to take into account the phase of the carrier signals of each received satellite transmission. There are two carriers, at 19cm and 24cm, therefore significant improvements in precision are possible when the phase difference between multiple satellites is measured successfully. Such receivers can achieve 1cm resolution for point positions and, with the use of multiple receivers as in DGPS, sub-1cm resolution. A final consideration for AMR applications is bandwidth. GPS will generally offer no better than 200 - 300ms latency, and so one can expect no better than 5Hz GPS updates. On a fast-moving AMR or flying robot, this can mean that local motion integration will be required for proper control due to GPS latency limitations.

2.2 Active Ranging

Active range sensors continue to be the most popular sensors used in AMR. Many ranging sensors have a low price point, and most importantly all ranging sensors provide easily interpreted outputs:

Direct measurements of distance from the robot to objects in its vicinity.

For obstacle detection and avoidance, most AMR rely heavily on active ranging sensors. But the local free-space information provided by range sensors can also be accumulated into representations beyond the robot's current local reference frame. Therefore, active range sensors are also commonly found as part of the localisation and environmental modelling processes of AMRs.

It is only with the slow advent of successful visual interpretation competency that we can expect the class of active ranging sensors to gradually lose their primacy as the sensor class of choice among AMR engineers.

Below, we present two (2) Time-of-Flight (ToF) active range sensors:

- the ultrasonic sensor,
- the laser rangefinder.

Continuing onwards, we then present two (2) geometric active range sensors:

- the optical triangulation sensor,
- the structured light sensor.

Time-of-Flight Active Ranging

ToF ranging makes use of the [propagation speed of sound](#) or an [electromagnetic wave](#). In general, the travel distance of a sound or electromagnetic wave is given by:

$$d = ct,$$

where d is the distance travelled usually round-trip (m), c the speed of wave propagation (ms^{-1}), and t is the time it takes to travel (s).

It is important to point out the propagation speed v of sound is approximately 0.3 m ms^{-1} whereas the speed of an electromagnetic signal is 0.3 m ns^{-1} , which is one million times faster. The ToF for a typical distance, say 3 m, is 10 ms for an ultrasonic system but only 10 ns for a laser rangefinder. It is therefore obvious that measuring the time of flight t with electromagnetic signals is more technologically challenging.⁷

The quality of ToF range sensors depends mainly on the following:

⁷This explains why laser range sensors have only recently become affordable and robust for use on mobile robots.

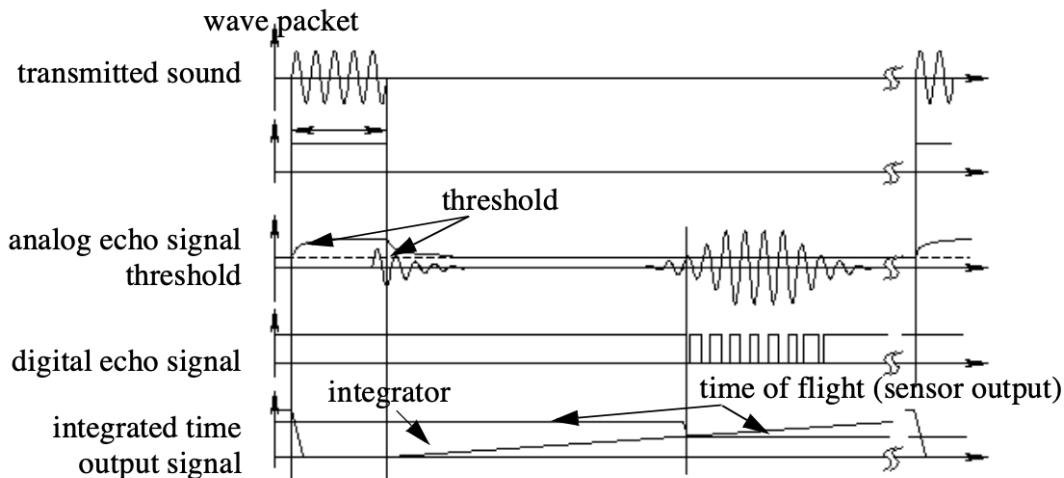


Figure 2.5: Signals of an ultrasonic sensor.

- Uncertainties in determining the exact time of arrival of the reflected signal,
- Inaccuracies in the time of flight measurement, particularly with laser range sensors,
- The dispersal cone of the transmitted beam mainly with ultrasonic range sensors
- Interaction with the target (e.g., surface absorption, specular reflections)
- Variation of propagation speed, and
- The speed of the AMR and target (in the case of a dynamic target).

As discussed below, each type of ToF sensor is sensitive to a particular subset of the above list of factors.

2.2.1 The Ultrasonic Sensor

The main ethos of an ultrasonic⁸ sensor is to transmit a packet of ultrasonic pressure waves and to measure the time it takes for this wave to reflect and return to the receiver. The distance d of the object causing the reflection can be calculated based on the propagation speed of sound⁹ c and the time of flight t .

$$d = \frac{c \times t}{2}$$

The speed of sound (v) in air is given by the following relation:

$$v = \sqrt{\gamma RT}$$

where γ is the ratio of specific heat, R is the gas constant ($\text{J mol}^{-1} \text{K}^{-1}$), and T is the temperature

⁸Ultrasound is sound with frequencies greater than 20 kHz.

⁹Of course in this regard careful consideration needs to be made if the medium is significantly different than that of air (i.e., water).

in Kelvin (K). In air, at standard pressure, and 20 °C the speed of sound is approximately:

$$v = 343 \text{ m s}^{-1}.$$

We can see the different signal output and input of an ultrasonic sensor in **Fig. 2.5**.

First, a series of sound pulses are emitted, which creates the wave packet. An integrator also begins to **linearly climb** in value, measuring the time from the transmission of these sound waves to detection of an echo. A threshold value is set for triggering an incoming sound wave as a valid echo.

This threshold is often decreasing in time, because the amplitude of the expected echo decreases over time based on dispersal as it travels longer.

But during transmission of the initial sound pulses and just afterwards, the threshold is set very high to suppress triggering the echo detector with the outgoing sound pulses. A transducer will continue to ring for up to several ms after the initial transmission, and this governs the blanking time of the sensor.

If, during the blanking time, the transmitted sound were to reflect off of an extremely close object and return to the ultrasonic sensor, it may fail to be detected.

However, once the blanking interval has passed, the system will detect any above-threshold reflected sound, triggering a digital signal and producing the distance measurement using the integrator value.

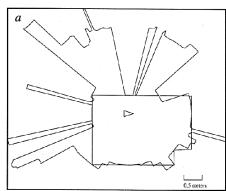
The ultrasonic wave typically has a frequency between 40 and 180 kHz and is usually generated by a piezo or electrostatic transducer. Often the same unit is used to measure the reflected signal, although the required blanking interval can be reduced through the use of separate output and input devices. Frequency can be used to select a useful range when choosing the appropriate ultrasonic sensor for a AMR. Lower frequencies correspond to a longer range, but with the disadvantage of longer post-transmission ringing and, therefore, the need for longer blanking intervals.

Most ultrasonic sensors used by AMRs have an effective range of roughly 12 cm to 5 metres. The published accuracy of commercial ultrasonic sensors varies between 98% and 99.1%. In AMR applications, specific implementations generally achieve a resolution of approximately 2 cm.

In most cases one may want a narrow opening angle for the sound beam in order to also obtain precise directional information about objects that are encountered. This is a major limitation since sound propagates in a cone-like manner with opening angles around 20° and 40°. Consequently, when using ultrasonic ranging one does not acquire depth data points but, rather, entire regions of constant depth. This means that the sensor tells us only that there is an object at a certain distance in within the area of the measurement cone. The sensor readings must be plotted as segments of an arc (sphere for 3D) and not as point measurements.¹⁰ However, recent research developments



Figure 2.6: An example of an ultrasonic sensor used in Raspberry Pi applications [35].



¹⁰The results of a 360° scan of a room.

show significant improvement of the measurement quality in using sophisticated echo processing. Ultrasonic sensors suffer from several additional drawbacks, namely in the areas of **error**, **bandwidth** and **cross-sensitivity**. The published accuracy values for ultrasonic sensors are nominal values based on successful, perpendicular reflections of the sound wave off an acoustically reflective material.

This does not capture the effective error modality seen on a AMR moving through its environment. As the ultrasonic transducer's angle to the object being ranged varies away from perpendicular, the chances become good that the sound waves will coherently reflect away from the sensor, just as light at a shallow angle reflects off of a mirror. Therefore, the true error behavior of ultrasonic sensors is compound, with a well-understood error distribution near the true value in the case of a successful retro-reflection, and a more poorly-understood set of range values that are grossly larger than the true value in the case of coherent reflection.

Of course the acoustic properties of the material being ranged have direct impact on the sensor's performance. Again, the impact is discrete, with one material possibly failing to produce a reflection that is sufficiently strong to be sensed by the unit. For example, foam, fur and cloth can, in various circumstances, acoustically absorb the sound waves. A final limitation for ultrasonic ranging relates to bandwidth. Particularly in moderately open spaces, a single ultrasonic sensor has a relatively slow cycle time.

For example, measuring the distance to an object that is 3 m away will take such a sensor 20ms, limiting its operating speed to 50 Hz. But if the robot has a ring of 20 ultrasonic sensors, each firing sequentially and measuring to minimize interference between the sensors, then the ring's cycle time becomes 0.4s and the overall update frequency of any one sensor is just 2.5 Hz. For a robot conducting moderate speed motion while avoiding obstacles using ultrasonic sensor, this update rate can have a measurable impact on the maximum speed possible while still sensing and avoiding obstacles safely.

Ultrasonic measurements may be limited through barrier layers with large salinity, temperature or vortex differentials.

Laser Rangefinder

The laser rangefinder is a ToF sensor which achieves significant improvements over the ultrasonic range sensor due to the **use of laser light instead of sound**. This type of sensor consists of a transmitter which illuminates a target with a collimated¹¹ beam (e.g. laser), and a receiver capable of detecting the component of light which is essentially coaxial with the transmitted beam. Often referred to as optical radar or Light Detection and Ranging (LIDAR), these devices produce a range estimate based on the time needed for the light to reach the target and return.

¹¹meaning all the rays in questions are made accurately parallel.

A mechanical mechanism with a mirror sweeps the light beam to cover the required scene in a plane or even in 3 dimensions, using a rotating mirror. One way to measure the ToF for the light beam is to use a pulsed laser and then measured the elapsed time directly, just as in the ultrasonic solution

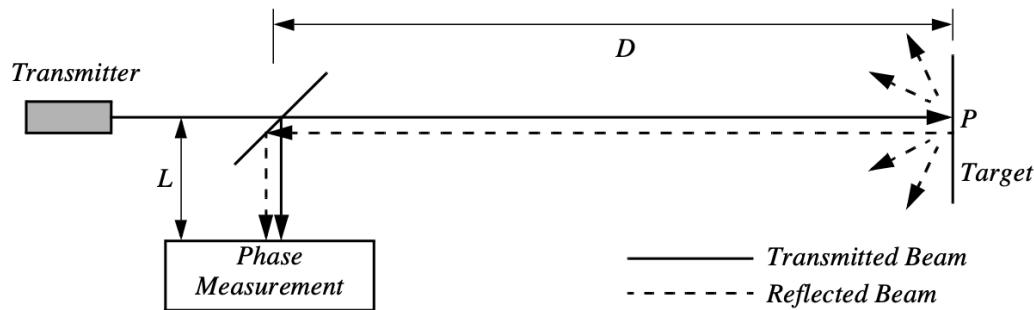


Figure 2.8: Schematic of laser rangefinding by phase-shift measurement.

described in just a little bit. Electronics capable of resolving ps are required in such devices and they are therefore very expensive. A second method is to measure the beat frequency between a frequency modulated continuous wave and its received reflection. Another, even easier method is to measure the phase shift of the reflected light.

Continuous Wave Radar It is a type of radar system where a known stable frequency continuous wave radio energy is transmitted and then received from any reflecting objects. Individual objects can be detected using the Doppler effect, which causes the received signal to have a different frequency from the transmitted signal, allowing it to be detected by filtering out the transmitted frequency.

Doppler-analysis of radar returns can allow the filtering out of slow or non-moving objects, thus offering immunity to interference from large stationary objects and slow-moving clutter. This makes it particularly useful for looking for objects against a background reflector, for instance, allowing a high-flying aircraft to look for aircraft flying at low altitudes against the background of the surface. Because the very strong reflection off the surface can be filtered out, the much smaller reflection from a target can still be seen.



Figure 2.7: A laser range finder used in robotics applications

Phase Shift Measurement Near infrared light, which could be from an Light-Emitting Diode (LED) or a laser, is collimated and transmitted from the transmitter T in Fig. 2.8 and hits a point P in the environment.

For surfaces having a roughness greater than the wavelength of the incident light, diffuse reflection will occur, meaning that the light is reflected almost isotropically¹². The wavelength of the infrared light emitted is 824 nm and so most surfaces with the exception of only highly polished reflecting objects, will be diffuse reflectors. The component of the infrared light which falls within the receiving aperture of the sensor will return almost parallel to the transmitted beam, for distant objects. The sensor transmits 100% amplitude modulated light at a known frequency and measures the phase

¹²Something that is isotropic has the same size or physical properties when it is measured in different directions

shift between the transmitted and reflected signals.

Fig. 2.9 shows how this technique can be used to measure range. The wavelength of the modulating signal obeys the equation $c = f\lambda$ where c is the speed of light and f the modulating frequency.

For example, $f = 5 \text{ MHz}$, the wavelength is $\lambda = 60 \text{ m}$.

The total distance D' covered by the emitted light is:

$$D' = L + 2D = L \frac{\theta}{2\pi} \lambda$$

where D and L are the distances defined in **Fig.** 2.8. The required distance D , between the beam splitter and the target, is therefore given by:

$$D = \frac{\lambda}{4\pi} \theta$$

where θ is the electronically measured phase difference between the transmitted and reflected light beams, and λ the known modulating wavelength. It can be seen that the transmission of a single frequency modulated wave can theoretically result in ambiguous range estimates since

For example if $\lambda = 60\text{m}$, a target at a range of 5 m would give an indistinguishable phase measurement from a target at 65 m , since each phase angle would be 360° apart.

We therefore define an **ambiguity interval** of λ , but in practice we note that the range of the sensor is much lower than λ due to the attenuation of the signal in air. It can be shown that the confidence in the range (phase estimate) is inversely proportional to the square of the received signal amplitude, directly affecting the sensor's accuracy. Hence dark, distant objects will not produce as good range estimates as close, bright objects.

As with ultrasonic ranging sensors, an important error mode involves coherent reflection of the energy. With light, this will only occur when striking a highly polished surface. Practically, a AMR may encounter such surfaces in the form of a polished desktop, file cabinet or of course a mirror. Unlike ultrasonic sensors, laser rangefinders cannot detect the presence of optically transparent materials such as glass, and this can be a significant obstacle in environments, for example museums, where glass is commonly used.

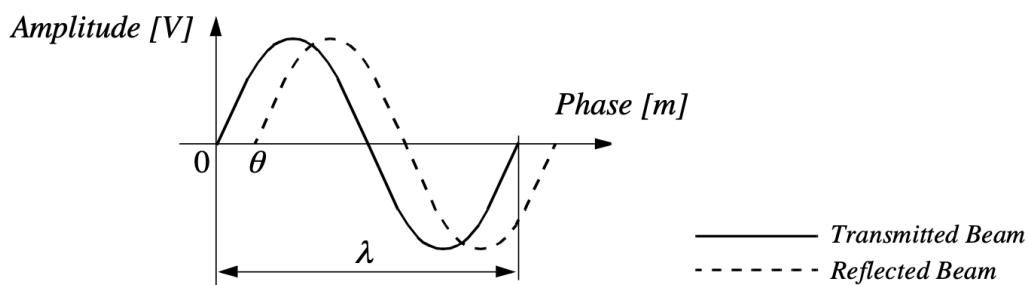


Figure 2.9: Range estimation by measuring the phase shift between transmitted and received signals.

Triangulation-based Active Ranging

Triangulation-based ranging sensors use geometrical properties in their measuring strategy to establish distance readings to objects. The simplest class of triangulation-based rangers are active because they project a known light pattern (e.g., a point, a line or a texture) onto the environment. The reflection of the known pattern is captured by a receiver and, together with known geometric values, the system can use simple triangulation to establish range measurements. If the receiver measures the position of the reflection along a single axis, we call the sensor an optical triangulation sensor in 1D. If the receiver measures the position of the reflection along two orthogonal axes, we call the sensor a structured light sensor.

Optical Triangulation (1D Sensor)

The principle of optical triangulation in 1D is straightforward, as depicted in **Fig. 2.10**. A collimated

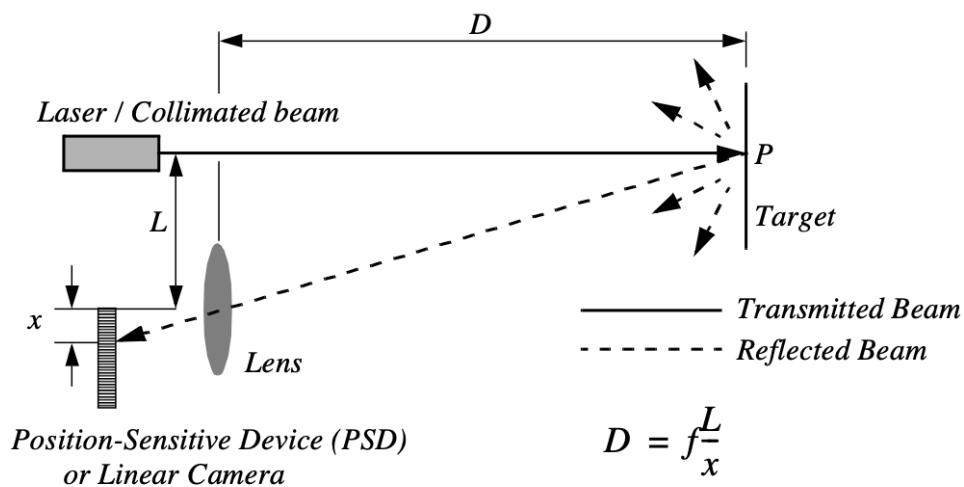


Figure 2.10: Principle of 1D laser triangulation.

beam is transmitted toward the target. The reflected light is collected by a lens and projected onto a position sensitive device¹³ or linear camera. Given the geometry of **Fig. 2.10** the distance D is given by:

$$D = f \frac{L}{x}$$

The distance is proportional to $\frac{1}{x}$, therefore the sensor resolution is best for close objects and becomes worse as distance increases. Sensors based on this principle are used in range sensing up to one or two m, but also in high precision industrial measurements with resolutions far below one μm . Optical triangulation devices can provide relatively high accuracy with very good resolution for close objects. However, the operating range of such a device is normally fairly limited by **geometry**. For



¹³A position sensitive device and/or position sensitive detector is an optical position sensor which can measure a position of a light spot in one or two-dimensions on a sensor surface.

example, an off-the-shelf optical triangulation sensor can operate over a distance range of between 8 cm and 80 cm.

It is inexpensive compared to ultrasonic and laser rangefinder sensors.

Although more limited in range than sonar, the optical triangulation sensor has high bandwidth and does not suffer from cross-sensitivities that are more common in the sound domain.

Structured Light (2D Sensor)

If one replaced the linear camera or Position Sensing Device (PSD) of an optical triangulation sensor with a two-dimensional receiver such as a CCD or CMOS camera, then one can recover distance to a large set of points instead of to only one point. The emitter must project a known pattern, or structured light, onto the environment. Many systems exist which either project light textures, which can be seen in **Fig. 2.12**, or emit collimated light by means of a rotating mirror. Yet another popular alternative is to project a laser stripe by turning a laser beam into a plane using a prism. Regardless of how it is created, the projected light has a known structure, and therefore the image taken by the CCD or CMOS receiver can be filtered to identify the pattern's reflection.

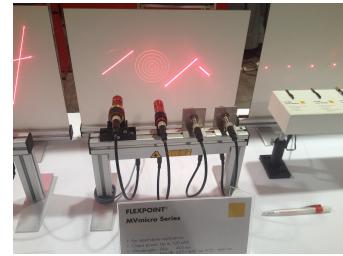


Figure 2.11: Structured light sources on display at the 2014 Machine Vision Show in Boston [36].

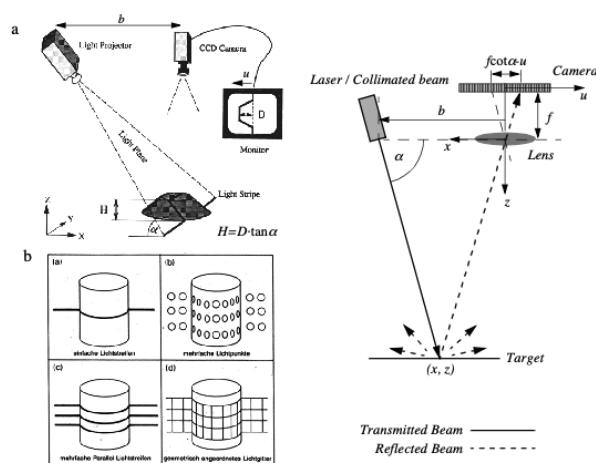


Figure 2.12: a) Principle of active two dimensional triangulation b) Other possible light structures c) One-dimensional schematic of the principle

The problem of recovering depth here far simpler than the problem of passive image analysis.

In passive image analysis, as we discuss later, existing features in the environment must be used to perform correlation, while the present method projects a **known pattern upon the environment** and thereby avoids the standard correlation problem altogether. Furthermore, the structured light sensor

is an active device; so, it will continue to work in dark environments as well as environments in which the objects are featureless¹⁴. In contrast, stereo vision would fail in such texture-free circumstances.

Figure 4.15c shows a one-dimensional active triangulation geometry. We can examine the trade-off in the design of triangulation systems by examining the geometry in figure 4.15c. The measured values in the system are α and u , the distance of the illuminated point from the origin in the imaging sensor.¹⁵ From figure 4.15c, simple geometry shows that:

$$x = \frac{bu}{f \cot \alpha - u} \quad \text{and} \quad z = \frac{bf}{f \cot \alpha - u}.$$

where f is the distance of the lens to the imaging plane. In the limit, the ratio of image resolution to range resolution is defined as the triangulation gain G_p and from equation 4.12 is given by:

$$\frac{\partial u}{\partial z} = G_p = \frac{bf}{z^2}$$

This shows that the ranging accuracy, for a given image resolution, is proportional to source/detector separation b and focal length f , and decreases with the square of the range z . In a scanning ranging system, there is an additional effect on the ranging accuracy, caused by the measurement of the projection angle α . From equation 4.12 we see that:

$$\frac{\partial \alpha}{\partial z} = G_{ff} = \frac{b \sin \alpha^2}{z^2}$$

We can summarise the effects of the parameters on the sensor accuracy as follows:

Baseline Length (b) the smaller b is the more compact the sensor can be. The larger b is the better the range resolution will be. Note also that although these sensors do not suffer from the correspondence problem, the disparity problem still occurs. As the baseline length b is increased, one introduces the chance that, for close objects, the illuminated point(s) may not be in the receiver's field of view.

Detector length and focal length f A larger detector length can provide either a larger field of view or an improved range resolution or partial benefits for both. Increasing the detector length however means a larger sensor head and worse electrical characteristics (increase in random error and reduction of bandwidth). Also, a short focal length gives a large field of view at the expense of accuracy and vice versa.

At one time, laser stripe-based structured light sensors were common on several mobile robot bases as an inexpensive alternative to laser range-finding devices. However, with the increasing quality of laser range-finding sensors in the 1990's the structured light system has become relegated largely to vision research rather than applied mobile robotics.

2.2.2 Motion and Speed Sensors

Some sensors directly measure the relative motion between the robot and its environment. Since such motion sensors detect **relative motion**, so long as an object is moving relative to the robot's

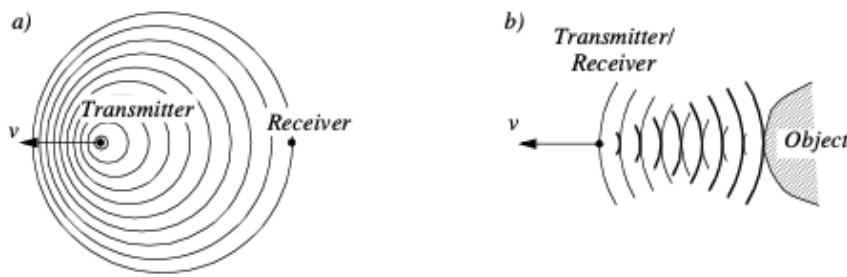


Figure 2.13: Doppler effect between two moving objects (a) or a moving and a stationary object(b)

reference frame, it will be detected and its speed can be estimated. There are a number of sensors that inherently measure some aspect of motion or change.

For example, a pyroelectric¹⁶ sensor detects change in heat.

When someone walks across the sensor's field of view, his motion triggers a change in heat in the sensor's reference frame. In the next subsection, we describe an important type of motion detector based on the **Doppler effect**. These sensors represent a well-known technology with decades of general applications behind them.



¹⁶An example of a pyroelectric sensor.

For fast-moving AMRs such as autonomous highway vehicles and unmanned flying vehicles, Doppler-based motion detectors are the obstacle detection sensor of choice.

Doppler Effect

Anyone who has noticed the change in siren pitch when an ambulance approaches and then passes by is familiar with the Doppler effect.¹⁷

A transmitter emits an electromagnetic or sound wave with a frequency f_t . It is either received by a receiver **Fig. 2.13(a)** or reflected from an object **Fig. 2.13 (b)**. The measured frequency f_r at the receiver is a function of the relative speed v between transmitter and receiver according to

$$f_r = f_t \frac{1}{1 + \frac{v}{c}}$$

if the transmitter is moving and

$$f_r = f_t \left(1 + \frac{v}{c} \right)$$

if the receiver is moving. In the case of a reflected wave **Fig. 2.13 (b)** there is a factor of two introduced, since any change x in relative separation affects the round-trip path length by $2x$.

In such situations it is generally more convenient to consider the change in frequency Δf , known as the Doppler shift, as opposed to the Doppler frequency notation above.

¹⁷For anyone who needs a bit more information, it is the change in the frequency of a wave in relation to an observer who is moving relative to the source of the wave. The Doppler effect is named after the physicist Christian Doppler, who described the phenomenon in 1842. A common example of Doppler shift is the change of pitch heard when a vehicle sounding a horn approaches and recedes from an observer. Compared to the emitted frequency, the received frequency is higher during the approach, identical at the instant of passing by, and lower during the recession.

$$\Delta f = f_t - f_r = \frac{2f_t v \cos \theta}{c} \quad \text{and} \quad v = \frac{\Delta f c}{2f_t \cos \theta}$$

A current application area is both autonomous and manned highway vehicles. Both micro-wave and laser radar systems have been designed for this environment. Both systems have equivalent range, but laser can suffer when visual signals are deteriorated by environmental conditions such as rain, fog, etc. Commercial microwave radar systems are already available for installation on highway trucks. These systems are called VORAD (vehicle on-board radar) and have a total range of approximately 150m. With an accuracy of approximately 97%, these systems report range rate from 0 to 160 km/hr with a resolution of 1 km/ hr. The beam is approximately 4° wide and 5° in elevation. One of the key limitations of radar technology is its bandwidth. Existing systems can provide information on multiple targets at approximately 2 Hz.

2.3 Vision Based Sensors

Vision is our most powerful sense. It provides us with an enormous amount of information about the environment and enables rich, intelligent interaction in dynamic environments. It is therefore not at all surprising that a great deal of effort has been devoted to providing machines with sensors which can at least try to mimic the capabilities of the human vision system.

The first step in this process is the creation of sensing devices that capture the same raw information which is the light the human vision system uses. The main topics which will be described are the two (2) current technologies for creating vision sensors:

1. CCD,
2. CMOS.

Of course, these sensors have specific limitations in performance compared to the human eye, and it is important to understand these limitations. Later sections describe vision-based sensors which are commercially available, similar to the sensors discussed previously, along with their disadvantages and most popular applications.

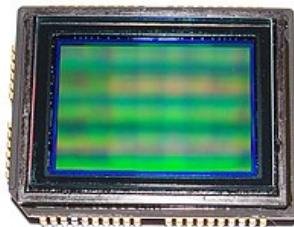


Figure 2.14: Sony ICX493AQ 10.14-megapixel APS-C (23.4 × 15.6 mm) CCD from digital camera Sony DSLR-A200 or DSLR-A300, sensor side [37].

CCD and CMOS Sensors

When it comes to the marketplace, CCD is the most popular fundamental ingredient for robotic vision systems.¹⁸ The CCD chip, which you can see in **Fig. 2.14** is an array of light-sensitive picture elements, or pixels, usually with between 20 000 and 2 million pixels total.

Each pixel can be thought of as a **light-sensitive, discharging capacitor** that is 5 to 25 µm in size. First, the capacitors of all pixels are fully charged, then the integration period begins. As photons of light strike each pixel, the electrons are liberated, which are captured by electric fields and retained at the pixel. Over time, each pixel accumulates a varying level of charge based on the total number of photons that have struck it. After the integration period is complete, the relative charges of all pixels need to be **frozen and read**.

In a CCD, the reading process is performed at one corner of the CCD chip.¹⁹ The bottom row of pixel charges are transported to this corner and read, then the rows above shift down and the process repeats. This means that each charge **must be transported across the chip**, and it is critical the value be preserved.

This requires specialised control circuitry and custom fabrication techniques to ensure the stability of transported charges.

¹⁸Willard Boyle and George E. Smith invented the CCD in 1969 at AT&T Bell Labs. Their original idea was to create a memory device. However, with its publication in 1970, other scientists began experimenting with the technology on a range of applications. Astronomers discovered that they could produce high-resolution images of distant objects, because CCDs offered a photo-sensitivity one hundred times greater than film [38].

¹⁹Because the entire array is read through a single amplifier the output can be highly optimised to give very low noise and extremely high dynamic range. CCDs can have over 100 dB dynamic range with less than 2e of noise [38].

²⁰This also includes CMOS as well.

The photo-diodes used in CCD chips²⁰ are **NOT** equally sensitive to all frequencies of light. They are sensitive to light between 400 nm and 1000 nm wavelength.²¹

²¹This number range is usually given for easier numbers as both CCD and CMOS have sensitivity values at approximately 350 - 1050 nm.

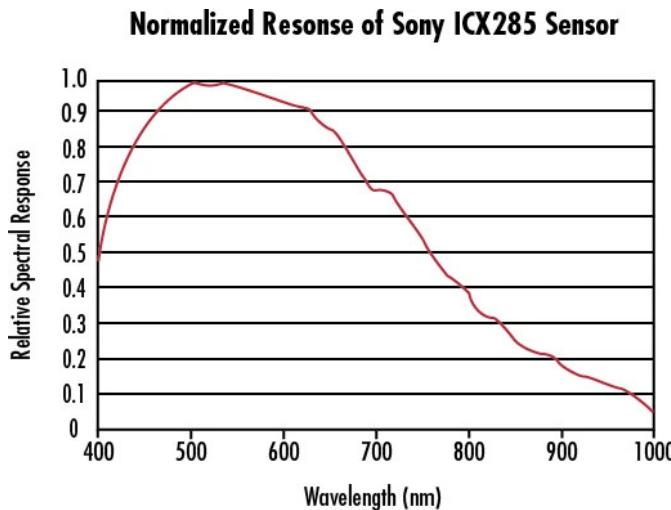


Figure 2.15: Normalized Spectral Response of a Typical Monochrome CCD.

It is important to remember that photodiodes are **less sensitive to the ultraviolet** part of the spectrum and are overly **sensitive to the infrared** portion (e.g. heat) which you can see in Fig. 2.15. You can see that the basic light-measuring process is colourless.²²

There are two (2) common approaches for creating color images. If the pixels on the CCD chip are grouped into 2-by-2 sets of four (4), then red, green and blue dyes can be applied to a colour filter so each individual pixel receives only light of just one color.

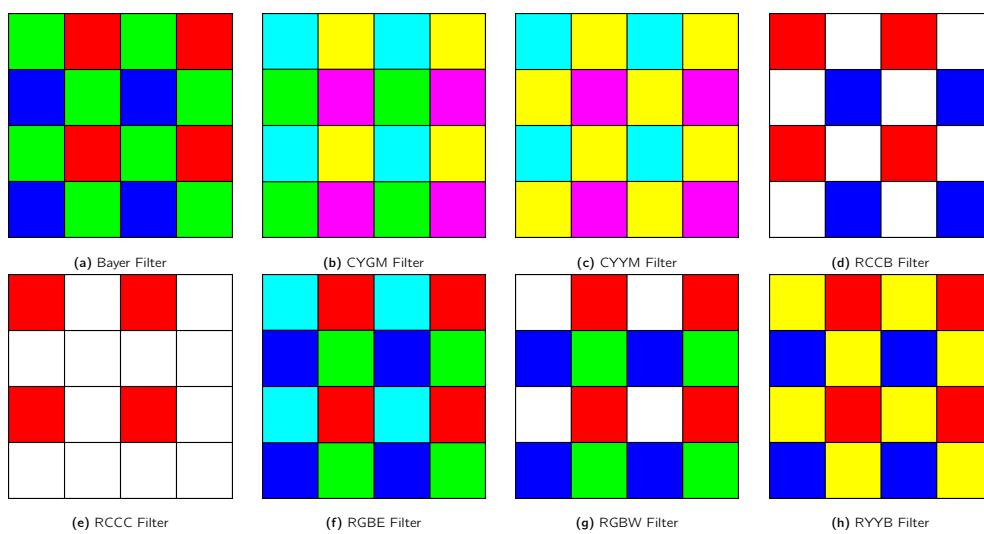


Figure 2.16: Types of colour filter used in commercial and industrial applications

Normally, two (2) pixels measure green while one pixel each measures red and blue light intensity. Of course, this 1-chip color CCD has a geometric resolution disadvantage.

The number of pixels in the system has been effectively cut by a factor of 4, and therefore the image resolution output by the CCD camera will be sacrificed.

The 3-chip color camera avoids these problems by splitting the incoming light into three (3) complete²³ copies. Three separate CCD chips receive the light, with one red, green or blue filter over each entire chip. Thus, in parallel, each chip measures light intensity for just one color, and the camera must combine the CCD chips' outputs to create a joint color image.

²³Albeit, with lower resolution.

Resolution is preserved in this solution, although the 3-chip color cameras are, as one would expect, significantly more expensive and therefore more rarely used in mobile robotics.

Both 3-chip and single chip color CCD cameras suffer from the fact that photo-diodes are much more sensitive to the near-infrared end of the spectrum. This means that the overall system detects blue light much more poorly than red and green. To compensate, the gain must be increased on the blue channel, and this introduces greater absolute noise on blue²⁴ than on red and green. It is not uncommon to assume at least 1 - 2 bits of additional noise on the blue channel.

²⁴This is generally defined as the amplifier noise.

The CCD camera has several camera parameters that affect its behavior. In some cameras, these parameter values are fixed. In others, the values are constantly changing based on built-in feedback loops. In higher-end cameras, the user can modify the values of these parameters via software embedded into the device. The iris position and shutter speed²⁵ regulate the amount of light being measured by the camera. The iris is simply a mechanical aperture that constricts incoming light, just as in standard 35mm cameras. Shutter speed regulates the integration period of the chip. In higher-end cameras, the effective shutter speed can be as brief at 1/30,000s and as long as 2s. Camera gain controls the overall amplification of the analog signal, prior to A/D conversion. However, it is very important to understand that, even though the image may appear brighter after setting high gain, the shutter speed and iris may not have changed at all. Thus gain merely amplifies the signal, and amplifies along with the signal all of the associated noise and error. Although useful in applications where imaging is done for human consumption (e.g. photography, television), gain is of little value to a mobile roboticist.

²⁵It's the speed at which the shutter of the camera closes. A fast shutter speed creates a shorter exposure - the amount of light the camera takes in - and a slow shutter speed gives a longer exposure.

In colour cameras, an additional control exists for white balance. Depending on the source of illumination in a scene²⁶ the relative measurements of red, green and blue light which combine to define pure white light will change dramatically which can be seen in **Fig. 2.17** which can also be adjusted with algorithms [39]. The human eyes compensate for all such effects in ways that are not fully understood, however, the camera can demonstrate glaring inconsistencies in which the same table looks blue in one image, taken during the night, and yellow in another image, taken during the day. White balance controls enable the user to change the relative gain for red, green and blue in order to maintain more consistent color definitions in varying contexts.

²⁶For example this could be fluorescent lamps, incandescent lamps, sunlight, underwater filtered light, etc.

The key disadvantages of CCD cameras are primarily in the areas of inconstancy and **dynamic range**.



Figure 2.17: Example of white balance. Here the same scene is emulated to be shot under different light conditions [40].

Information: Dynamic Range

Dynamic range in photography describes the ratio between the maximum and minimum measurable light intensities (white and black, respectively). In the real world, one never encounters true white or black - only varying degrees of light source intensity and subject reflectivity. Therefore the concept of dynamic range becomes more complicated, and depends on whether you are describing a capture device (such as a camera or scanner), a display device (such as a print or computer display), or the subject itself.

As mentioned above, a number of parameters can change the brightness and colours with which a camera creates its image.

Manipulating these parameters in a way to provide consistency over time and over environments, for example ensuring a green shirt always looks green, and something dark grey is always dark grey, remains an open problem [41].

The second type of disadvantages relates to the behavior of a CCD chip in environments with **extreme illumination**. In cases of very low illumination, each pixel will receive only a small number of photons. The longest possible shutter speed and camera optics (i.e. pixel size, chip size, lens focal length and diameter) will determine the minimum level of light for which the signal is stronger than random error noise. In cases of very high illumination, a pixel fills its well with free electrons and, as the well reaches its limit, the probability of trapping additional electrons falls and therefore the linearity between incoming light and electrons in the well degrades. This is termed **saturation**²⁷ and can indicate the existence of a further problem related to cross-sensitivity [43]. When a well has reached its limit, then additional light within the remainder of the integration period may cause further charge to leak into neighbouring pixels, causing them to report incorrect values or even reach secondary saturation. This effect, called **blooming**, means that individual pixel values are **NOT** truly **independent**. The camera parameters may be adjusted for an environment with a particular light level, but the problem remains that the dynamic range of a camera is limited by the well capacity of the individual pixels.



²⁷Example of blooming caused by saturation of a sensor pixel. The sun is so bright in the image that there is blooming on the sun itself, leaking into the surrounding pixels, and a vertical smear across the whole image [42].

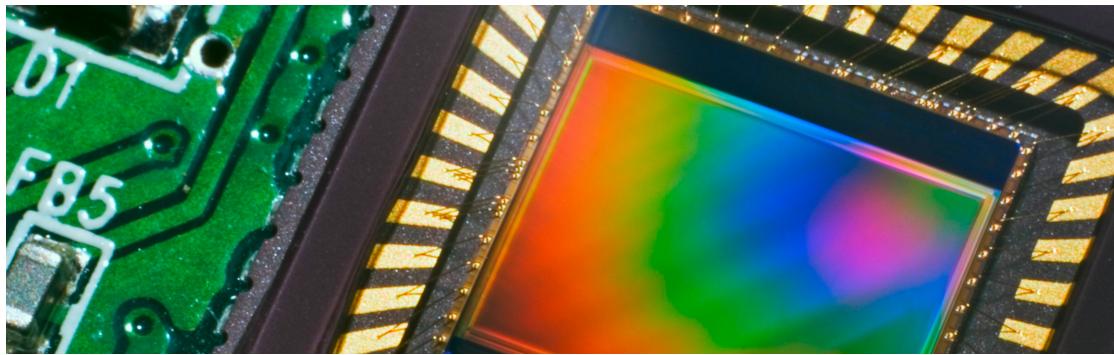


Figure 2.18: A close-up view of a CMOS sensor and its circuitry [44].

For example, a high quality CCD may have pixels that can hold 40 000 electrons. The noise level for reading the well may be 11 electrons, and therefore the dynamic range will be 40,000:11, or 3,600:1, which is 35 dB.

2.3.1 CMOS Technology

The Complementary Metal Oxide Semiconductor (CMOS) chip is a significant departure from the CCD. Similar to CCD, it too has an array of pixels, but located alongside each pixel are **several transistors specific to that pixel**. Just as in CCD chips, all of the pixels accumulate charge during the integration period. During the data collection step, the CMOS takes a new approach:

The pixel-specific circuitry next to every pixel measures and amplifies the pixel's signal, all in parallel for every pixel in the array.

Using more traditional traces from general semiconductor chips, the resulting pixel values are all carried to their destinations. CMOS has a number of advantages over CCD technologies. First and foremost, there is no need for the specialized clock drivers and circuitry required in the CCD to transfer each pixel's clock down all of the array columns and across all of its rows.²⁸

This also means that specialized semiconductor manufacturing processes are not required to create CMOS chips.

Therefore, the same production lines that create microchips can create inexpensive CMOS chips as well. The CMOS chip is so much simpler that it consumes significantly less power, it operates with a power consumption a tenth the power consumption of a CCD chip [46].

In a AMR, power is a scarce resource and therefore this is an important advantage.

On the other hand, the CMOS chip also faces several disadvantages.

- Most importantly, the circuitry next to each pixel consumes valuable real estate on the face of the light-detecting array. Many photons hit the transistors rather than the photodiode, making



²⁸-CAM80CUNX is an 8MP Ultra-lowlight MIPI CSI-2 camera capable of streaming 4K @ 44 fps. This 8MP camera is based on SONY STARVIS IMX415 CMOS image sensor [45]

the CMOS chip significantly less sensitive than an equivalent CCD chip.

- CMOS, compared to CCD is still finding ground in the marketplace, and as a result, the best resolution that one can purchase in CMOS format continues to be far inferior to the best CCD chips available.
- CMOS sensors have a lower dynamic range,
- CMOS sensors have higher levels of noise.

Compared to the human eye, these chips all have worse performance, cross-sensitivity and a limited dynamic range. As a result, vision sensors today continue to be fragile. Only over time, as the underlying performance of imaging chips improves, will significantly more robust vision-based sensors for AMRs be available.

Information: Shot Noise

Shot noise or Poisson noise is a type of noise which can be modeled by a Poisson process. In electronics shot noise originates from the discrete nature of electric charge. Shot noise also occurs in photon counting in optical devices, where shot noise is associated with the particle nature of light.

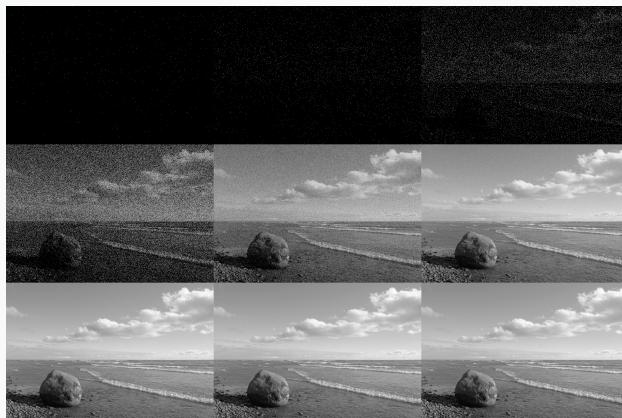


Figure 2.19: Photon noise simulation. Number of photons per pixel increases from left to right and from upper row to bottom row [47].

2.3.2 Visual Ranging Sensors

Range sensing is extremely important in AMR as it is a basic input for successful obstacle avoidance. As we have seen earlier, a number of sensors are popular in robotics specifically for their ability to recover depth estimates:

ultrasonic, laser rangefinder, optical rangefinder, etc.

It is natural to attempt to implement ranging functionality using vision chips as well. However, a fundamental problem with visual images makes rangefinding relatively difficult.

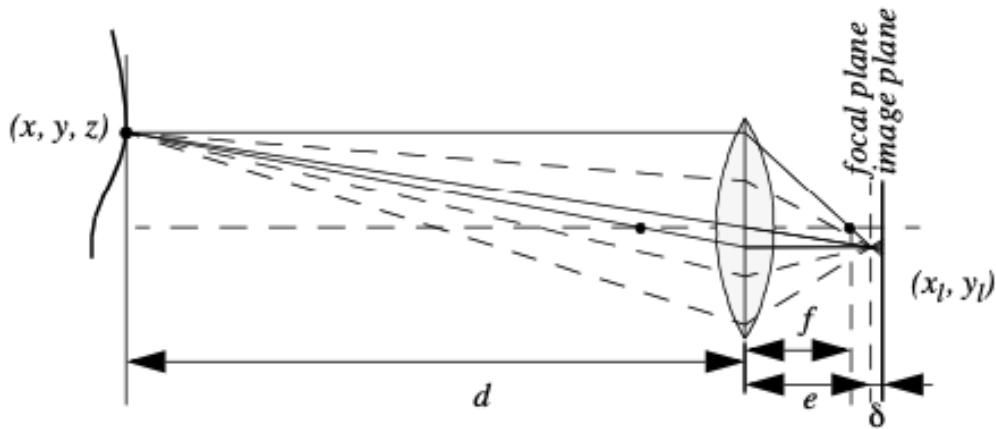


Figure 2.20: Depiction of the camera optics and its impact on the image. To get a sharp image, the image plane must coincide with the focal plane. Otherwise the image of the point (x, y, z) will be blurred in the image as can be seen in the drawing above.

Any vision chip collapses the three-dimensional world into a two-dimensional image plane, thereby losing depth information. If one can make strong assumptions regarding the size of objects in the world, or their particular colour and reflectance, then one can directly interpret the appearance of the two-dimensional image to recover depth. But such assumptions are rarely possible in real-world AMR applications.

Without such assumptions, a single picture does not provide enough information to recover spatial information.

The general solution is to recover depth by looking at several images of the scene to gain more information, which will be hopefully enough to at least partially recover depth. The images used **must be different**, so that taken together they provide additional information. They could differ in viewpoint, which would allow the use of stereo or motion algorithms.

An alternative is to create different images, not by changing the viewpoint, but by changing the camera geometry, such as the focus position or lens iris. This is the fundamental idea behind depth from focus and depth from defocus techniques. We will now look into the general approach to the depth from focus techniques as it presents a straightforward and efficient way to create a vision-based range sensor.

2.3.3 Depth from Focus

The depth from focus class of techniques relies on the fact that image properties not only change as a function of the **scene**, but also as a function of the **camera parameters**. The relationship between camera parameters and image properties is depicted in **Fig. 2.20**. The fundamental formula governing image formation relates the distance of the object from the lens, **d** in **Fig. 2.20**, to the

distance e from the lens to the focal point, based on the focal length f of the lens:

$$\frac{1}{f} = \frac{1}{d} + \frac{1}{e}$$

²⁹A three-dimensional counterpart to a pixel.
If the image plane is located at distance e from the lens, then for the specific object voxel²⁹ depicted, all light will be focused at a single point on the image plane and the object voxel will be focused. However, when the image plane is **NOT** at e , as is seen in **Fig. 2.20**, then the light from the object voxel will be cast on the image plane as a **blur circle**. To a first approximation, the light is homogeneously distributed throughout this blur circle, and the radius R of the circle can be characterized according to the equation:

$$R = \frac{L\delta}{2e}$$

where L is the diameter of the lens or aperture and δ is the displacement of the image plan from the focal point.

Given these formulae, several basic optical effects are clear.

³⁰The aperture is the opening in the lens that allows light to enter the camera and onto the sensor or film.

For example, if the aperture³⁰ or lens is reduced to a point, as in a pin-hole camera, then the radius of the blur circle approaches zero.

This is consistent with the fact that decreasing the iris aperture opening causes the depth of field to increase until all objects are in focus. Of course, the disadvantage of doing so is that we are allowing less light to form the image on the image plane and so this is practical only in bright circumstances. The second property to be deduced from these optics equations relates to the sensitivity of blurring as a function of the distance from the lens to the object.

Suppose the image plane is at a fixed distance 1.2 from a lens with diameter $L = 0.2$ and focal length $f = 0.5$. We can see from Equation (4.20) that the size of the blur circle R changes proportionally with the image plane displacement . If the object is at distance $d = 1$, then from Equation (4.19) we can compute $e=1$ and therefore $\delta = 0.2$. Increase the object distance to $d = 2$ and as a result $\delta = 0.533$. Using Equation (4.20) in each case we can compute $R = 0.02$ $R = 0.08$ respectively. This demonstrates high sensitivity for defocusing when the object is close to the lens. In contrast suppose the object is at $d = 10$. In this case we compute $e = 0.526$. But if the object is again moved one unit, to $d = 11$, then we compute $e = 0.524$. Then resulting blur circles are $R = 0.117$ and $R =$



Figure 2.21: Three images of the same scene taken with a camera at three different focusing positions. Note the significant change in texture sharpness between the near surface and far surface [48].

0.129, far less than the quadrupling in R when the obstacle is 1/10 the distance from the lens. This analysis demonstrates the fundamental limitation of depth from focus techniques: they lose sensitivity as objects move further away (given a fixed focal length). Interestingly, this limitation will turn out to apply to virtually all visual ranging techniques, including depth from stereo and depth from motion. Nevertheless, camera optics can be customised for the depth range of the intended application. For example, a "zoom" lens with a very large focal length f will enable range resolution at significant distances, of course at the expense of field of view. Similarly, a large lens diameter, coupled with a very fast shutter speed, will lead to larger, more detectable blur circles. Given the physical effects summarised by the above equations, one can imagine a visual ranging sensor that makes use of multiple images in which camera optics are varied (e.g. image plane displacement) and the same scene is captured (see Fig. 4.20). In fact this approach is not a new invention. The human visual system uses an abundance of cues and techniques, and one system demonstrated in humans is depth from focus. Humans vary the focal length of their lens continuously at a rate of about 2 Hz. Such approaches, in which the lens optics are actively searched in order to maximise focus, are technically called depth from focus. In contrast, depth from defocus means that depth is recovered using a series of images that have been taken with different camera geometries. Depth from focus methods are one of the simplest visual ranging techniques. To determine the range to an object, the sensor simply moves the image plane (via focusing) until maximizing the sharpness of the object. When the sharpness is maximised, the corresponding position of the image plane directly reports range. Some autofocus cameras and virtually all autofocus video cameras use this technique. Of course, a method is required for measuring the sharpness of an image or an object within the image. The most common techniques are approximate measurements of the sub-image gradient:

$$\text{sharpness}_1 = \sum_{x,y} |I(x, y) - I(x-1, y)| \quad (2.3)$$

$$\text{sharpness}_2 = \sum_{x,y} (I(x, y) - I(x-2, y-2))^2 \quad (2.4)$$

A significant advantage of the horizontal sum of differences technique (Equation (4.21)) is that the calculation can be implemented in analog circuitry using just a rectifier, a low-pass filter and a high-pass filter. This is a common approach in commercial cameras and video recorders. Such systems will be sensitive to contrast along one particular axis, although in practical terms this is rarely an issue. However depth from focus is an active search method and will be slow because it takes time to change the focusing parameters of the camera, using for example a servo-controlled focusing ring. For this reason this method has not been applied to AMRs. A variation of the depth from focus technique has been applied to a AMR, demonstrating obstacle avoidance in a variety of environments as well as avoidance of concave obstacles such as steps and ledges [95]. This robot uses three monochrome cameras placed as close together as possible with different, fixed lens focus positions (Fig. 4.21).

Several times each second, all three frame-synchronised cameras simultaneously capture three images of the same scene. The images are each divided into five columns and three rows, or 15 subregions. The approximate sharpness of each region is computed using a variation of Equation (4.22), leading to a total of 45 sharpness values. Note that Equation 22 calculates sharpness along diagonals but skips one row. This is due to a subtle but important issue. Many cameras produce images in

interlaced mode. This means that the odd rows are captured first, then afterwards the even rows are captured. When such a camera is used in dynamic environments, for example on a moving robot, then adjacent rows show the dynamic scene at two different time points, differing by up to 1/30 seconds. The result is an artificial blurring due to motion and not optical defocus. By comparing only even-number rows we avoid this interlacing side effect.

Recall that the three images are each taken with a camera using a different focus position. Based on the focusing position, we call each image close, medium or far. A 5x3 coarse depth map of the scene is constructed quickly by simply comparing the sharpness values of each three corresponding regions. Thus, the depth map assigns only two bits of depth information to each region using the values close, medium and far. The critical step is to adjust the focus positions of all three cameras so that flat ground in front of the obstacle results in medium readings in one row of the depth map. Then, unexpected readings of either close or far will indicate convex and concave obstacles respectively, enabling basic obstacle avoidance in the vicinity of objects on the ground as well as drop-offs into the ground. Although sufficient for obstacle avoidance, the above depth from focus algorithm presents unsatisfyingly coarse range information. The alternative is depth from defocus, the most desirable of the focus-based vision techniques. Depth from defocus methods take as input two or more images of the same scene, taken with different, known camera geometry. Given the images and the camera geometry settings, the goal is to recover the depth information of the three-dimensional scene represented by the images. We begin by deriving the relationship between the actual scene properties (irradiance and depth), camera geometry settings and the image g that is formed at the image plane. The focused image $f(x,y)$ of a scene is defined as follows. Consider a pinhole aperture ($L=0$) in lieu of the lens. For every point p at position (x,y) on the image plane, draw a line through the pinhole aperture to the corresponding, visible point P in the actual scene. We define $f(x,y)$ as the irradiance (or light intensity) at p due to the light from P . Intuitively, $f(x,y)$ represents the intensity image of the scene perfectly in focus

2.4 Feature Extraction

An AMR must be able to determine its relationship to the environment by making measurements with its sensors and then using those measured signals. A wide variety of sensing technologies are available, as we discussed previously. But every sensor we have presented is imperfect:

measurements always have error and, therefore, uncertainty associated with them.

Therefore, sensor inputs must be used in a way that enables the robot to interact with its environment successfully in spite of measurement uncertainty. There are two (2) strategies for using uncertain sensor input to guide the robot's behavior. One strategy is to use each sensor measurement as a raw and individual value. Such raw sensor values could for example be tied directly to robot behavior, whereby the robot's actions are a function of its sensor inputs. Alternatively, the raw sensors values could be used to update an intermediate model, with the robot's actions being triggered as a function of this model rather than the individual sensor measurements.

The second strategy is to extract information from one or more sensor readings first, generating a higher-level percept that can then be used to inform the robot's model and perhaps the robot's actions directly. We call this process feature extraction, and it is this next, optional step in the perceptual interpretation pipeline (Fig. 4.34) that we will now discuss.

In practical terms, mobile robots do not necessarily use feature extraction and scene interpretation for every activity. Instead, robots will interpret sensors to varying degrees depending on each specific functionality. For example, in order to guarantee emergency stops in the face of immediate obstacles, the robot may make direct use of raw forward-facing range readings to stop its drive motors. For local obstacle avoidance, raw ranging sensor strikes may be combined in an occupancy grid model, enabling smooth avoidance of obstacles meters away. For map-building and precise navigation, the range sensor values and even vision sensor measurements may pass through the complete perceptual pipeline, being subjected to feature extraction followed by scene interpretation to minimize the impact of individual sensor uncertainty on the robustness of the robot's map-making and navigation skills. The pattern that thus emerges is that, as one moves into more sophisticated, long-term perceptual tasks, the feature extraction and scene interpretation aspects of the perceptual pipeline become essential.

2.4.1 Defining Feature

Features are recognizable structures of elements in the environment. They usually can be extracted from measurements and mathematically described. Good features are always perceivable and easily detectable from the environment. We distinguish between low-level features (geometric primitives) like lines, circles or polygons and high-level features (objects) such as edges, doors, tables or a trash can. At one extreme, raw sensor data provides a large volume of data, but with low distinctiveness of each individual quantum of data. Making use of raw data has the potential advantage that every bit of information is fully used, and thus there is a high conservation of information. Low level

features are abstractions of raw data, and as such provide a lower volume of data while increasing the distinctiveness of each feature. The hope, when one incorporates low level features, is that the features are filtering out poor or useless data, but of course it is also likely that some valid information will be lost as a result of the feature extraction process. High level features provide maximum abstraction from the raw data, thereby reducing the volume of data as much as possible while providing highly distinctive resulting features. Once again, the abstraction process has the risk of filtering away important information, potentially lowering data utilization.

Although features must have some spatial locality, their geometric extent can range widely. For example, a corner feature inhabits a specific coordinate location in the geometric world. In contrast, a visual "fingerprint" identifying a specific room in an office building applies to the entire room, but has a location that is spatially limited to the one, particular room. In mobile robotics, features play an especially important role in the creation of environmental models. They enable more compact and robust descriptions of the environment, helping a mobile robot during both map-building and localization. When designing a mobile robot, a critical decision revolves around choosing the appropriate features for the robot to use. A number of factors are essential to this decision:

Target Environment For geometric features to be useful, the target geometries must be readily detected in the actual environment. For example, line features are extremely useful in office building environments due to the abundance of straight walls segments while the same feature is virtually useless when navigating Mars.

Available Sensors Obviously the specific sensors and sensor uncertainty of the robot impacts the appropriateness of various features. Armed with a laser rangefinder, a robot is well qualified to use geometrically detailed features such as corner features due to the high quality angular and depth resolution of the laser scanner. In contrast, a sonar-equipped robot may not have the appropriate tools for corner feature extraction.

Computational Power Vision-based feature extraction can effect a significant computational cost, particularly in robots where the vision sensor processing is performed by one of the robot's main processors.

Environment representation Feature extraction is an important step toward scene interpretation, and by this token the features extracted must provide information that is consonant with the representation used for the environment model. For example, non-geometric vision-based features are of little value in purely geometric environment models but can be of great value in topological models of the environment. Figure 4.35 shows the application of two different representations to the task of modeling an office building hallway. Each approach has advantages and disadvantages, but extraction of line and corner features has much more relevance to the representation on the left. Refer to Chapter 5, Section 5.5 for a close look at map representations and their relative tradeoffs. In the following two sections, we present specific feature extraction techniques based on the two most popular sensing modalities of mobile robotics: range sensing and visual appearance-based sensing.

2.4.2 Using Range Data

Most of today's features extracted from ranging sensors are geometric primitives such as line segments or circles. The main reason for this is that for most other geometric primitives the parametric description of the features becomes too complex and no closed form solution exists. Here we will describe line extraction in detail, demonstrating how the uncertainty models presented above can be applied to the problem of combining multiple sensor measurements. Afterwards, we briefly present another very successful feature for indoor mobile robots, the corner feature, and demonstrate how these features can be combined in a single representation.

Line Extraction

Geometric feature extraction is usually the process of comparing and matching measured sensor data against a predefined description, or template, of the expected feature. Usually, the system is overdetermined in that the number of sensor measurements exceeds the number of feature parameters to be estimated. Since the sensor measurements all have some error, there is no perfectly consistent solution and, instead, the problem is one of optimization. One can, for example, extract the feature that minimizes the discrepancy with all sensor measurements used (e.g. least squares estimation). In this section we present an optimization-based solution to the problem of extracting a line feature from a set of uncertain sensor measurements. For greater detail than is presented below, refer to [19], pp. 15 and 221.

Probabilistic Line Extraction

4.36. There is uncertainty associated with each of the noisy range sensor measurements, and so there is no single line that passes through the set. Instead, we wish to select the best possible match, given some optimization criterion. More formally, suppose n ranging measurement points in polar coordinates $x = (\rho, \theta)$ are produced by the robot's sensors. We know that there is uncertainty associated with each measurement, and so we can model each measurement using two random variables $X = (P, Q)$. In this analysis we assume that uncertainty with respect to the actual value θ of P and Q are independent. Based on Equation (4.56) we can state this formally: Furthermore, we will assume that each random variable is subject to a Gaussian probability density curve, with a mean at the true value and with some specified variance: Given some measurement point (ρ, θ) , we can calculate the corresponding Euclidean coordinates $x = (\cos \theta, \sin \theta)$. If there were no error, we would want to find a line for which all measurements lie on that line: Of course there is measurement error, and so this quantity will not be zero. When it is non-zero, this is a measure of the error between the measurement point (ρ, θ) and the line, specifically in terms of the minimum orthogonal distance between the point and the line. It is always important to understand how the error that shall be minimized is being measured. For example a number of line extraction techniques do not minimize this orthogonal point-line distance, but instead the distance parallel to the y -axis between the point and the line. A good illustration of the variety of

optimization criteria is available in [18] where several algorithms for fitting circles and ellipses are presented which minimize algebraic and geo-metric distances. For each specific (x_i, y_i) , we can write the orthogonal distance d between (x_i, y_i) and ℓ the line as:

Part II

Probability and Statistics

Chapter 3

Theory of Probability

Table of Contents

3.1	Introduction	67
3.2	Experiments & Outcomes	71
3.3	Probability	72
3.4	Permutations & Combinations	77
3.5	Random Variables and Probability Distributions	81
3.6	Mean and Variance of a Distribution	85
3.7	Binomial, Poisson, and Hyper-geometric Distributions	88
3.8	Normal Distribution	93
3.9	Distribution of Several Random Variables	96

3.1 Introduction

When the data we are working are influenced by “**chance**”, by factors whose effect we cannot predict exactly¹, we have to rely on **probability theory**. The application of this theory nowadays appears in numerous fields such as from studying a game of cards to the global financial market and allow us to model processes of chance called **random experiments**.

¹This could be weather data, stock prices, life spans or ties, etc.

In such an experiment we observe a **random variable** X , that is, a function whose values in a trial² occur “by chance” according to a **probability distribution** which gives the individual probabilities, which possible values of X may occur in the long run.

²a performance of an experiment.

i.e., each of the six faces of a die should occur with the same probability, $1/6$.

Or we may simultaneously observe more than one random variable, for instance, height and weight of persons or hardness and tensile strength of steel. But enough about spoiling all the fun and let's

begin with looking at data.

Representing Data

Data can be represented numerically or graphically in different ways

i.e., a news website may contain tables of stock prices and currency exchange rates, curves or bar charts illustrating economical or political developments, or pie charts showing how inflation is calculated.

And there are numerous other representations of data for special purposes. In this section, we will discuss the use of standard representations of data in statistics³.

³There are various software dedicated to analyse and visualise statistical data. Some of these include: R, a statistical programming language, Python, MATLAB, ...

Exercise 3.1: Recording Data

Sample values, such as observations and measurements, should be recorded in the order in which they occur. Sorting, that is, ordering the sample values by size, is done as a first step of investigating properties of the sample and graphing it.

As an example let's look at super alloys.

Super alloys is a collective name for alloys used in jet engines and rocket motors, requiring high temperature (typically 1000° C), high strength, and excellent resistance to oxidation.

Thirty (30) specimens of Hastelloy C (nickel-based steel, investment cast) had the tensile strength (in 1000 lb>sq in.), recorded in the order obtained and rounded to integer values.

$$\begin{array}{cccccccccccccccccccc} 89 & 77 & 88 & 91 & 88 & 93 & 99 & 79 & 87 & 84 & 86 & 82 & 88 & 89 & 78 \\ 90 & 91 & 81 & 90 & 83 & 83 & 92 & 87 & 89 & 86 & 89 & 81 & 87 & 84 & 89 \end{array} \quad (3.1)$$

Of course depending on the need the data needs to be sorted which is shown below:

$$\begin{array}{cccccccccccccccccccc} 77 & 78 & 79 & 81 & 81 & 82 & 83 & 83 & 84 & 84 & 86 & 86 & 87 & 87 & 87 \\ 88 & 88 & 88 & 89 & 89 & 89 & 89 & 89 & 90 & 90 & 91 & 91 & 92 & 93 & 99 \end{array}$$

Graphic Representation of Data

Let's now use the data we have seen in Example 1 and see the methods we can use for graphic representations.

Exercise 3.2: Leaf Plots

One of the simplest yet most useful representations of data [49]. For Eq. (3.1) it is shown in Table ??.

The numbers in Eq. (3.1) range from 78 to 99; which you can also see this in the sorted list. To visualise this data feature, we divide these numbers into five (5) groups:

75-79, 80-84, 85-89, 90-94, 95-99.

The integers in the tens position of the groups are 7, 8, 8,

9, 9. These form the stem which can be seen in Table ??.
The first leaf is 789, representing 77, 78, 79. The second leaf is 1123344, representing 81, 81, 82, 83, 83, 84, 84. And so on. The number of times a value occurs is called its **absolute frequency**.

Therefore in this example, 78 has absolute frequency 1, the value 89 has absolute frequency 5, etc. ■

Exercise 3.3: Histogram

For large sets of data, histograms are better in displaying the distribution of data than stem-and-leaf plots. The principle is explained in Fig. 3.1.

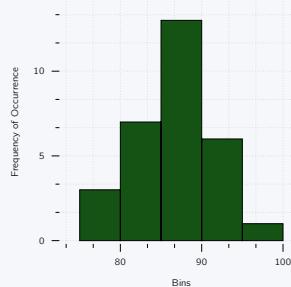


Figure 3.1: The histogram of the data given in Exercise 1.

The bases of the rectangles in seen in Fig. 3.1 are the x-intervals⁴ where there rage is:

$$74.5 - 79.5, \quad 79.5 - 84.5, \quad 84.5 - 89.5, \\ 89.5 - 94.5, \quad 94.5 - 99.5,$$

whose midpoints, known as **class marks**, are

$$x = 77, 82, 87, 92, 97,$$

respectively. The height of a rectangle with class mark x is the relative class frequency $f_{\text{rel}}(x)$, defined as the number of data values in that class interval, divided by n ($= 30$ in our case). Hence the areas of the rectangles are proportional to these relative frequencies,

$$0.10, 0.23, 0.43, 0.17, 0.07,$$

so that histograms give a good impression of the distribution of data.

⁴known as class intervals.

Mean, Standard Deviation, and Variance

Medians and quartiles are easily obtained by ordering and counting⁵.

⁵This can be done without the need of calculators.

However this method does not give full information on data as you can change data values to some extent without changing the median.

The average size of the data values can be measured in a more refined way by the mean:

$$\bar{x} = \frac{1}{n} \sum_{j=1}^n x_j = \frac{1}{n} (x_1 + x_2 + \dots + x_n). \quad (3.2)$$

This is the **arithmetic mean** of the data values, obtained by taking their sum and dividing by the data size (n). Therefore the arithmetic mean for Eq. (3.1) is:

$$\bar{x} = \frac{1}{30} (89 + 77 + \dots + 89) = \frac{260}{3} \approx 86.7 \quad \blacksquare$$

As we can see every data value contributes, and changing one of them will change the mean. Similarly, the spread⁶ of the data values can be measured in a more refined way by the **standard deviation** s or by its square, the **variance**⁷

$$s^2 = \frac{1}{n-1} \sum_{j=1}^n (x_j - \bar{x})^2 = \frac{1}{n-1} [(x_1 - \bar{x})^2 + \dots + (x_n - \bar{x})^2] \quad (3.3)$$

⁶also known as variability.

⁷The symbol for variance is interesting as each domain have their own definition, as s^2 , σ^2 and $\text{Var}()$ are all acceptable symbols.

Therefore, to obtain the variance of the data, take the difference (i.e., $x_j - \bar{x}$) of each data value from the mean, square it, take the sum of these n squares, and divide it by $n - 1$.

To get the standard deviation s , take the square root of s^2 .

⁸which we calculated previously Returning back to our super alloy example, using $\bar{x} = 260/3^8$, we get for the data given in Eq. (3.1) the variance:

$$s^2 = \frac{1}{29} \left[\left(89 - \frac{260}{3} \right)^2 + \left(77 - \frac{260}{3} \right)^2 + \dots + \left(89 - \frac{260}{3} \right)^2 \right] = \frac{2006}{87} \approx 23.06 \blacksquare$$

Therefore, the standard deviation is calculated to be:

$$s = \sqrt{2006/87} \approx 4.802$$

The standard deviation has the same dimension as the data values, which is an advantage, whereas, the variance is preferable to the standard deviation in developing statistical methods.

Empirical Rule

For any round-shaped symmetric distribution of data the intervals:

$$\bar{x} \pm s, \quad \bar{x} \pm 2s, \quad \bar{x} \pm 3s, \quad \text{contain about } 68\%, \quad 95\%, \quad 99.7\%.$$

respectively, of the data points. This information is quite useful in doing quick calculation of statistical properties such as the quality of production which will be the focus in Chapter 4.

Exercise 3.4: Empirical Rule Outliers and z-Score

For the data set given in Example 1.1, with $\bar{x} = 86.7$ and $s = 4.8$, the three (3) intervals in the Rule are:

$$81.9 \leq x \leq 91.5, \quad 77.1 \leq x \leq 96.3, \quad 72.3 \leq x \leq 101.1$$

and contain 73% (22 values remain, 5 are too small, and 5 too large), 93% (28 values, 1 too small, and 1 too large), and 100%, respectively.

If we reduce the sample by omitting the outlier value of 99, mean and standard deviation reduce to $\bar{x}_{\text{red}} = 86.2$, and $s_{\text{red}} = 4.3$, approximately, and the percentage values become 67% (5 and 5 values outside), 93% (1 and 1 outside), and 100%.

Finally, the relative position of a value x in a set of mean \bar{x} and standard deviation s can be measured by the **z-score**:

$$z(s) = \frac{x - \bar{x}}{s}$$

This is the distance of x from the mean \bar{x} measured in multiples of s . For instance:

$$z(s) = \frac{(83 - 86.7)}{4.8} = -0.77$$

This is negative because 83 lies below the mean. By the empirical rule, the extreme z-values are about -3 and 3. \blacksquare

3.2 Experiments & Outcomes

Now we have the basis covered, it is time to look at probability theory⁹. This theory has the purpose of providing mathematical models of situations affected or even governed by **change effects**, for instance, in weather forecasting, life insurance, quality of technical products (computers, batteries, steel sheets, etc.), traffic problems, and, of course, games of chance with cards or dice, and the accuracy of these models can be tested by suitable observations or experiments.

⁹Sometimes known as probability calculus.

Let's start by defining some standard terms:

experiment A process of measurement or observation, in a laboratory, in a factory, ...

randomness Situation where absolute prediction is not possible.

trial A single performance of an experiment

outcome The result of a trial¹⁰

¹⁰also known as sample point.

sample space Defined as S , is the set of all possible outcomes of an experiment.

Exercise 3.5: Sample Spaces of Random Experiments & Events

- Inspecting a lightbulb | $S = \{\text{Defective, Non-defective}\}$.
- Rolling a die | $S = \{1, 2, 3, 4, 5, 6\}$
 - events are
 - $A = 1, 3, 5$ ("Odd number")
 - $B = 2, 4, 6$ ("Even number"), etc.
- Counting daily traffic accidents in Vienna | $S = \{\text{the integers in some interval}\}$.

3.3 Probability

The **probability** of an event A in an experiment is to measure **how frequently** A is roughly to occur if we make many trials. If we flip a coin, then heads H and tails T will appear **about** equally¹¹ often.

¹¹on the condition, the measurements are done for a long time.

we say that H and T are "**equally likely**."

¹²called a fair dice Similarly, for a regularly shaped die of homogeneous material¹² each of the six (6) outcomes $1, \dots, 6$ will be equally likely. These are examples of experiments in which the sample space S consists of finitely many outcomes (points) that for reasons of some symmetry can be regarded as equally likely.

Let's formulate this in a theory.

Theory 3.1: First Definition of Probability

If the sample space S of an experiment consists of **finitely** many outcomes (points) being equally likely, the probability $P(A)$ of an event A is defined to be:

$$P(A) = \frac{\text{Number of points in } A}{\text{Number of points in } S}.$$

From this definition it follows immediately, in particular, the probability of all events occurring in the sample space S is:

$$P(S) = 1.$$

Exercise 3.6: Fair Die

In rolling a fair die once:

1. What is the probability $P(A)$ of A of obtaining a 5 or a 6?
2. The probability of B : "Even number"?

Solution

The six outcomes are equally likely, so that each has probability $1/6$. Therefore:

$$P(A) = \frac{2}{6} = \frac{1}{3} \quad \text{and} \quad P(B) = \frac{3}{6} = \frac{1}{2} \blacksquare$$

The above theory takes care of many games as well as some practical applications, but not of all experiments, as in many problems we do not have finitely many equally likely outcomes. To arrive at a more general definition of probability, we regard probability as the counterpart of **relative frequency**:

$$f_{\text{rel}}(A) = \frac{f(A)}{n} = \frac{\text{Number of times } A \text{ occurs}}{\text{Number of trials}} \quad (3.4)$$

Now if A did not occur, then $f(A) = 0$. If A always occurred, then $f(A) = n$. These are of course extreme cases. Division by n gives:

$$0 \leq f_{\text{rel}}(A) \leq 1 \quad (3.5)$$

¹³meaning that some event always occurs

In particular, for $A = S$ we have $f(S) = n$ as S always occurs¹³. Division by n gives:

$$f_{\text{rel}}(S) = 1 \quad (3.6)$$

Finally, if A and B are **mutually exclusive**, they cannot occur together. Therefore the absolute frequency of their union $A = B$ must equal the sum of the absolute frequencies of A and B . Division

by n gives the same relation for the relative frequencies:

$$f_{\text{rel}}(A \cup B) = f_{\text{rel}}(A) + f_{\text{rel}}(B) \quad (3.7)$$

We can now extend the definition of probability to experiments in which equally likely outcomes are not available.

Theory 3.2: General Definition of Probability

Given a sample space S , with each event A of S (A being a subset of S) there is associated a number $P(A)$, called the **probability** of A , such the following **axioms of probability** are satisfied.

- For every A in S ,

$$0 \leq P(A) \leq 1. \quad (3.8)$$

- The entire sample space S has the probability

$$P(S) = 1. \quad (3.9)$$

- For **mutually exclusive** events A and B :

$$P(A \cup B) = P(A) + P(B) \quad (A \cap B = \emptyset). \quad (3.10)$$

- If S is **infinite**¹⁴, the previous statement has to be replaced by Eq. (3.4), where for mutually exclusive events A_1, A_2, \dots ,

$$P(A_1 \cup A_2 \cup \dots) = P(A_1) + P(A_2) + \dots \quad (3.11)$$

¹⁴i.e., has infinitely many points.

In the infinite case the subsets of S on which $P(A)$ is defined are restricted to form a so-called σ -algebra.

Basic Theorems of Probability

We will see that the axioms of probability will enable us to build up probability theory and its application to statistics. We begin with three (3) basic theorems. The first one is useful if we can get the probability of the complement A^c more easily than $P(A)$ itself.

Theory 3.3: Complementation Rule

For an event A and its complement A^c in a sample space S ,

$$P(A^c) = 1 - P(A) \quad (3.12)$$

Exercise 3.7: Coin Tossing

Five (5) coins are tossed simultaneously.

Find the probability of the event A :

At least one head turns up. Assume that the coins are fair.

Solution

As each coin can turn up either heads or tails, the sample space consists of $2^5 = 32$ outcomes. Given the coins are fair, we may assign the same probability $(1/32)$ to each outcome. Then the event A^c (No heads turn up) consists of only 1 outcome. Hence $P(A^c) = 1/32$, and the answer is:

$$P(A) = 1 - P(A^c) = \frac{31}{32} \quad \blacksquare$$

Theory 3.4: Addition Rule for Mutually Exclusive Events

For **mutually exclusive events** A_1, \dots, A_m in a sample space S ,

$$P(A_1 \cup A_2 \cup \dots \cup A_m) = P(A_1) + P(A_2) + \dots + P(A_m). \quad (3.13)$$

Exercise 3.8: Mutually Exclusive Events

If the probability that on any workday a garage will get 10-20, 21-30, 31-40, over 40 cars to service is 0.20, 0.35, 0.25, 0.12, respectively, what is the probability that on a given workday the garage gets at least 21 cars to service?

Solution

As these are mutually exclusive events, the answer is:

$$0.35 + 0.25 + 0.12 = 0.72 \blacksquare$$

However, most situations, events will **NOT** be mutually exclusive. Then we have the following theorem to formalise the previous statement.

Theory 3.5: Addition Rule for Arbitrary Events

For events A and B in a sample space, their union is defined as:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B). \quad (3.14)$$

For **mutually exclusive** events A and B we have $A \cap B = \emptyset$ by definition:

$$P(\emptyset) = 0 \quad (3.15)$$

Exercise 3.9: Union of Arbitrary Events

In tossing a fair die, what is the probability of getting an odd number or a number less than 4?

Solution

Let A be the event "Odd number" and B the event "Number less than 4." As these events are linked we can write:

$$P(A \cup B) = \frac{3}{6} + \frac{3}{6} - \frac{2}{6} = \frac{2}{3}$$

as $A \cup B = \text{Odd number less than 4} = \{1, 3\}$ ■

Conditional Probability and Independent Events

It is often required to find the probability of an event B given the condition of an event A occurs. This probability is called the **conditional probability** of B given A and is denoted by $P(B|A)$.

In this case A serves as a new, reduced, sample space, and that probability is the fraction of $P(A)$ which corresponds to $A \cap B$. Therefore,

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{where} \quad P(B) \neq 0 \quad (3.16)$$

Similarly, the conditional probability of A given B is:

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad \text{where} \quad P(A) \neq 0 \quad (3.17)$$

Theory 3.6: Multiplication Rule

Given A and B are events defined in a sample space S and $P(A) \neq 0, P(B) \neq 0$, then

$$P(A \cap B) = P(A) P(B|A) = P(B) P(A|B). \quad (3.18)$$

Exercise 3.10: Multiplication Rule

In producing screws, let:

- A mean "screw too slim",
- B mean "screw too short."

Let $P(A) = 0.1$ and let the conditional probability that a slim screw is also too short be $P(B|A) = 0.2$. What is the probability that a screw that we pick randomly from the lot produced will be both too slim and too short?

Solution

$$P(A \cap B) = P(A) P(B|A) = 0.1 \times 0.2 = 0.02 = 2\% \quad \blacksquare$$

Independent Events

If events A and B are such that

$$P(A \cap B) = P(A) P(B), \quad (3.19)$$

they are called **independent events**. Assuming $P(A) \neq 0, P(B) \neq 0$, we see from Eq. (3.16) - Eq. (3.18):

$$P(A|B) = P(A), \quad P(B|A) = P(B).$$

This means that the probability of A does not depend on the occurrence or nonoccurrence of B, and conversely. This justifies the term **independent**.

Independence of m Events

Similarly, m events A_1, \dots, A_m are called **independent** if:

$$P(A_1 \cap \dots \cap A_m) = P(A_1) \dots P(A_m) \quad (3.20)$$

as well as for every k different events $A_{j_1}, A_{j_2}, \dots, A_{j_k}$.

$$P(A_{j_1} \cap A_{j_2} \cap \dots \cap A_{j_k}) = P(A_{j_1}) P(A_{j_2}) \dots P(A_{j_k}) \quad (3.21)$$

where $k = 2, 3, \dots, m - 1$. Accordingly, three events A, B, C are independent if and only if

$$P(A \cap B) = P(A) P(B), \quad (3.22)$$

$$P(B \cap C) = P(B) P(C), \quad (3.23)$$

$$P(C \cap A) = P(C) P(A), \quad (3.24)$$

$$P(A \cap B \cap C) = P(A) P(B) P(C). \quad (3.25)$$

Sampling

Our next example has to do with randomly drawing objects, *one at a time*, from a given set of objects. This is called **sampling from a population**, and there are two ways of sampling, as follows.

■ **In sampling with replacement**, the object that was drawn at random is placed back to the given set and the set is mixed thoroughly. Then we draw the next object at random.

■ **In sampling without replacement** the object that was drawn is put aside.

Exercise 3.11: Sampling w/o Replacement

A box contains 10 screws, three (3) of which are defective. Two screws are drawn at random. Find the probability that neither of the two screws is defective.

Solution

We consider the events

- A First drawn screw non-defective,
- B Second drawn screw non-defective.

We can see:

$$P(A) = \frac{1}{10}$$

as 7 of the 10 screws are non-defective and we sample at random, so that each screw has the same probability ($\frac{1}{10}$) of being picked.

If we sample with replacement, the situation before the second drawing is the same as at the beginning, and $P(B) = \frac{7}{10}$. The events are independent, and the answer is

$$P(A \cap B) = P(A) P(B) = 0 \cdot 7 \cdot 0.7 = 0.49\%.$$

If we sample without replacement, then $P(A) = \frac{7}{10}$, as before. If A has occurred, then there are 9 screws left in the box, 3 of which are defective.

Thus $P(B|A) = \frac{6}{9} = \frac{2}{3}$, therefore:

$$P(A \cap B) = \frac{7}{10} \cdot \frac{2}{3} = 47\% \blacksquare$$

3.4 Permutations & Combinations

Permutations and combinations help in finding probabilities $P(A) = a/k$ by systematically counting the number a of points of which an event A consists.

where, k is the number of points of the sample space S .

The practical difficulty is that a may often be surprisingly large, so that actual counting becomes hopeless. For example, if in assembling some instrument you need 10 different screws in a certain order and you want to draw them randomly from a box¹⁵ the probability of obtaining them in the required order is only 1/3,628,800 because there are exactly:

$$10! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 10 = 3,628,800$$

¹⁵Of course, this goes without saying, there is nothing but screws in this imaginary box.

orders in which they can be drawn. Similarly, in many other situations the numbers of orders, arrangements, etc. are often incredibly large.

3.4.1 Permutations

A **permutation** of given things¹⁶ is an arrangement of these things in a row in some order.

¹⁶such as *elements* or *objects*.

i.e., for three (3) letters a, b, c there are $3! = 1 \cdot 2 \cdot 3 = 6$ permutations: abc, acb, bca, cab, cba

Let's write this behaviour down as a theory:

Theory 3.7: Permutations

Different things

The number of permutations of n different things taken all at a time is

$$n! = 1 \cdot 2 \cdot 3, \dots, n. \quad (3.26)$$

Classes of Equal Things

If n given things can be divided into c classes of alike things differing from class to class, then the number of permutations of these things taken all at a time is

$$\frac{n!}{n_1!n_2!\cdots n_c!} \quad \text{where} \quad n_1 + n_2 + \cdots + n_c = n, \quad (3.27)$$

where n_j is the number of things in the j^{th} class.

Permutation of n things taken k at a time

A permutation containing only k of the n given things. Two such permutations consisting of the same k elements, in a different order, are different, by definition.

i.e., there are 6 different permutations of the three letters a, b, c , taken two letters at a time, ab, ac, bc, ba, ca, cb .

Permutation of n things taken k at a time with repetitions

An arrangement obtained by putting any given thing in the first position, any given thing, including a repetition of the one just used, in the second, and continuing until k positions are filled.

i.e., there are $3^2 = 9$ different such permutations of a, b, c taken 2 letters at a time, namely, the preceding 6 permutations and aa, bb, cc .

Theory 3.8: Permutations

The number of different permutations of n different things taken k at a time **without repetitions** is

$$n(n-1)(n-2)\cdots(n-k+1) = \frac{n!}{(n-k)!}, \quad (3.28)$$

and **with repetitions** is,

$$n^k. \quad (3.29)$$

Exercise 3.12: An Encrypted Message

In an encrypted message the letters are arranged in groups of five (5) letters, called words. Knowing the letter can be repeated, we see that the number of different such words is

$$26^5 = 11,881,376 \blacksquare$$

For the case of different such words containing each letter no more than once is

$$\frac{26!}{(26-5)!} = 26 \cdot 25 \cdot 24 \cdot 23 \cdot 22 = 7,893,600 \blacksquare$$

3.4.2 Combinations

In a permutation, the **order of the selected things is essential**. In contrast, a **combination** of a given things means any selection of one or more things **without regard to order**. There are two (2) kinds of combinations, as follows:

1. The number of **combinations of n different things, taken k at a time, without repetitions** is the number of sets that can be made up from the n given things, each set containing k different things and no two (2) sets containing exactly the same k things.
2. The number of **combinations of n different things, taken k at a time, with repetitions** is the number of sets that can be made up of k things chosen from the given n things, each being used as often as desired.

i.e, there are three (3) combinations of the three (3) letters a, b, c , taken two (2) letters at a time, without repetitions, namely, ab, ac, bc , and six such combinations with repetitions, namely, ab, ac, bc, ca, bb, cc .

Theory 3.9: Combinations

The number of different combinations of n different things taken, k at a time, **without repetitions**, is:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\cdots(n-k+1)}{1\cdot2\cdots k}, \quad (3.30)$$

and the number of those combinations **with repetitions** is:

$$\binom{n+k-1}{k}. \quad (3.31)$$

Exercise 3.13: Sampling Light-bulbs

The number of samples of five (5) light-bulbs that can be selected from a lot of 500 bulbs is

$$\binom{500}{5} = \frac{500!}{5!495!} = \frac{500 \cdot 499 \cdot 498 \cdot 497 \cdot 476}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5} = 255,244,687,600 \blacksquare$$

3.4.3 Factorial Function

In Eq. (3.26)-Eq. (3.31) the **factorial function** is relatively straightforward. By definition¹⁷,

$$0! = 1.$$

Values may be computed recursively from given values by

$$(n+1)! = (n+1)n!.$$

For large n the function is very large and hard to keep track of. A convenient approximation for large n is the **Stirling formula**, defined as:

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \quad \text{where} \quad e = 2.718\cdots \quad (3.32)$$

where \sim is read **asymptotically equal**¹⁸ and means that the ratio of the two sides of Eq. (3.32) approaches 1 as n approaches infinity.

¹⁷This is done by convention. An intuitive way to look at it is $n!$ counts the number of ways to arrange distinct objects in a line, and there is only one way to arrange nothing.

¹⁸it means the percentage difference between the vertical distances between points on the two graphs approaches 0.

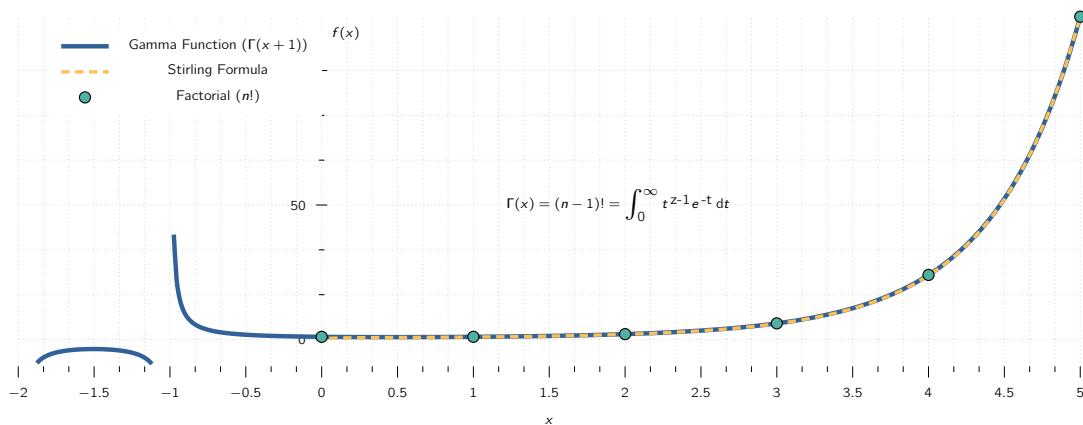


Figure 3.2: A visual comparison of the Stirling formula and the actual values of the factorial function.

3.4.4 Binomial Coefficients

The **binomial coefficients** are defined by the following formula:

$$\binom{a}{k} = \frac{(a)(a-1)(a-2)\cdots(a-k+1)}{k!} \quad \text{where } (k \geq 0, \text{ integer}) \quad (3.33)$$

The numerator has k factors. Furthermore, we define

$$\binom{a}{0} = 1, \quad \text{in particular,} \quad \binom{0}{0} = 1.$$

For integer $a = n$ we obtain from Eq. (3.33):

$$\binom{n}{k} = \binom{n}{n-k} \quad (n \geq 0 \quad \text{and} \quad 0 \leq k \leq n).$$

Binomial coefficients may be computed recursively, because

$$\binom{a}{k} + \binom{a}{k+1} = \binom{a+1}{k+1} \quad (k \geq 0, \text{ integer}).$$

Formula Eq. (3.33) also gives:

$$\binom{-m}{k} = (-1)^k \binom{m+k-1}{k} \quad \text{where} \quad k \geq 0, \text{ integer} \quad \text{and} \quad m > 0.$$

There are two (2) important relations worth mentioning:

$$\sum_{s=0}^{n-1} \binom{k+s}{k} = \binom{n+k}{k+1} \quad (k \geq 0 \quad \text{and} \quad n \geq 1)$$

and

$$\sum_{k=0}^r \binom{p}{k} \binom{q}{r-k} = \binom{p+q}{r} \quad (r \geq 0, \text{ integer}).$$

3.5 Random Variables and Probability Distributions

In the beginning of this chapter we considered frequency distributions of data¹⁹. These distributions show the **absolute** or **relative** frequency of the data values.

¹⁹Remember we did a histogram and a stem-and-leaf plot.

Similarly, a **probability distribution** or, a **distribution**, shows the probabilities of events in an experiment. The quantity we observe in an experiment will be denoted by X and called a **random variable**²⁰ as the value it will assume in the next trial depends on the **stochastic process**

²⁰or **stochastic variable** if you want to be pedantic.

i.e., if you roll a die, you get one of the numbers from 1 to 6, but you don't know which one will show up next. An example would be, $X = \text{Number a die turns up}$, which is a random variable.

If we count²¹, we have a **discrete random variable and distribution**. If we measure (electric voltage, rainfall, hardness of steel), we have a **continuous random variable and distribution**. For both cases (discrete, discontinuous), the distribution of X is determined by the **distribution function**:

$$F(x) = P(X \leq x) \quad (3.34)$$

This is the probability that in a trial, X will assume any value not exceeding x .

The terminology is unfortunately **NOT** uniform across the field as $F(x)$ is sometimes also called the **cumulative distribution function**.

For Eq. (3.34) to make sense in both the discrete and the continuous case we formulate conditions as follows.

Theory 3.10: Random Variable

A **random variable** X is a function defined on the sample space S of an experiment. Its values are real numbers. For every number a the probability:

$$P(X = a),$$

with which X assumes a is defined. Similarly, for any interval I , the probability

$$P(X \in I),$$

with which X assumes any value in I is defined²².

²²Although this definition is very general, in practice only a very small number of distributions will occur over and over again in applications.

From Eq. (3.34) we can define the fundamental formula for the probability corresponding to an interval $a < x \leq b$:

$$P(a < X \leq b) = F(b) - F(a). \quad (3.35)$$

This follows because $X \leq a$ (X assumes any value **NOT** exceeding a) and $a < X \leq b$ (X assumes any value in the interval $a < x \leq b$) are **mutually exclusive** events, so based on Eq. (3.34):

$$\begin{aligned} F(b) &= P(X \leq b) = P(X \leq a) + P(a < X \leq b) \\ &= F(a) + P(a < X \leq b) \end{aligned}$$

and subtraction of $F(a)$ on both sides gives Eq. (3.35).

3.5.1 Discrete Random Variables and Distributions

By definition, a random variable X and its distribution are **discrete** if X assumes only **finitely** many or at most countably many values x_1, x_2, x_3, \dots , called the **possible values** of X , with positive probabilities,

$$p_1 = P(X = x_1), p_2 = P(X = x_2), p_3 = P(X = x_3), \dots$$

whereas the probability $P(X \in I)$ is zero for any interval I containing no possible value. Clearly, the discrete distribution of X is also determined by the **probability function** $f(x)$ of X , defined by

$$f(x) = \begin{cases} p_j & \text{if } x = x_j \\ 0 & \text{otherwise} \end{cases} \quad \text{where } j = 1, 2, \dots, \quad (3.36)$$

From this we get the values of the **distribution function** $F(x)$ by taking sums,

$$F(x) = \sum_{x_j \leq x} f(x_j) = \sum_{x_j \leq x} p_j \quad (3.37)$$

where for any given x we sum all the probabilities p_j for which x_j is smaller than or equal to that of x . This is a **step function** with upward jumps of size p_j at the possible values x_j of X and constant in between. The two (2) useful formulas for discrete distributions are readily obtained as follows. For the probability corresponding to intervals we have from Eq. (3.35) and Eq. (3.37):

$$P(a < X \leq b) = F(b) - F(a) = \sum_{a < x_j \leq b} p_j \quad (3.38)$$

²³Be careful about $<$ and \leq as the former means it is NOT included and the latter means it is.

This is the sum of all probabilities p_j for which x_j satisfies $a < x_j \leq b$ ²³. From this and $P(S) = 1$ we obtain the following formula.

$$\sum_j p_j = 1 \quad (\text{sum of all probabilities}). \quad (3.39)$$

Exercise 3.14: Waiting Time Problem

In tossing a fair coin, let X be the Number of trials until the first head appears. Then, by independence of events we get (where H is heads, and T is tails):

$$\begin{aligned} P(X = 1) &= P(H) = \frac{1}{2} \\ P(X = 2) &= P(TH) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} \\ P(X = 3) &= P(TTH) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8} \end{aligned}$$

and in general, $P(X = n) = \left(\frac{1}{2}\right)^n$, $n = 1, 2, 3, \dots$ which when all possible event are summed up will always give 1.

3.5.2 Continuous Random Variables and Distributions

Discrete random variables appear in experiments in which we count²⁴. Continuous random variables appear in experiments in which we measure (lengths of screws, voltage in a power line, etc.). By definition, a random variable X and its distribution are of *continuous type* or, briefly, **continuous**, if its distribution function $F(x)$, defined in Eq. (3.34), can be given by an integral²⁵:

$$F(x) = \int_{-\infty}^x f(v) dv \quad (3.40)$$

²⁴defectives in a production, days of sunshine in Kufstein, customers in a line, etc.

²⁵we write v as a toss-away variable because x is needed as the upper limit of the integral.

whose integrand $f(x)$, called the **density** of the distribution, is **non-negative**, and is continuous, perhaps except for finitely many x -values. Differentiation gives the relation of f to F as

$$f(x) = F'(x) \quad (3.41)$$

for every x at which $f(x)$ is continuous.

From Eq. (3.35) and Eq. (3.40) we obtain the very important formula for the probability corresponding to an interval²⁶:

²⁶This is an analog of Eq. (3.38)

$$P(a < X \leq b) = F(b) - F(a) = \int_a^b f(v) dv \quad (3.42)$$

Which can be seen visually in **Fig. 3.3**. From Eq. (3.40) and $P(S) = 1$ we also have the analogue of Eq. (3.39):

$$\int_{-\infty}^{\infty} f(v) dv = 1. \quad (3.43)$$

Continuous random variables are **simpler than discrete ones** with respect to intervals as, in the continuous case the four probabilities corresponding to $a < X \leq b$, $a < X < b$, $a \leq X \leq b$,

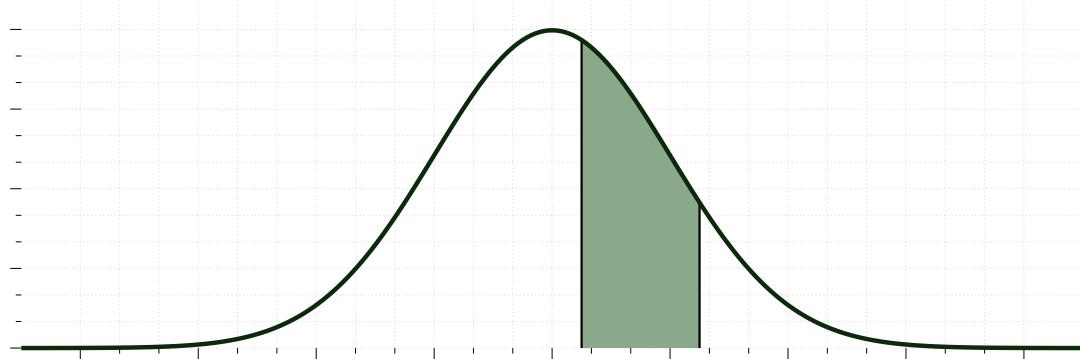


Figure 3.3: A visual representation of the Eq. (3.42).

and $a \leq X \leq b$ with any fixed a and b ($> a$) are all the same.

The next example illustrates notations and typical applications of our present formulas.

Exercise 3.15: Continuous Distribution

Let X have the density function:

$$f(x) = 0.75(1 - x^2) \quad \text{if} \quad -1 \leq x \leq 1,$$

and zero otherwise. Find:

1. The distribution function.
2. Find the probabilities $P\left(-\frac{1}{2} \leq X \leq \frac{1}{2}\right)$ and $P\left(\frac{1}{2} \leq X \leq 2\right)$
3. Find x such that $P(X \leq x) = 0.95$.

Solution

From Eq. (3.40), we obtain $F(x) = 0$ if $x \leq -1$,

$$F(x) = 0.75 \int_{-1}^x (1 - v^2) dv = 0.5 + 0.75x - 0.25x^3 \quad \text{if} \quad -1 < x \leq 1,$$

and $F(x) = 1$ if $x > 1$. From this and Eq. (3.42) we get:

$$P\left(-\frac{1}{2} \leq X \leq \frac{1}{2}\right) = F\left(\frac{1}{2}\right) - F\left(-\frac{1}{2}\right) = 0.75 \int_{-1/2}^{1/2} (1 - v^2) dv = 68.75\%$$

because $P\left(-\frac{1}{2} \leq X \leq \frac{1}{2}\right) = P\left(-\frac{1}{2} < X \leq \frac{1}{2}\right)$ for a continuous distribution we can write:

$$P\left(\frac{1}{4} \leq X \leq 2\right) = F(2) - F\left(\frac{1}{4}\right) = 0.75 \int_{1/4}^1 (1 - v^2) dv = 31.64\%.$$

Note that the upper limit of integration is 1, not 2. Finally,

$$P(X \leq x) = F(x) = 0.5 + 0.75x - 0.25x^3 = 0.95.$$

Algebraic simplification gives $3x - x^3 = 1.8$. A solution is $x = 0.73$, approximately ■

3.6 Mean and Variance of a Distribution

The mean μ and variance σ^2 of a random variable X and of its distribution are the theoretical counterparts of the mean \bar{x} and variance s^2 of a frequency distribution and serve a similar purpose.

The mean characterises the central location and the variance the spread (the variability) of the distribution. The **mean** μ is defined by:

$$(a) \quad \mu = \sum_j x_j f(x_j) \quad (\text{Discrete distribution}) \quad (3.44a)$$

$$(b) \quad \mu = \int_{-\infty}^{\infty} x f(x) dx \quad (\text{Continuous distribution}) \quad (3.44b)$$

and the **variance** σ^2 by:

$$(a) \quad \sigma^2 = \sum_j (x_j - \mu)^2 f(x_j) \quad (\text{Discrete distribution}) \quad (3.45a)$$

$$(b) \quad \sigma^2 = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx \quad (\text{Continuous distribution}) \quad (3.45b)$$

σ (the positive square root of σ^2) is called the standard deviation²⁷ of X and its distribution. f is the probability function or the density, respectively, in (a) and (b).

²⁷Sometimes it is known as $\text{Var}(x)$

The mean μ is also denoted by $E(X)$ and is called the **expectation of X** because it gives the average value of X to be expected in many trials.

Quantities such as μ and σ^2 that measure certain properties of a distribution are called **parameters**. μ and σ^2 are the two (2) most important ones.

From Eq. (3.45a) and Eq. (3.45b), we see that²⁸:

$$\sigma^2 > 0$$

²⁸except for a discrete distribution with only one possible value.

We assume that μ and σ^2 exist²⁹, as is the case for practically all distributions that are useful in applications.

²⁹and finite.

Exercise 3.16: Mean and Variance

The random variable X , *Number of heads in a single toss of a fair coin*, has the possible values $X = 0$ and $X = 1$ with probabilities $P(X = 0) = \frac{1}{2}$ and $P(X = 1) = \frac{1}{2}$. From Eq. (3.44a) we thus obtain the mean:

$$\mu = 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = \frac{1}{2},$$

and Eq. (3.45a) gives the variance:

$$\sigma^2 = (0 - \frac{1}{2})^2 \cdot \frac{1}{2} + (1 - \frac{1}{2})^2 \cdot \frac{1}{2} = \frac{1}{4} \blacksquare$$

Symmetry

We can obtain the mean μ without calculation if a distribution is symmetric. Indeed, we can write:

Theory 3.11: Mean of a Symmetric Distribution

If a distribution is **symmetric** with respect to $x = c$, that is,

$$f(c - x) = f(c + x)$$

then $\mu = c$.

Transformation of Mean and Variance

Given a random variable X with mean μ and variance σ^2 , we want to calculate the mean and variance of $X^* = a_1 + a_2 X$, where a_1 and a_2 are given constants.

This problem is important in statistics, where it often appears.

Theory 3.12: Transformation of Mean and Variance

If a random variable X has mean μ and variance σ^2 , then the random variable:

$$X^* = a_1 + a_2 X \quad \text{where} \quad a_2 > 0$$

has the mean μ^* and variance σ^{*2} , where

$$\mu^* = a_1 + a_2 \mu \quad \text{and} \quad \sigma^{*2} = a_2^2 \sigma^2.$$

In particular, the **standardised random variable** Z corresponding to X , given by:

$$Z = \frac{X - \mu}{\sigma}$$

has the mean 0 and the variance 1.

Expectation & Moments

³⁰the value of X to be expected on the average

If we recall, Eq. (3.44a) and Eq. (3.44b) define the mean of X ³⁰, written $\mu = E(X)$. More generally, if $g(x)$ is **non-constant** and continuous for all x , then $g(X)$ is a random variable. Therefore its **mathematical expectation** or, briefly, its expectation $E(g(X))$ is the value of $g(X)$ to be expected on the average, defined by:

$$E(g(X)) = \sum_j g(x_j) f(x_j) \quad \text{or} \quad E(g(X)) = \int_{-\infty}^{\infty} g(x) f(x) dx$$

In the formula on the Left Hand Side (LHS), f is the probability function of the discrete random variable X . In the formula on the RHS, f is the density of the continuous random variable X . Important special cases are the k^{th} of X (where $k = 1, 2, \dots$)

$$E(X^k) = \sum_j x_j^k f(x_j) \quad \text{or} \quad \int_{-\infty}^{\infty} x^k f(x) dx$$

and the k^{th} of X ($k = 1, 2, \dots$)

$$E([X - \mu]^k) = \sum_j (x_j - \mu)^k f(x_j) \quad \text{or} \quad \int_{-\infty}^{\infty} (x - \mu)^k f(x) dx.$$

This includes the first moment, the **mean** of X

$$\mu = E(X) \quad \text{where} \quad k = 1 \quad (3.46)$$

It also includes the second central moment, the **variance** of X

$$\sigma^2 = E([X - \mu]^2) \quad \text{where} \quad k = 2. \quad (3.47)$$

3.7 Binomial, Poisson, and Hyper-geometric Distributions

These are the three (3) most important **discrete** distributions, with numerous applications therefore are worth of a bit of a detailed look.

Of course these are not the only distributions present. There are as many distributions as there are problems with some distributions used in wide variety of fields (Gaussian) whereas some are used only in a very narrow field (Nakagami).

Binomial Distribution

The **binomial distribution** occurs in problems involving chance³¹.

What we are interested is in the number of times an event A occurs in n **independent** trials. In each trial, the event A has the same probability $P(A) = p$. Then in a trial, A will **NOT** occur with probability $q = 1 - p$. In n trials the random variable that interests us is:

$$X = \text{Number of times the event } A \text{ occurs in } n \text{ trials.} \quad (3.48)$$

X can assume the values $0, 1, \dots, n$, and we want to determine the corresponding probabilities. Now $X = x$ means that A occurs in x trials and in $n - x$ trials it does not occur. We can write this down as follows:

$$\underbrace{A \ A \ \dots A}_{x \text{ times}} \quad \text{and} \quad \underbrace{B \ B \ \dots B}_{n - x \text{ times}} \quad (3.49)$$

Here $B = A^c$ is the complement of A , meaning that A does not occur. We now use the assumption that the trials are independent³². Hence Eq. (3.49) has the probability:

$$\underbrace{p \ p \ \dots p}_{x \text{ times}} \cdot \underbrace{q \ q \ \dots q}_{n - x \text{ times}} = p^x q^{n-x} \quad (3.50)$$

Now Eq. (3.49) is just one order of arranging xA 's and $n - xB$'s. We will now calculate the number of permutations of n things³³ consisting of two (2) classes;

³³the n outcomes of the n trials

1. class 1 containing the $n_1 = x$ A 's
2. class 2 containing the $n - n_1 = n - x$ B 's

This number is:

$$\frac{n!}{x!(n - x)!} = \binom{n}{x}. \quad (3.51)$$

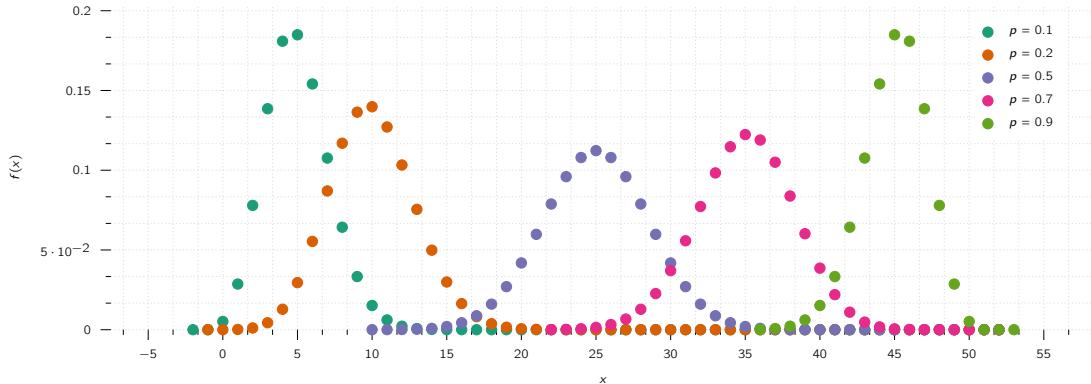


Figure 3.4: Binomial distribution with different values of probability with a sample size of 50.

Accordingly, Eq. (3.50), multiplied by this binomial coefficient, gives the probability $P(X = x)$ of $X = x$, that is, of obtaining A precisely x times in n trials. Hence X has the probability function:

$$f(x) = \binom{n}{x} p^x q^{n-x} \quad (x = 0, 1, \dots, n) \quad (3.52)$$

and $f(x) = 0$ otherwise. The distribution of X with probability function (2) is called the **binomial distribution** or *Bernoulli distribution*. The occurrence of A is called *success*³⁴ and the non-occurrence of A is called *failure*.

³⁴regardless of what it actually is; it may mean that you miss your plane or lose your watch

The mean and variance of the binomial distribution is:

$$\mu = np \quad \text{and} \quad \sigma^2 = npq$$

For the *symmetric case* of equal chance of success and failure ($p = q = \frac{1}{2}$) this gives the mean $n/2$, the variance $n/4$, and the probability function

$$f(x) = \binom{n}{x} \left(\frac{1}{2}\right)^x \quad (x = 0, 1, \dots, n).$$

Exercise 3.17: Binomial Distribution

Calculate the probability of obtaining at least two (2) "six" in rolling a fair die 4 times.

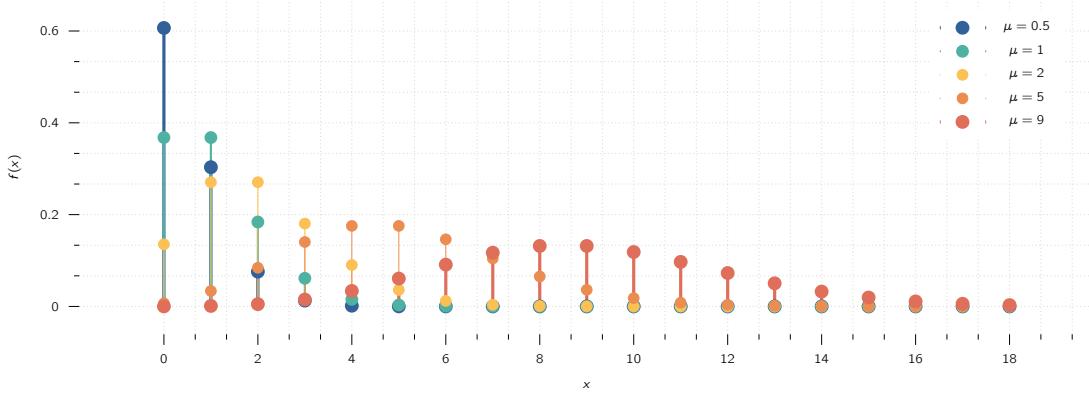
Solution

$p = P(A) = P(\text{six}) = \frac{1}{6}$, $q = \frac{5}{6}$, $n = 4$. The event "At least two (2) "six" occurs if we obtain 2 or 3 or 4 "six" Hence the answer is:

$$\begin{aligned} P &= f(2) + f(3) + f(4) = \binom{4}{2} \left(\frac{1}{6}\right)^2 \left(\frac{5}{6}\right)^2 + \binom{4}{3} \left(\frac{1}{6}\right)^3 \left(\frac{5}{6}\right) + \binom{4}{4} \left(\frac{1}{6}\right)^4 \\ &= \frac{1}{6^4} (6 \cdot 25 + 4 \cdot 5 + 1) = \frac{171}{1296} = 13.2\%. \end{aligned}$$

Poisson Distribution

The discrete distribution with infinitely many possible values and probability function:

Figure 3.5: The Poisson distribution with different mean (μ) values.

$$f(x) = \frac{\mu^x}{x!} e^{-\mu} \quad \text{where} \quad x = 0, 1, \dots \quad (3.53)$$

is called the **Poisson distribution**, named after *S. D. Poisson*. **Fig.** 3.5 shows Eq. (3.53) for some values of μ^{35} .

³⁵While μ is used here, some textbook use λ

It can be proved that this distribution is obtained as a limiting case of the binomial distribution, if we let $p \rightarrow 0$ and $n \rightarrow \infty$ so that the mean $\mu = np$ approaches a finite value. The Poisson distribution has the mean μ and the variance:

$$\sigma^2 = \mu. \quad (3.54)$$

Fig. 3.5 gives the impression that, with increasing mean, the spread of the distribution increases, thereby illustrating formula Eq. (3.54), and that the distribution becomes more and more symmetric.³⁶

³⁶approximately

Exercise 3.18: Poisson Distribution

If the probability of producing a defective screw is $p = 0.01$, what is the probability that a lot of 100 screws will contain more than 2 defectives?

Solution

The complementary event is A^c . No more than 2 defectives. For its probability we get, from the binomial distribution with mean $\mu = np = 1$, the value.

$$P(A^c) = \binom{100}{0} 0.99^{100} + \binom{100}{1} 0.01 \cdot 0.99^{99} + \binom{100}{2} 0.01^2 \cdot 0.99^{98}.$$

Since p is very small, we can approximate this by the much more convenient Poisson distribution with mean $\mu = np = 100 \cdot 0.01 = 1$, obtaining.

$$P(A^c) = e^{-1} \left(1 + 1 + \frac{1}{2} \right) = 91.97\%.$$

Thus $P(A) = 8.03\%$. Show that the binomial distribution gives $P(A) = 7.94\%$, so that the Poisson approximation is quite good ■

Exercise 3.19: The Parking Problem

If on the average, 2 cars enter a certain parking lot per minute, what is the probability that during any given minute four (4) or more cars will enter the lot?

Solution

To understand that the Poisson distribution is a model of the situation, we imagine the minute to be divided into very many short time intervals. Let p be the (constant) probability that a car will enter the lot during any such short interval, and assume independence of the events that happen during those intervals. Then, we are dealing with a binomial distribution with very large n and very small p , which we can approximate by the Poisson distribution with

$$\mu = np = 2$$

because 2 cars enter on the average, the complementary event of the event "4 cars or more during a given minute" is "3 cars or fewer enter the lot" and has the probability

$$f(0) + f(1) + f(2) + f(3) = e^{-2} \left(\frac{2^0}{0!} + \frac{2^1}{1!} + \frac{2^2}{2!} + \frac{2^3}{3!} \right) = 0.857.$$

Which means the result is 14.3% ■

3.7.1 Sampling with Replacement

This means that we draw things from a given set one by one, and after each trial we replace the thing drawn³⁷ before we draw the next thing. This guarantees **independence of trials** and leads to the **binomial distribution**. Indeed, if a box contains N things, for example, screws, M of which are defective, the probability of drawing a defective screw in a trial is $p = M/N$. Hence the probability of drawing a nondefective screw is $q = 1 - p = 1 - M/N$, and Eq. (3.52) gives the probability of drawing x defectives in n trials in the form:

$$f(x) = \binom{n}{x} \left(\frac{M}{N}\right)^x \left(1 - \frac{M}{N}\right)^{n-x} \quad (x = 0, 1, \dots, n). \quad (3.55)$$

³⁷put it back to the given set and mix.

3.7.2 Sampling without Replacement: Hyper-geometric Distribution

Sampling without replacement means that we return no screw to the box. Then we no longer have independence of trials, and instead of Eq. (3.55) the probability of drawing x defectives in n trials is:

$$f(x) = \frac{\binom{M}{x} \binom{N-M}{n-x}}{\binom{N}{n}} \quad \text{where} \quad x = 1, 2, \dots, n. \quad (3.56)$$

The distribution with this probability function is called the **hyper-geometric distribution**³⁸.

The hypergeometric distribution has the mean and the variance:

$$\mu = n \frac{M}{N} \quad \text{and} \quad \sigma^2 = \frac{nM(N-M)(N-n)}{N^2(N-1)}.$$

³⁸because its moment generating function can be expressed by the hypergeometric function, which is a fact only useful to write it in a margin.

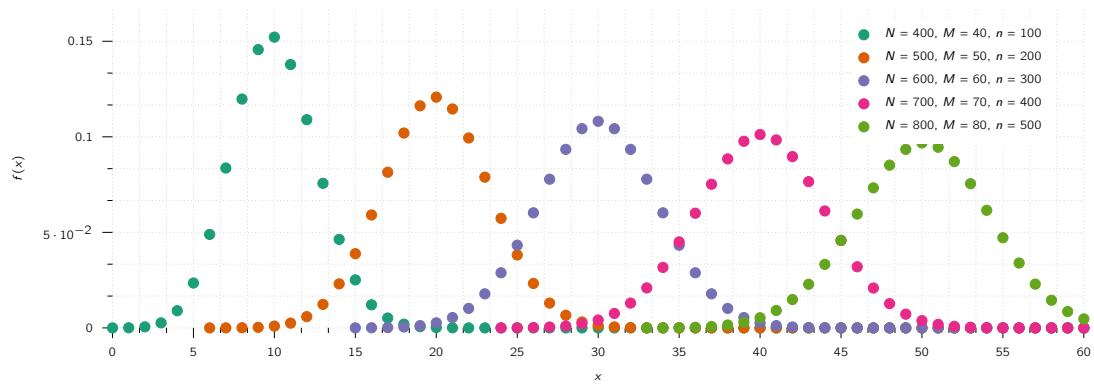


Figure 3.6: The probability density distribution of hyper-geometric distribution with different parameters.

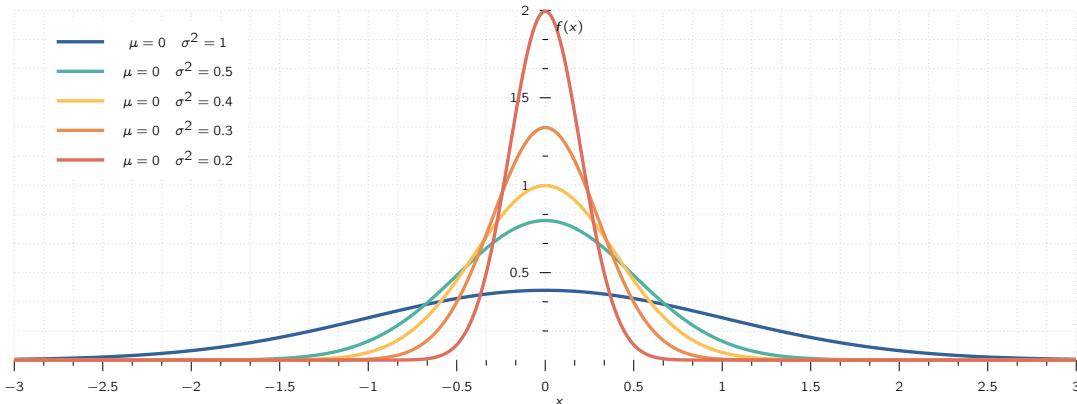


Figure 3.7: The poster child of probability and statistics, the normal distribution.

3.8 Normal Distribution

Turning from discrete to continuous distributions, in this section we discuss the normal distribution. This is the most important continuous distribution because in applications many random variables are normal random variables³⁹ or they are approximately normal or can be transformed into normal random variables in a relatively simple fashion. Furthermore, the normal distribution is a useful approximation of more complicated distributions, and it also occurs in the proofs of various statistical tests.

³⁹that is, they have a normal distribution.

The **normal distribution** or *Gauss distribution* is defined as the distribution with the density:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad (3.57)$$

where \exp is the exponential function with base $e = 2.718\dots$. This is simpler than it may at first look. $f(x)$ has these features (see also Fig. 3.7).

1. μ is the mean, and σ the standard deviation.
2. $1/(\sigma\sqrt{2\pi})$ is a constant factor that makes the area under the curve of $f(x)$ from $-\infty$ to ∞ equal to 1, as it must be⁴⁰.
3. The curve of $f(x)$ is symmetric with respect to $x = \mu$ because the exponent is quadratic. Hence for $\mu = 0$ it is symmetric with respect to the y -axis $x = 0$ ⁴¹.
4. The exponential function in Eq. (3.57) goes to zero very fast—the faster the smaller the standard deviation σ is, as it should be, as seen in Fig. 3.7.

⁴⁰Having a probability higher than 1 does NOT make sense

⁴¹This distribution is also known as bell-shaped curves.

3.8.1 Distribution Function

From Eq. (3.55) and Eq. (3.57) we see that the normal distribution has the **distribution function** of the following form:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x \exp\left[-\frac{1}{2}\left(\frac{v-\mu}{\sigma}\right)^2\right] dv. \quad (3.58)$$

Here we needed x as the upper limit of integration and wrote v (instead of x) in the integrand.

For the corresponding **standardised normal distribution** with mean 0 and standard deviation 1 we denote $F(x)$ by $\Phi(z)$. Then we simply have from Eq. (3.58).

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-u^2/2} du. \quad (3.59)$$

This integral cannot be integrated by one of the methods of calculus.

But this is no serious handicap because its values can be obtained from standardised tables. These values are needed in working with the normal distribution. The curve of $\Phi(z)$ is *S*-shaped. It increases monotone from 0 to 1 and intersects the vertical axis at $\frac{1}{2}$, as shown in **Fig. 3.8**.

Theory 3.13: Relationship between PDF and CDF

The distribution function $F(x)$ of the normal distribution with any μ and σ is related to the standardised distribution function $\Phi(z)$ in Eq. (3.59) by the formula

$$F(x) = \Phi\left(\frac{x-\mu}{\sigma}\right).$$

Theory 3.14: Normal Probabilities for Intervals

The probability a normal random variable X with mean μ and standard deviation σ assume any value in an interval $a < x \equiv b$ is:

$$P(a < X \leq b) = F(b) - F(a) = \Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right).$$

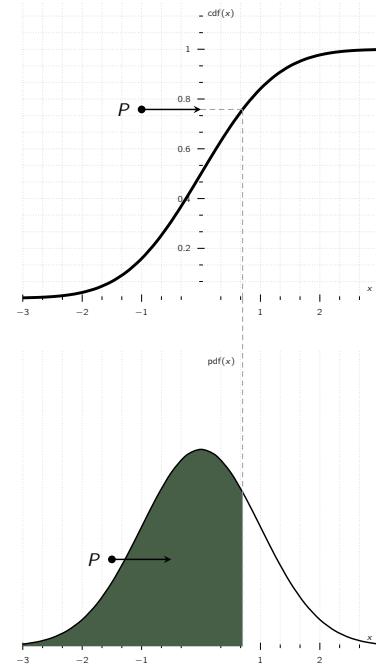


Figure 3.8: A visual representation between the relationship of PDF and CDF.

3.8.2 Numeric Values

In practical work with the normal distribution it is good to remember that about 67% of all values of X to be observed will be between $\mu \pm \sigma$, about 95% between $\mu \pm 2\sigma$, and practically all between

the **three-sigma limits** $\mu \pm 3\sigma$:

$$P(\mu - \sigma < X \leq \mu + \sigma) \approx 68\% \quad (3.60a)$$

$$P(\mu - 2\sigma < X \leq \mu + 2\sigma) \approx 95.5\% \quad (3.60b)$$

$$P(\mu - 3\sigma < X \leq \mu + 3\sigma) \approx 99.7\%. \quad (3.60c)$$

The aforementioned formulas show that a value deviating from μ by more than σ , 2σ , or 3σ will occur in one of about 3, 20, and 300 trials, respectively.

In tests⁴², we shall ask, conversely, for the intervals that correspond to certain given probabilities; practically most important use the probabilities of 95%, 99%, and 99.9%. For these, the answers are $\mu \pm 2\sigma$, $\mu \pm 2.6\sigma$, and $\mu \pm 3.3\sigma$, respectively.

⁴²Which we shall cover in Chapter 4.

More precisely,

$$P(\mu - 1.96\sigma < X \leq \mu + 1.96\sigma) \approx 95\% \quad (3.61a)$$

$$P(\mu - 2.58\sigma < X \leq \mu + 2.58\sigma) \approx 99\% \quad (3.61b)$$

$$P(\mu - 3.29\sigma < X \leq \mu + 3.29\sigma) \approx 99.9\%. \quad (3.61c)$$

3.8.3 Normal Approximation of the Binomial Distribution

The probability function of the binomial distribution, as a reminder, is:

$$f(x) = \binom{n}{x} p^x q^{n-x} \quad (x = 0, 1, \dots, n). \quad (3.62)$$

If n is large, the binomial coefficients and powers become very inconvenient. It is of great practical⁴³ importance that, in this case, the normal distribution provides a good approximation of the binomial distribution, according to the following theorem, one of the most important theorems in all probability theory.

⁴³and theoretical

Theory 3.15: Limit Theorem of De Moivre and Laplace

For large n ,

$$f(x) \sim f^*(x) \quad \text{where} \quad x = 0, 1, \dots, n$$

Here f is given by Eq. (3.62). The function

$$f^*(z) = \frac{1}{\sqrt{2\pi npq}} \exp\left(-\frac{z^2}{2}\right), \quad \text{and} \quad z = \frac{x - np}{\sqrt{npq}}$$

is the density of the normal distribution with mean $\mu = np$ and variance $\sigma^2 = npq$ (the mean and variance of the binomial distribution). Furthermore, for any nonnegative integers a and b ($> a$):

$$P(a \leq X \leq b) = \sum_{x=a}^b \binom{n}{x} p^x q^{n-x} \sim \Phi(\beta) - \Phi(\alpha)$$

where,

$$\alpha = \frac{a - np - 0.5}{\sqrt{npq}} \quad \text{and} \quad \beta = \frac{b - np + 0.5}{\sqrt{npq}}$$

3.9 Distribution of Several Random Variables

Distributions of two (2) or more random variables are of interest for two (2) reasons:

1. They occur in experiments in which we observe several random variables, for example, carbon content X and hardness Y of steel, amount of fertiliser X and yield of corn Y , height X_1 , weight X_2 , and blood pressure X_3 of persons, and so on.
2. They will be needed in the mathematical justification of the methods of statistics in Chapter 4.

In this section we consider two (2) random variables X and Y or, as we also say, a **two-dimensional random variable** (X, Y) . For (X, Y) the outcome of a trial is a pair of numbers $X = x$, $Y = y$, briefly $(X, Y) = (x, y)$, which we can plot as a point in the XY -plane.

The **two-dimensional probability distribution** of the random variable (X, Y) is given by the **distribution function**

$$F(x, y) = P(X \leq x, Y \leq y). \quad (3.63)$$

This is the probability that in a trial, X will assume any value not greater than x and in the same trial, Y will assume any value not greater than y . $F(x, y)$ determines the probability distribution uniquely, because extending the analogy we developed previously, $P(a < X \leq b) = F(b) - F(a)$, we now have for a rectangle defined using the following equation:

$$P(a_1 < X \leq b_1, a_2 < Y \leq b_2) = F(b_1, b_2) - F(a_1, b_2) - F(b_1, a_2) + F(a_1, a_2). \quad (3.64)$$

As before, in the two-dimensional case we shall also have **discrete** and **continuous** random variables and distributions.

3.9.1 Discrete Two-Dimensional Distribution

In analogy to the case of a single random variable, we call (X, Y) and its distribution **discrete** if (X, Y) can assume only finitely many or at most countably infinitely many pairs of values (x_1, y_1) , $(x_2, y_2), \dots$ with positive probabilities, whereas the probability for any domain containing none of those values of (X, Y) is zero.

Let (x_i, y_j) be any of those values and let $P(X = x_i, Y = y_j) = p_{ij}$ (where we admit that p_{ij} may be 0 for certain pairs of subscripts i). Then we define the **probability function** $f(x, y)$ of (X, Y) by:

$$f(x, y) = p_{ij} \quad \text{if} \quad x = x_i, y = y_j \quad \text{and} \quad f(x, y) = 0 \quad \text{otherwise};$$

where, $i = 1, 2, \dots$ and $j = 1, 2, \dots$ independently. In analogy to Eq. (3.37), we now have for the distribution function the formula:

$$F(x, y) = \sum_{x_i \leq x} \sum_{y_j \leq y} f(x_i, y_j).$$

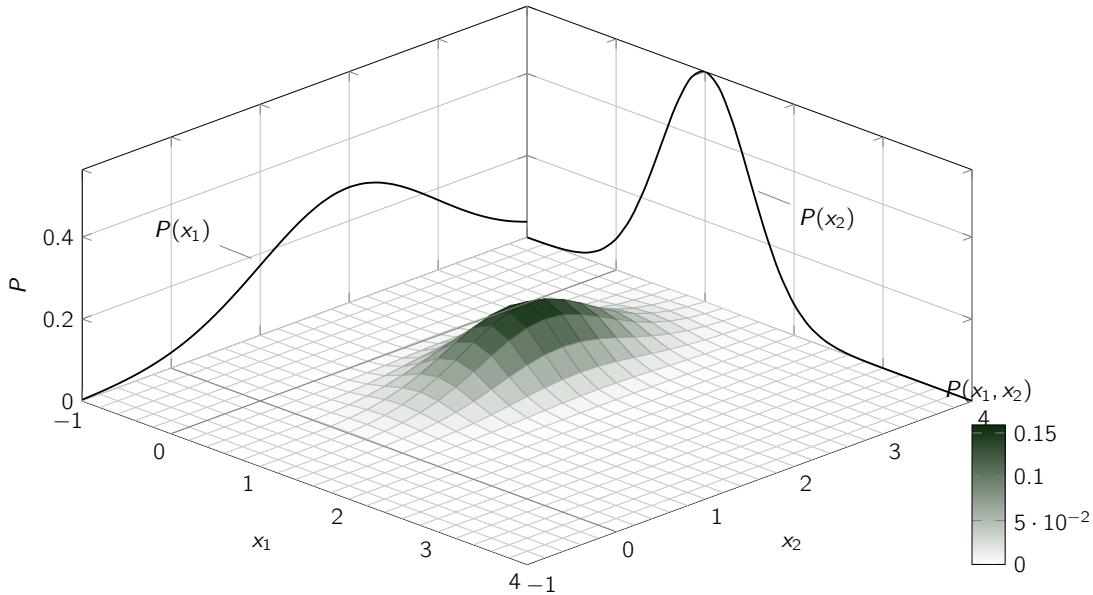


Figure 3.9: Many samples from a bivariate normal distribution. The marginal distributions are shown on the z-axis. The marginal distribution of X is also approximated by creating a histogram of the X coordinates without consideration of the Y coordinates.

Instead of Eq. (3.39), we now have the condition:

$$\sum_i \sum_j f(x_i, y_j) = 1.$$

3.9.2 Continuous Two-Dimensional Distribution

In analogy to the case of a single random variable, we call (X, Y) and its distribution **continuous** if the corresponding distribution function $F(x, y)$ can be given by a double integral:

$$F(x, y) = \int_{-\infty}^y \int_{-\infty}^x f(x^*, y^*) dx^* dy^* \quad (3.65)$$

whose integrand f , called the **density** of (X, Y) , is non-negative everywhere, and is continuous, possibly except on finitely many curves.

From Eq. (3.65) we obtain the probability that (X, Y) assume any value in a rectangle (Fig. 523) given by the formula:

$$P(a_1 < X \leq b_1, a_2 < Y \leq b_2) = \int_{a_1}^{b_1} \int_{a_2}^{b_2} f(x, y) dx dy$$

3.9.3 Marginal Distributions of a Discrete Distribution

This is a rather natural idea, without counterpart for a single random variable.

It amounts to being interested only in one of the two variables in (X, Y) , say, X , and asking for its distribution, called the **marginal distribution** of X in (X, Y) . So we ask for the probability $P(X = x, Y \text{ arbitrary})$.

Since (X, Y) is discrete, so is X . We get its probability function, call it $f_1(x)$, from the probability function $f(x, y)$ of (X, Y) by summing over y :

$$f_1(x) = P(X = x, Y, \text{arbitrary}) = \sum_y f(x, y) \quad (3.66)$$

where we sum all the values of $f(x, y)$ that are not 0 for that x .

From Eq. (3.66) we see that the distribution function of the marginal distribution of X is

$$F_1(x) = P(X \leq x, Y, \text{arbitrary}) = \sum_{x^* \leq x} f_1(x^*).$$

Similarly, the probability function

$$f_2(y) = P(X, \text{arbitrary}, Y \equiv y) = \sum_x f(x, y)$$

determines the **marginal distribution** of Y in (X, Y) . Here we sum all the values of $f(x, y)$ that are not zero for the corresponding y . The distribution function of this marginal distribution is

$$F_2(y) = P(X, \text{arbitrary}, Y \equiv y) = \sum_{y^* \equiv y} f_2(y^*).$$

Exercise 3.20: Marginal Distributions of a Discrete Two-Dimensional Random Variable

In drawing 3 cards with replacement from a bridge deck let us consider

$$(X, Y) \quad \text{where } X = \text{Number of queens} \quad \text{and } Y = \text{Number of kings or aces.}$$

The deck has 52 cards. These include 4 queens, 4 kings, and 4 aces. Therefore, in a single trial a queen has probability:

$$\frac{4}{52} = \frac{1}{13}$$

and a king or ace:

$$\frac{8}{52} = \frac{2}{13}$$

This gives the probability function of (X, Y) as:

$$f(x, y) = \frac{3!}{x!y!(3-x-y)!} \left(\frac{1}{13}\right)^x \left(\frac{2}{13}\right)^y \left(\frac{10}{13}\right)^{3-x-y} \quad \text{where } (x + y \leq 3)$$

and $f(x, y) = 0$ otherwise.

3.9.4 Marginal Distributions of a Continuous Distribution

This is conceptually the same as for discrete distributions, with probability functions and sums replaced by densities and integrals. For a continuous random variable (X, Y) with density $f(x, y)$ we now have the **marginal distribution** of X in (X, Y) , defined by the distribution function

$$F_1(x) = P(X \leq x, -\infty < Y < \infty) = \int_{-\infty}^x f_1(x^*) dx^*$$

with the density f_1 of X obtained from $f(x, y)$ by integration over y ,

$$f_1(x) = \int_{-\infty}^{\infty} f(x, y) dy.$$

Interchanging the roles of X and Y , we obtain the **marginal distribution** of Y in (X, Y) with the distribution function

$$F_2(y) = P(-\infty < X < \infty, Y \leq y) = \int_{-\infty}^y f_2(y^*) dy^*$$

and density

$$f_2(y) = \int_{-\infty}^{\infty} f(x, y) dx.$$

3.9.5 Independence of Random Variables

X and Y in a, discrete or continuous, random variable (X, Y) are said to be **independent** if

$$F(x, y) = F_1(x)F_2(y)$$

holds for all (x, y) . Otherwise these random variables are said to be **dependent**. Necessary and sufficient for independence is

$$f(x, y) = f_1(x)f_2(y)$$

for all x and y . Here the f 's are the above probability functions if (X, Y) is discrete or those densities if (X, Y) is continuous.

Exercise 3.21: Independence and Dependence

In tossing a 50 cent and a 20 cent coin, with X being the number of heads on the 50 cent, and Y number of heads on the 20 cent, we may assume the values 0 or 1 and are independent.

Extension of Independence to n -Dimensional Random Variables. This will be needed throughout Chapter 4. The distribution of such a random variable $\mathbf{X} = (X_1, \dots, X_n)$ is determined by a **distribution function** of the form

$$F(x_1, \dots, x_n) = P(X_1 \leq x_1, \dots, X_n \leq x_n).$$

The random variables X_1, \dots, X_n are said to be **independent** if

$$F(x_1, \dots, x_n) = F_1(x_1)F_2(x_2) \cdots F_n(x_n)$$

for all (x_1, \dots, x_n) . Here $F_j(x_j)$ is the distribution function of the marginal distribution of X_j in \mathbf{X} , that is,

$$F_j(x_j) = P(X_j \leq x_j, X_k \text{ arbitrary}, k = 1, \dots, n, k \neq j).$$

Otherwise these random variables are said to be **dependent**.

3.9.6 Functions of Random Variables

When $n = 2$, we write $X_1 = X$, $X_2 = Y$, $x_1 = x$, $x_2 = y$. Taking a non-constant continuous function $g(x, y)$ defined for all x, y , we obtain a random variable $Z = g(X, Y)$.

For example, if we roll two (2) dice and X and Y are the numbers the dice turn up in a trial, then $Z = X + Y$ is the sum of those two (2) numbers.

In the case of a **discrete** random variable (X, Y) we may obtain the probability function $f(z)$ of $Z = g(X, Y)$ by summing all $f(x, y)$ for which $g(x, y)$ equals the value of z considered; thus

$$f(z) = P(Z = z) = \sum_{g(x,y)=z} f(x, y).$$

Hence the distribution function of Z is

$$F(z) = P(Z \leq z) = \sum_{g(x,y) \leq z} f(x, y),$$

where we sum all values of $f(x, y)$ for which $g(x, y) \leq z$.

In the case of a **continuous** random variable (X, Y) we similarly have

$$F(z) = P(Z \leq z) = \iint_{g(x,y) \leq z} f(x, y) \, dx \, dy$$

where for each z we integrate the density $f(x, y)$ of (X, Y) over the region $g(x, y) \leq z$ in the xy -plane, the boundary curve of this region being $g(x, y) = z$.

3.9.7 Addition of Means

The number

$$E(g(X, Y)) = \begin{cases} \sum_x \sum_y g(x, y) f(x, y) & \text{where } X, Y \text{ are discrete} \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) f(x, y) \, dx \, dy & \text{where } X, Y \text{ are continuous} \end{cases} \quad (3.67)$$

is called the **mathematical expectation** or, briefly, the **expectation of** $g(X, Y)$. Here it is assumed that the double series converges absolutely and the integral of $|g(x, y)|/(x, y)$ over the y -plane meaning it is finite. exists⁴⁴. Since summation and integration are linear processes, we have from Eq. (3.67):

$$E(ag(X, Y) + bh(X, Y)) = aE(g(X, Y)) + bE(h(X, Y))$$

An important special case is

$$E(X + Y) = E(X) + E(Y),$$

and by induction we have the following result.

Theory 3.16: Addition of Means

The mean (expectation) of a sum of random variables equals the sum of the means (expectations), that is,

$$E(X_1 + X_2 + \dots + X_n) = E(X_1) + E(X_2) + \dots + E(X_n).$$

We can also deduce the following statement:

Theory 3.17: Multiplication of Means

The mean (expectation) of the product of independent random variables equals the product of the means (expectations), that is,

$$E(X_1 X_2 \dots X_n) = E(X_1) E(X_2) \dots E(X_n).$$

and in the continuous case the proof of the relation is similar⁴⁵.

⁴⁵This is left as an exercise to the reader.

3.9.8 Addition of Variances

A final matter to cover is how we can sum up variances. Similar to before, let $Z = X + Y$ and denote the mean and variance of Z by μ and σ^2 .

Then we first have:

$$\sigma^2 = E([Z - \mu]^2) = E(Z^2) - [E(Z)]^2$$

From (24) we see that the first term on the right equals

$$E(Z^2) = E(X^2 + 2XY + Y^2) = E(X^2) + 2E(XY) + E(Y^2).$$

For the second term on the right we obtain from Theorem 1

$$[E(Z)]^2 = [E(X) + E(Y)]^2 = [E(X)]^2 + 2E(X)E(Y) + [E(Y)]^2$$

By substituting these expressions into the formula for σ^2 we have

$$\begin{aligned} \sigma^2 &= E(X^2) - [E(X)]^2 + E(Y^2) - [E(Y)]^2 \\ &\quad + 2[E(XY) - E(X)E(Y)]. \end{aligned}$$

the expression in the first line on the right is the sum of the variances of X and Y , which we denote by σ_1^2 and σ_2^2 , respectively.

The quantity in the second line (except for the factor 2) is:

$$\sigma_{XY} = E(XY) - E(X)E(Y), \quad (3.68)$$

and is called the **covariance** of X and Y . Consequently, our result is

$$\sigma^2 = \sigma_1^2 + \sigma_2^2 + 2\sigma_{XY}.$$

If X and Y are **independent**, then

$$E(XY) = E(X)E(Y);$$

hence $\sigma_{XY} = 0$, and

$$\sigma^2 = \sigma_1^2 + \sigma_2^2$$

Extension to more than two variables gives the basic

Theory 3.18: Addition of Variances

The variance of the sum of independent random variables equals the sum of the variances of these variables.

Chapter 4

Statistical Methods

Table of Contents

4.1	Introduction	103
4.2	Point Estimation of Parameters	107
4.3	Confidence Intervals	111
4.4	Testing of Hypotheses and Making Decisions	116
4.5	Goodness of Fit	121
4.6	Regression and Correlation	124
4.7	Bayesian Statistics	133

4.1 Introduction

Statistical¹ methods consists of a wide range of tools for designing and evaluating random experiments to obtain information about practical problems:

such as exploring the relation between iron content and density of iron ore, the quality of raw material or manufactured products, the efficiency of air-conditioning systems, the performance of certain cars, the effect of advertising, the reactions of consumers to a new product, etc.

¹The word is derived from New Latin *statistica* or *statisticus* ("of the state")

Therefore, it is an important topic for any engineer as **Random variables** occur more frequently in engineering² than one would think. For example, properties of mass-produced articles³ always exhibit **random variation**, due to small⁴ differences in raw material or manufacturing processes.

²and of course elsewhere.

³such as screws, light bulbs, electric machines, etc.

Therefore, the diameter of screws is a random variable X and we have **non-defective screws**, with diameter between given tolerance limits, and **defective screws**, with diameter outside those limits.

⁴often uncontrollable

We can ask for the distribution of X , for the percentage of defective screws to be expected, and for necessary improvements of the production process.

Samples are selected from populations:

20 screws from 1000 screws, 100 of 5000 voters, 8 behaviours in a wildlife observation.

⁵It would be inconceivable for a company who produces over a billion light bulbs to test all their products. That is why we have return policies.

⁶of being drawn when we sample.

as inspecting the entire sample, would be expensive, time-consuming, impossible or even senseless.⁵

To obtain a meaningful sense of information, samples must be **random selections**. Each of the 1000 screws must have the same chance of being sampled,⁶ at least approximately. Only then will the sample mean:

$$\bar{x} = \frac{1}{20} (x_1 + \cdots + x_{20}) \quad \text{where} \quad n = 20,$$

will be a **good approximation** of the population mean μ , and the accuracy of the approximation will generally improve with increasing n , as we shall see.

This is also applicable to other statistical quantities such as standard deviation, variance, etc.

Independent sample values will be obtained in experiments with an infinite sample space S certainly for the **normal distribution**. This is also true in sampling with replacement. It is approximately true in drawing **small samples** from a large finite population.⁷ However, if we sample without replacement from a small population, the effect of dependence of sample values may be considerable.

⁷for instance, 5 or 10 of 1000 items.

Random numbers help in obtaining samples that are in fact random selections. This is sometimes not easy to accomplish as there are numerous subtle factors which can bias sampling.⁸ Random numbers can be obtained from a **random number generator**

It is important to state that the numbers generated by a computer are **NOT** truly random, as are calculated by a tricky formula that produces numbers that do have practically all the essential features of true randomness. Because these numbers eventually repeat, they must not be used in cryptography, for example, where true randomness is required.

Information: Generating Random Numbers

To select a sample of size $n = 10$ from 80 given ball bearings, we number the bearings from 1 to 80. We then let the generator randomly produce 10 of the integers from 1 to 80 and include the bearings with the numbers obtained in our sample, for example,

44 55 57 03 61 51 68 22 34 77

or whichever number pops up in your head.⁹

Representing and processing data were considered in the previous chapter in connection with **frequency distributions**. These are the **empirical counterparts** of probability distributions and helped motivating axioms and properties in probability theory. The new aspect in this chapter is **randomness**:

i.e., the data are samples selected **randomly** from a population.

⁸Such as by personal interviews, by poorly working machines, by the choice of non-typical observation conditions, etc.

⁹Of course in a professional setting you can't just write numbers like that as there is also a pattern when we make successive random numbers. Before the prevalence of computers there used to be books containing random numbers which people consulted.

Accordingly, we can already use the plots we have used in probability, such as stem-and-leaf plots, box plots, and histograms.

In this chapter, the mean \bar{x} we defined previously, will now be referred as **sample mean**.

$$\bar{x} = \frac{1}{n} \sum_{j=1}^n x_j = \frac{1}{n} (x_1 + x_2 + \dots + x_n) . \quad (4.1)$$

We call n the **sample size**, and similar to mean, the variance s^2 is called the **sample variance**:

$$s^2 = \frac{1}{n-1} \sum_{j=1}^n (x_j - \bar{x})^2 = \frac{1}{n-1} [(x_1 - \bar{x})^2 + \dots + (x_n - \bar{x})^2] , \quad (4.2)$$

and its positive square root, s is the **sample standard deviation**.

\bar{x}, s^2, s are called **sample parameters** of a dataset.

Information: Optimal Read: John Snow and the Founding of Epidemiology

John Snow (1813–1858) was an English physician and a leader in the development of anaesthesia and medical hygiene. He is considered one of the founders of modern epidemiology and early germ theory, in part because of his work in tracing the source of a cholera outbreak in London's Soho, which he identified as a particular public water pump. Snow was a skeptic of the then-dominant miasma theory stating diseases such as cholera and bubonic plague were caused by pollution or a noxious form of "bad air". The germ theory of disease had not yet been developed, so Snow did not understand the mechanism by which the disease was transmitted. His observation of the evidence led him to discount the theory of foul air. He first published his theory in an 1849 essay, *On the Mode of Communication of Cholera*, followed by a more detailed treatise in 1855 incorporating the results of his investigation of the role of the water supply in the Soho epidemic of 1854.

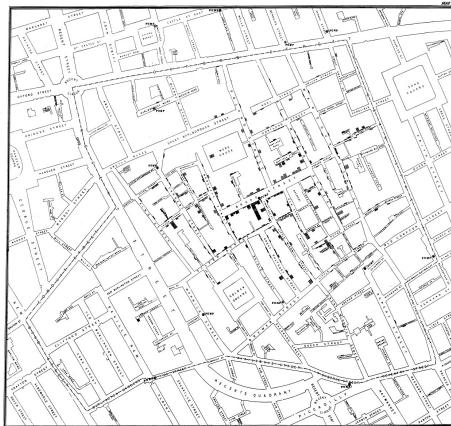


Figure 4.1: Original map by John Snow showing the clusters of cholera cases in the London epidemic of 1854, drawn and lithographed by Charles Cheffins.

By talking to local residents, he identified the source of the outbreak as the public water pump on Broad Street (now Broadwick Street). Although Snow's chemical and microscope examination of a water sample from the Broad Street pump did not conclusively prove its danger, his studies of the pattern of the disease were convincing enough to persuade the local council to disable the well pump by removing its handle.

This action has been commonly credited as ending the outbreak, but Snow observed that the epidemic may have already been in rapid decline.

4.2 Point Estimation of Parameters

Before we dive deep into statistics, let's spend some time to learn the most basic practical tasks in statistics and corresponding statistical methods to accomplish them. The first is point **estimation of parameters**, that is, of **quantities** appearing in distributions:

such as p in the binomial distribution and μ and σ in the normal distribution.

A **point estimate** of a parameter is a number,¹⁰ which is computed from a given sample and serves as an **approximation of the unknown exact value** of the parameter of the population. An interval estimate is an interval¹¹ obtained from a sample. Think of it as a value which is a sensible guess for that parameter.

¹⁰which is a point on the real line.

¹¹also known as confidence interval.

Estimation of parameters is of great practical importance in many applications.

As an approximation¹² of the mean of a population we may take the mean \bar{x} of a corresponding sample. This gives the estimate $\hat{\mu} = \bar{x}$ for μ , that is,

$$\hat{\mu} = \bar{x} = \frac{1}{n} (x_1 + \dots + x_n), \quad (4.3)$$

¹²To describe something which is an approximation or an educated guess, we use hat (i.e., \hat{x}) notation. This is applicable for fields in statistics, machine learning or data science.

where n is the sample size. Similarly, an estimate $\hat{\sigma}^2$ for the variance of a population is the variance s^2 of a corresponding sample, that is:

$$\hat{\sigma}^2 = s^2 = \frac{1}{n-1} \sum_{j=1}^n (x_j - \bar{x})^2. \quad (4.4)$$

As can be seen, both Eq. (4.3) and Eq. (4.4) are estimates¹³ of parameters for distributions in which μ or σ^2 appear explicitly as parameters, such as the normal and Poisson distributions.

An estimator is not expected to estimate the population parameter without error. We do not expect \bar{x} to estimate μ exactly, but we certainly hope that it is not far off.

¹³There are numerous practical examples for parameter estimation, such as using political polls to estimate voter turnout or the proportion of voters who will vote for a particular candidate, estimating the proportion of consumers interested in a product before massive production, or estimating the success rate of a product

For the binomial distribution, $p = \mu/n$. From Eq. (4.3) we obtain for p the estimate:

$$\hat{p} = \frac{\bar{x}}{n}. \quad (4.5)$$

It is important to mention Eq. (4.3) is a special case of the so-called **method of moments**. Here, the parameters to be estimated are expressed in terms of the moments of the distribution. In the resulting formulas, those moments of the distribution are replaced by the corresponding moments of the sample, which gives the estimates. Here the k^{th} moment of a sample x_1, \dots, x_n is:

$$m_k = \frac{1}{n} \sum_{j=1}^n x_j^k. \quad (4.6)$$

4.2.1 Maximum Likelihood Method

Another method for obtaining estimates is the so-called **maximum likelihood method** conceived by R. A. Fisher.¹⁴ To explain it, we consider a discrete (or continuous) random variable X whose probability function (or density) $f(x)$ depends on a single parameter θ . We take a corresponding sample of n independent values x_1, \dots, x_n . Then in the discrete case the probability given a sample of size n consists precisely of those n values is

$$L(x_1, x_2; \theta_1) = f(x_1) f(x_2) \cdots f(x_n). \quad (4.7)$$

In the continuous case the probability that the sample consists of values in the small intervals $x_j \leq x \leq x_j + \Delta x$ where ($j = 1, 2, \dots, n$) is

$$f(x_1) \Delta x f(x_2) \Delta x \cdots f(x_n) \Delta x = L(\Delta x)^n, \quad (4.8)$$

as $f(x_j)$ depends on θ , the function L in Eq. (4.8) given by Eq. (4.7) depends on x_1, \dots, x_n and θ .

We imagine x_1, \dots, x_n to be given and **fixed**.

Then L is a function of θ , which is called the **likelihood function**. The basic idea of the maximum likelihood method is quite simple, as follows.

We choose an approximation for the unknown value of θ for which L is **as large as possible**.

If L is a differentiable function of θ , a necessary condition for L to have a maximum in an interval¹⁵ is

$$\frac{\partial L}{\partial \theta} = 0 \quad (4.9)$$

A solution of Eq. (4.9) depending on x_1, \dots, x_n is called a **maximum likelihood estimate** for θ .

¹⁶It is worth noting that if the function is NOT derivable in some points of the domain, critical point is not a maximum, and the global maximum occurs on

the boundary of the domain, it may be required to take the 2nd derivative [50]. However, if you know that the function you are trying to estimate the parameter has one (1) critical point, a single derivative would suffice.

[50]. However, if you know that the function you are trying to estimate the parameter has one (1) critical point, a single derivative would suffice.

$$\frac{\partial \ln L}{\partial \theta} = 0 \quad (4.10)$$

as $f(x_j) > 0$, a maximum of L is in general positive, and $\ln L$ is a monotone increasing function of L . This often simplifies calculations due to constant multiplication of the same function n times to get L . The use of \ln turns the exponentiation parameter to a multiplication parameter.

Several Parameters

If the distribution of X involves r parameters $\theta_1, \dots, \theta_r$, then instead of Eq. (4.9) we have the r conditions $\partial \ln L / \partial \theta_1, \dots, \partial \ln L / \partial \theta_r = 0$, and instead of Eq. (4.10) we have:

$$\frac{\partial \ln L}{\partial \theta_1} = 0, \dots, \frac{\partial \ln L}{\partial \theta_r} = 0. \quad (4.11)$$

Exercise 4.1: Maximum Likelihood of Gaussian Distribution

Find maximum likelihood estimates for $\theta_1 = \mu$ and $\theta_2 = \sigma$ in the case of the normal distribution.

Solution

We obtain the likelihood function:

$$L = \left(\frac{1}{\sqrt{2\pi}} \right)^n \left(\frac{1}{\sigma} \right)^n e^{-h}$$

$$\text{where } h = \frac{1}{2\sigma^2} \sum_{j=1}^n (x_j - \mu)^2.$$

Taking logarithms, we have

$$\ln L = -n \ln \sqrt{2\pi} - n \ln \sigma - h.$$

The first equation for the parameters is $\frac{\partial \ln L}{\partial \mu} = 0$, written out:

$$\frac{\partial \ln L}{\partial \mu} = -\frac{\partial h}{\partial \mu} = \frac{1}{\sigma^2} \sum_{j=1}^n (x_j - \mu) = 0,$$

$$\text{therefore } \sum_{j=1}^n x_j - n\mu = 0.$$

The solution is the desired estimate $\hat{\mu}$ for μ : we find

$$\hat{\mu} = \frac{1}{n} \sum_{j=1}^n x_j = \bar{x}.$$

The second equation for the parameter is $\frac{\partial \ln L}{\partial \sigma} = 0$, written out

$$\frac{\partial \ln L}{\partial \sigma} = -\frac{n}{\sigma} - \frac{\partial h}{\partial \sigma} = -\frac{1}{\sigma} + \frac{1}{\sigma^3} \sum_{j=1}^n (x_j - \mu)^2 = 0.$$

Replacing μ by $\hat{\mu}$ and solving for σ^2 , we obtain the estimate:

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{j=1}^n (x_j - \bar{x})^2 \blacksquare$$

Exercise 4.2: Maximum Likelihood of Poisson Distribution

Consider a Poisson distribution with probability mass function:

$$f(x|\mu) = \frac{e^{-\mu}\mu^x}{x!} \quad \text{where } x = 0, 1, 2, \dots$$

Suppose that a random sample x_1, x_2, \dots, x_n is taken from the distribution. What is the maximum likelihood estimate of μ ?

Solution

The likelihood function is

$$L(x_1, x_2, \dots, x_n; \mu) = \prod_{i=1}^n f(x_i|\mu) = \frac{e^{-n\mu} \sum_{i=1}^n x_i}{\prod_{i=1}^n x_i!}.$$

Now consider its logarithmic representation:

$$\ln L(x_1, x_2, \dots, x_n; \mu) = -n\mu + \sum_{i=1}^n x_i \ln \mu - \ln \prod_{i=1}^n x_i!$$

And taking its partial derivative to the parameter gives:

$$\frac{\partial \ln L(x_1, x_2, \dots, x_n; \mu)}{\partial \mu} = -n + \sum_{i=1}^n \frac{x_i}{\mu}.$$

Solving for $\hat{\mu}$, the maximum likelihood estimator, involves setting the derivative to zero and solving for the parameter. Therefore,

$$\hat{\mu} = \sum_{i=1}^n \frac{x_i}{n} = \bar{x}$$

If you were to test it, the second derivative of the log-likelihood function is negative, which implies that the solution above indeed is a maximum. As μ is the mean of the Poisson distribution, the sample average would certainly seem like a reasonable estimator \blacksquare .

Applications

There are numerous applications for **Maximum Likelihood Estimation**, for example when dealing with design and control systems in engineering [51], it is used to estimate system parameters based on noisy measurements. In the financial sector [52], it can help estimate parameters of models like Black-Scholes-Merton [53], which describe the dynamics of financial derivatives, and in Biology, it is used in genetic mapping [54] and genome-wide association studies [55].

Exercise 4.3: For Science

Suppose ten (10) rats are used in a biomedical study where they are injected with cancer cells and then given a cancer drug that is designed to increase their survival rate. The survival times, in months, are:

14 17 27 18 12 8 22 13 19 12

Assume exponential distribution applies which is given as:

$$f(x; \beta) = \begin{cases} \frac{1}{\beta} \exp \frac{x}{\beta}, & x > 0, \\ 0, & \text{elsewhere.} \end{cases}$$

Give a maximum likelihood estimate of the mean survival time.

Solution

We know that the probability density function for the exponential random variable X . Therefore, the log-likelihood

function for the data, given $n = 10$, is:

$$\ln L(x_1, x_2, \dots, x_{10}; \beta) = -10 \ln \beta - \frac{1}{\beta} \sum_{i=1}^{10} x_i.$$

Setting

$$\frac{\partial \ln L}{\partial \beta} = -\frac{10}{\beta} + \frac{1}{\beta^2} \sum_{i=1}^{10} x_i = 0 \quad \text{which means}$$

$$\hat{\beta} = \frac{1}{10} \sum_{i=1}^{10} x_i = \bar{x} = 16.2 \blacksquare$$

Evaluating the second derivative of the log-likelihood function at the value $\hat{\beta}$ above gives a negative value. As a result, the estimator of the parameter β , the population mean, is the sample average \bar{x} .

Exercise 4.4: Sampling the Population

It is known that a sample consisting of the values:

12 11.2 13.5 12.3 13.8 11.9

comes from a population with the density function:

$$f(x; \theta) = \begin{cases} \frac{\theta}{\theta+1}, & x > 1, \\ 0, & \text{elsewhere,} \end{cases}$$

where $\theta > 0$. Find the maximum likelihood estimate of θ .

Solution

The likelihood function of n observations from this population can be written as:

$$L(x_1, x_2, \dots, x_{10}; \theta) = \prod_{i=1}^n \frac{\theta}{x_i^{\theta+1}} = \frac{\theta^n}{(\prod_{i=1}^n x_i)^{\theta+1}},$$

which implies that

$$\ln L(x_1, x_2, \dots, x_{10}; \theta) = n \ln \theta - (\theta + 1) \sum_{i=1}^n \ln x_i.$$

Setting $0 = \partial \ln L / \partial \theta = n/\theta - \sum_{i=1}^n \ln(x_i)$ results in

$$\hat{\theta} = \frac{n}{\sum_{i=1}^n \ln x_i} = 0.3970 \blacksquare$$

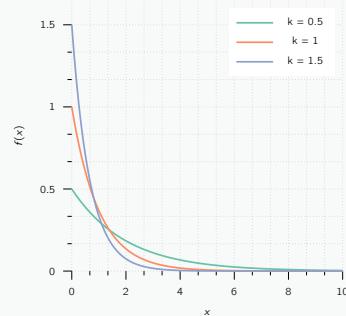
Since the second derivative of L is $-n/\theta^2$, which is always negative, the likelihood function does achieve its maximum value at $\hat{\theta}$.

Information: Exponential Distribution

It is the probability distribution of the distance between events in a Poisson point process, i.e., a process in which events occur continuously and independently at a constant average rate, such as time between production errors, or length along a roll of fabric in the weaving manufacturing process. It also has the feature of being memoryless:¹⁷

$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0. \end{cases}$$

Here $\lambda > 0$ is the parameter of the distribution, often called the **rate parameter**. The exponential distribution occurs naturally when describing the lengths of the inter-arrival times in a homogeneous Poisson process. Such as the time until a radioactive particle decays, or the time between clicks of a Geiger counter, or the time between receiving one telephone call and the next.



¹⁷i.e., previous failures or elapsed time does not affect future trials.

4.3 Confidence Intervals

Confidence intervals¹⁸ for an unknown parameter θ of some distribution (e.g., $\theta = \mu$) are intervals $\theta_1 \leq \theta \leq \theta_2$ which contain θ , **NOT** with certainty but with a **high probability** γ , which we can choose where 95% and 99% are popular choices. Such an interval is calculated from a sample. $\gamma = 95\%$ means probability $1 - \gamma = 5\% = 1/20$ of being wrong. Instead of writing $\theta_1 \leq \theta \leq \theta_2$, we denote this more **distinctly** by writing:

$$\text{CONF}_\gamma \{ \theta_1 \leq \theta \leq \theta_2 \} \quad (4.12)$$



¹⁸Established by Jerzy Neyman. He proposed and studied randomised experiments in 1923. Furthermore, his paper *On the Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposive Selection* [56], given at the Royal Statistical Society in 1934, was the groundbreaking event leading to modern scientific sampling. He introduced the confidence interval in his paper in 1937 [57]. Another noted contribution is the Neyman-Pearson lemma, the basis of hypothesis testing [58].

Such a special symbol, CONF, seems worthwhile to avoid the misunderstanding that θ **must** lie between θ_1 and θ_2 .

γ is called the **confidence level**, and θ_1 and θ_2 are called the **lower** and **upper confidence limits**, respectively and **depend** on the γ value. The larger we **choose** γ , the smaller is the error probability $1 - \gamma$, but the longer is the confidence interval.

As $\gamma \rightarrow 1$, the interval goes to infinity.

The choice of γ depends on the type of application.

In taking no umbrella, a 5% chance of getting wet is **NOT** a problem. In a medical decision of life or death, a 5% chance of being wrong may be too large and a 1% chance of being wrong ($\gamma = 99\%$) may be more desirable.

Confidence intervals are more valuable than point estimates. We can take the midpoint of Eq. (4.12) as an approximation of θ and half the length of Eq. (4.12) as an **error bound**.¹⁹

θ_1 and θ_2 in Eq. (4.12) are calculated from a sample x_1, \dots, x_n . These are n observations of a random variable X . Now comes a **trick**.

We regard x_1, \dots, x_n as single observations of n random variables X_1, \dots, X_n .²⁰ Then $\theta_1 = \theta_1(x_1, \dots, x_n)$ and $\theta_2 = \theta_2(x_1, \dots, x_n)$ in Eq. (4.12) are observed values of two (2) random variables $\Theta_1 = \Theta_1(X_1, \dots, X_n)$ and $\Theta_2 = \Theta_2(X_1, \dots, X_n)$. The condition Eq. (4.12) involving γ can now be written:²¹

$$P(\Theta_1 \leq \theta \leq \Theta_2) = \gamma. \quad (4.13)$$

Let us see what all this means in concrete practical cases.

¹⁹not in the strict sense of numerical means, but except for an error whose probability we know.

²⁰with the same distribution as X

²¹As an example, to say that there is 95% confidence is shorthand for "95% of all possible samples of a given size from this population will result in an interval that captures the unknown parameter."

In each case in this section we shall first state the steps of obtaining a confidence interval in the form of a table, then consider a typical example, and finally justify those steps theoretically.

For Mean with Known Variance in Normal Distribution

The method of tackling is this problem is as follows:

²²95%, 99%, depending on the application.

1. Choose a confidence level for γ ²².
2. Determine the corresponding c :

γ	0.90	0.95	0.99	0.999
c	1.645	1.960	2.576	3.291

Table 4.1: Useful c values based on a given confidence (γ) value.

3. Calculate the mean \bar{x} of the sample x_1, \dots, x_n .
4. Calculate $k = c\sigma/\sqrt{n}$. The confidence interval for μ is

$$\text{CONF}_\gamma \{\bar{x} - k \leq \mu \leq \bar{x} + k\}. \quad (4.14)$$

Exercise 4.5: Confidence Interval for Mean with known Variance in Normal Distribution

Determine 95% confidence interval for the mean of a normal distribution with variance $\sigma^2 = 9$, using a sample of $n = 100$ values with mean $\bar{x} = 5$.

Solution

1. First we define γ as 0.95 based on the 95% confidence.
2. Then looking at our reference table find the corresponding c which equals 1.960.
3. $\bar{x} = 5$ is given.

4. We need:

$$k = c \frac{\sigma}{\sqrt{n}} = 1.960 \frac{3}{\sqrt{100}} = 0.588$$

Therefore

$$\bar{x} - k = 4.412 \quad \text{and} \quad \bar{x} + k = 5.588$$

and the confidence interval is:

$$\text{CONF}_{0.95} \{4.412 \leq \mu \leq 5.588\} \blacksquare$$

Theory 4.19: Sum of Independent Normal Random Variables

Let X_1, \dots, X_n be independent normal random variables each of which has mean μ and variance σ^2 .

Then the following holds:

- a. The sum $X_1 + \dots + X_n$ is normal with mean $n\mu$ and variance $n\sigma^2$.
- b. The following random variable \bar{X} is normal with mean μ and variance σ^2/n .

$$\bar{X} = \frac{1}{n} (X_1 + \dots + X_n) \quad (4.15)$$

- c. The following random variable Z is normal with mean 0 and variance 1.

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$$

Exercise 4.6: Sample Size Needed for a Confidence Interval of Prescribed Length

How large must n be in the Example **Confidence Interval for mean with known variance in Normal Distribution** to obtain a 95% confidence interval of length $L = 0.4$?

Solution

The interval in Example **Confidence Interval for mean with known variance in Normal Distribution** has the length:

$$L = 2k = 2c\sigma/\sqrt{n}.$$

Solving for n , we obtain

$$n = \left(\frac{2c\sigma}{L} \right)^2$$

In the present case the answer is:

$$n = \left(\frac{2 \times 1.96 \times 3}{0.4} \right)^2 \approx 870 \blacksquare$$

For Mean of the Normal Distribution with Unknown Variance

For practical applications, σ^2 is frequently **unknown**. Then the method described previously does **NOT** help and the whole theory changes, although the steps of determining a confidence interval for μ remain quite similar.

We see that k differs from previous method, namely, the sample standard deviation s has taken the place of the unknown standard deviation σ of the population as it is now the variance we are trying to estimate, and c now depends on the sample size n and must be determined from **Table ??** to **Table ??**. That table lists values z for given values of the distribution function.

$$F(z) = K_m \int_{-\infty}^z \left(1 + \frac{u^2}{m} \right)^{-(m+1)/2} du \quad (4.16)$$

of the t -distribution. Here, $m = 1, 2, \dots$ is a parameter, called the **number of degrees of freedom** of the distribution.²³ In the present case, $m = n - 1$ where n is the number of sample we have to determine variance. The constant K_m is such that $F(\infty) = 1$. By integration it turns out that:²⁴

$$K_m = \frac{\Gamma\left(\frac{1}{2}m + \frac{1}{2}\right)}{\sqrt{m\pi}\Gamma\left(\frac{1}{2}m\right)},$$

where Γ is the gamma function.²⁵

The method of tackling is this problem is as follows:

1. Choose a confidence level γ .²⁶

2. Determine the solution c of the equation,

$$F(c) = \frac{1}{2}(1 + \gamma)$$

from the table of the t -distribution with $m = n - 1$ degrees of freedom

3. Compute the mean \bar{x} and the variance s^2 of the sample x_1, \dots, x_n .

4. Compute $k = cs/\sqrt{n}$. The confidence interval is:

$$\text{CONF}_\gamma\{\bar{x} - k \leq \mu \leq \bar{x} + k\}.$$

²³abbreviated d.f.

²⁴For most practical application a reference Table would suffice to determine the parameter.

²⁵Do not worry if these equations do not make sense as it is here for literature purposes.

²⁶95%, 99%, or the like.

²⁷which uses more information, namely, the known value of σ^2

This illustrates that 4.1²⁷ provides shorter confidence intervals than Table XX, which enforces the idea getting a shorter interval range by increasing the sample size.

Exercise 4.7: Confidence Interval for Mean of Normal Distribution with Unknown Variance

The five (5) independent measurements of flash point of Diesel oil (D-2) gave the values (in°F):

144 147 146 142 144

If we assume normality, determine a 99% confidence interval for the mean.

Solution

1. $\gamma = 0.99$ is required based on 99% confidence level.
2. $F(c) = \frac{1}{2}(1 + \gamma) = 0.99$ and looking at the reference table with $n - 1 = 4$ d.f., which gives $c = 4.60$.
3. Calculating the mean and the variance gives $\bar{x} = 144.6$ and $s = 3.8$,

4. $k = \sqrt{3.8} \times 4.60 / \sqrt{5} = 4.01$. Therefore the confidence interval is:

$$\text{CONF}_{0.99} \{140.5 \leq \mu \leq 148.7\} \blacksquare$$

If the variance σ^2 were known and equal to the sample variance s^2 , therefore $\sigma^2 = 3.8$, then the Reference Table would give:

$$k = \frac{c\sigma}{\sqrt{n}} = 2.576 \frac{\sqrt{3.8}}{\sqrt{3}} = 2.25$$

and

$$\text{CONF}_{0.99} \{142.35 \leq \mu \leq 146.85\} \blacksquare$$

We see that the present interval is almost twice as long as that with a known variance $\sigma^2 = 3.8$.



²⁸William Gosset (1876 – 1937) was an English statistician, chemist and brewer who served as Head Brewer of Guinness and Head Experimental Brewer of Guinness and was a pioneer of modern statistics. He published his results under the pen name student.

Theory 4.20: Student's t-Distribution

Let X_1, \dots, X_n be independent normal random variables with the same mean μ and the same variance σ^2 . Then the random variable:

$$T = \frac{\bar{X} - \mu}{S/\sqrt{n}} \quad (4.17)$$

has a t-distribution²⁸ with $n - 1$ degrees of freedom (d.f.). Here X_1, \dots, X_n is given by Eq. (4.15) and S is defined as:

$$S^2 = \frac{1}{n-1} \sum_{j=1}^n (x_j - \bar{x})^2. \quad (4.18)$$

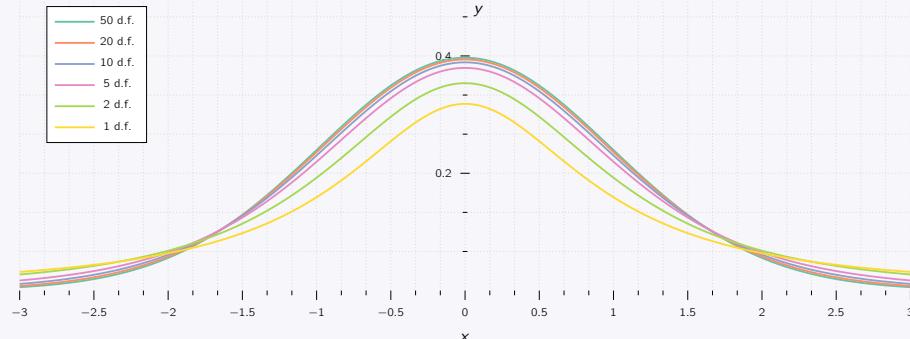


Figure 4.2: The student-t distribution with different degrees of freedom m .

Estimating the Variance of the Normal Distribution

The method for calculating the confidence interval is similar to the previous methods, with slight change in some steps which are as follows:

1. Choose a confidence level γ ²⁹
2. Determine solutions c_1 and c_2 of the equations:

$$F(c_1) = \frac{1}{2}(1 - \gamma) \quad \text{and} \quad F(c_2) = \frac{1}{2}(1 + \gamma).$$

where the necessary values are calculated from the table of the chi-square distribution with $n - 1$ degrees of freedom., given in **Table ??** to **Table ??**.

3. Calculate $(n - 1)s^2$, where s^2 is the variance of the sample x_1, \dots, x_n .
4. Calculate $k_1 = (n - 1)s^2/c_1$ and $k_2 = (n - 1)s^2/c_2$. The confidence interval is

$$\text{CONF}_\gamma\{k_2 \cong \sigma^2 \cong k_1\}. \quad (4.19)$$

Theory 4.21: Chi-Square Distribution

Under the assumptions in Theorem **Student's t-Distribution** the random variable:

$$Y = (n - 1) \frac{S^2}{\sigma^2}$$

with S^2 given by Eq. (4.18) has a chi-square distribution with $n - 1$ degrees of freedom.

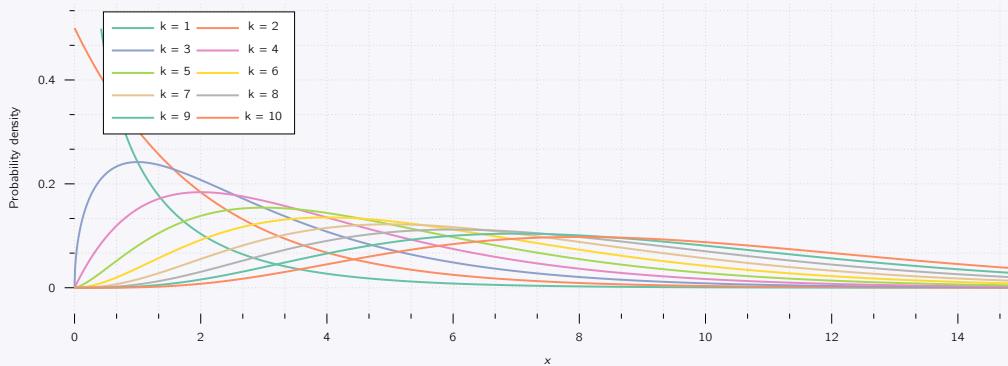


Figure 4.3: Chi-square distribution with different degrees of freedom.

The chi-squared distribution, which can be seen in Fig. 4.3 is used primarily in **hypothesis testing**, and to a lesser extent for confidence intervals for population variance when the underlying distribution is **normal**. Unlike more widely known distributions such as the normal distribution and the exponential distribution, the chi-squared distribution is not as often applied in the direct modelling of natural phenomena.

The primary reason for which the chi-squared distribution is extensively used in hypothesis testing is its relationship to the normal distribution. Many hypothesis tests use a test statistic, such as the **t-student**. For these hypothesis tests, as the sample size n increases, the sampling distribution of the test statistic approaches the normal distribution.³⁰ Because the test statistic (t) is asymptotically normally distributed, provided the sample size is sufficiently large, the distribution used for hypothesis testing may be approximated by a normal distribution.

So wherever a normal distribution could be used for a hypothesis test, a chi-squared distribution could be used.

²⁹as usual this can be 95%, 99%, or the like.

³⁰This is the result of the central limit theorem.



4.4 Testing of Hypotheses and Making Decisions

The ideas of confidence intervals and of tests³¹ are the two (2) most important ideas in modern statistics. In a statistical test we make inference from sample to population through testing a **hypothesis**, resulting from experience or observations, from a theory or a quality requirement, and so on.

In many cases the result of a test is used as a basis for a **decision**:

to buy, or not to buy a certain model of car, depending on a test of the fuel efficiency (km L^{-1}), or, to apply some medication, depending on a test of its effect; to proceed with a marketing strategy, depending on a test of consumer reactions, etc.

As with most abstract mathematical concepts, it is better to explain such a test in terms of a typical example and then introduce the corresponding standard notions of statistical testing.

³¹The modern development of tests are generally attributed to Egon Sharpe Pearson and Neymar whom was mentioned previously. Egon Sharpe was one of three children of Karl Pearson and Maria, and, like his father, a British statistician. He is known throughout the world as co-author of the Neyman-Pearson theory of testing statistical hypotheses, and responsible for many important contributions to problems of statistical inference and methodology, especially in the development and use of the likelihood ratio criterion.

³²also called **null hypothesis**.

³³or alternative hypothesis

Information: Test of a Hypothesis

Let's say we want to buy 100 coils of a certain kind of wire, provided we can verify the manufacturer's claim, the wire has a specific strength of $\mu = \mu_0 = 200 \text{ kN m kg}^{-1}$, or more.

This is a test of the hypothesis:³² $\mu = \mu_0 = 200$. We shall **NOT** buy the wire if the statistical tests shows that actually $\mu = \mu_1 < \mu_0$, the wire is weaker, the claim does **NOT** hold. μ_1 is called the **alternative** of the test.³³ We shall **accept** the hypothesis if the test suggests that it is true, except for a small error probability α , called the **significance level** of the test.

Otherwise we reject the hypothesis.

Hence α is the probability of rejecting a hypothesis although it is true. The choice of α is up to us, 5% and 1% are popular values.

For the test we need a sample. We randomly select 25 coils of the wire, cut a piece from each coil, and determine the breaking limit experimentally. Suppose that this sample of $n = 25$ values of the breaking limit has the mean $\bar{x} = 197 \text{ kN m kg}^{-1}$, which is somewhat less than the claim, and the standard deviation $s = 6 \text{ kN m kg}^{-1}$.

At this point we could only speculate when this difference $197 - 200 = -3$ is due to randomness, is a chance effect, or whether it is **significant**, due to the actual inferior quality of the wire. To continue beyond speculation requires probability theory, as follows.

We assume that the breaking limit is normally distributed. Then

$$T = \frac{\bar{X} - \mu_0}{S/\sqrt{n}}$$

with $\mu = \mu_0$ has a **t-distribution** with $n - 1$ degrees of freedom ($n - 1 = 24$ for our sample). Also $\bar{x} = 197$ and $s = 6$ are observed values of \bar{X} and S to be used later. We can now choose a significance level, say, $\alpha = 95\%$. From the Reference Table, we then obtain a critical value c such that $P(T \leq c) = \alpha = 5\%$. For $P(T \leq \bar{c}) = 1 - \alpha = 95\%$ the table gives $\bar{c} = 1.71$, so that $c = -\bar{c} = -1.71$ because of the symmetry of the distribution shown in Fig. 4.4.

We now reason as follows—this is the crucial idea of the test. If the hypothesis is true, we have a chance of only $\alpha (= 5\%)$ that we observe a value t of T (calculated from a sample) that will fall between $-\infty$ and -1.71 . Hence, if we nevertheless do observe such a t , we start that the hypothesis cannot be true and we reject it.

A simple calculation gives:

$$T = \frac{(197 - 200)}{6/\sqrt{25}} = -2.5,$$

as an observed value of T . Since $-2.5 < -1.71$, we reject the hypothesis, the manufacturer's claim, and accept the alternative result of $\mu = \mu_1 < 200$, which means the wire seems to be weaker than claimed ■

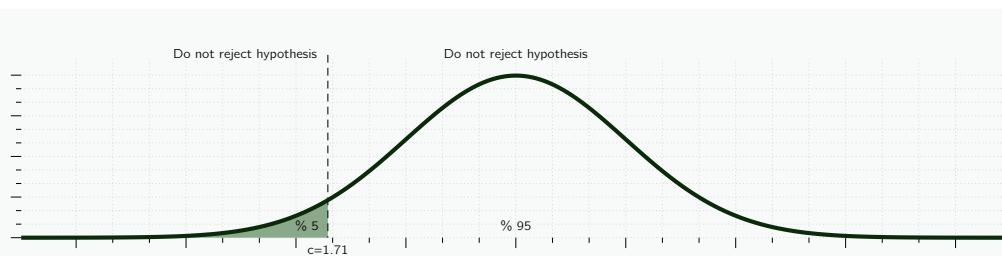


Figure 4.4: The t -distribution used in example. As can be seen, anything left of the critical line would tell us to reject the hypothesis, whereas if the t value lies on the RHS, then the test would tell us the null hypothesis is true.

This aforementioned example perfectly captures the **steps of a test**:

1. Formulate the **hypothesis** $\theta = \theta_0$ to be tested. In our previous example it is $\theta_0 = \mu_0$.
2. Formulate an **alternative** $\theta = \theta_1$, which in our example is $\theta_1 = \mu_1$.
3. Choose a **significance level** α with values such as 5%, 1%, or, 0.1%.
4. Use a random variable $\hat{\Theta} = g(X_1, \dots, X_n)$ whose distribution depends on the hypothesis and on the alternative, and this distribution is known in both cases.

Determine a critical value c from the distribution of $\hat{\Theta}$, assuming the hypothesis to be true. In the example, $\hat{\Theta} = T$, and c is, obtained from $P(T \leq c) = \alpha$.

5. Use a sample x_1, \dots, x_n to determine an observed value $\hat{\theta} = g(x_1, \dots, x_n)$ of $\hat{\Theta}$, where in our example it is t .
6. Accept or reject the hypothesis, depending on the size of $\hat{\theta}$ **relative** to c .

There are two (2) important facts require further discussion and careful attention.

1. The choice of an alternative. In the example, $\mu_1 < \mu_0$, but other applications may require $\mu_1 > \mu_0$ or $\mu_1 \neq \mu_0$ as not all application can fit the same criterion. Some applications may require an upper-bound, some lower-bound.
2. Addressing errors. We know that α , the significance level of the test, is the probability of reflecting a **true** hypothesis. And we shall discuss the probability β of accepting a **false** hypothesis.

One-Sided and Two-Sided Alternatives

Let θ be an **unknown parameter** in a distribution, and suppose we want to test the hypothesis $\theta = \theta_0$.

Then there are three (3) main kinds of alternatives, namely,

$$\theta > \theta_0 \quad (4.20)$$

$$\theta < \theta_0 \quad (4.21)$$

$$\theta \neq \theta_0 \quad (4.22)$$

Here Eq. (4.20), and Eq. (4.21) are **one-sided alternatives**, and Eq. (4.22) is a **two-sided alternative**.

³⁴or called the critical region We call rejection region³⁴ the region such that we reject the hypothesis if the observed value in the test falls in this region. In Eq. (4.20) the critical c lies to the right of θ_0 because so does the alternative. Hence the rejection region extends to the right. This is called a **right-sided test**. In Eq. (4.21) the critical c lies to the left of θ_0 (as in **Test of a Hypothesis**), the rejection region extends to the left, and we have a **left-sided test**. These are one-sided tests. In Eq. (4.22)

All three (3) kinds of alternatives occur in practical problems. For example, Eq. (4.20) may arise if θ_0 is the maximum tolerable inaccuracy of a voltmeter or some other instrument. Alternative Eq. (4.21) may occur in testing strength of material, as in **Test of a Hypothesis**. Finally, θ_0 in Eq. (4.22) may be the diameter of axle-shafts, and shafts that are too thin or too thick are equally undesirable, so that we have to watch for deviations in both directions.

4.4.1 Errors in Tests

Tests always involve **risks of making false decisions**:

I Rejecting a true hypothesis (Type I error)

■ α = Probability of making a Type I error.

II Accepting a false hypothesis (Type II error).

■ β = Probability of making a Type II error.

Clearly, we cannot avoid these errors.

No absolutely certain conclusions about populations can be drawn from samples.

But we show there are ways and means of choosing suitable levels of risks, that is, of values α and β . The choice of α depends on the nature of the problem.³⁵

³⁵e.g., a small risk $\alpha = 1\%$ is used if it is a matter of life or death.

Let us discuss this systematically for a test of a hypothesis $\theta = \theta_0$ against an alternative that is a single number θ_1 , for simplicity. We let $\theta_1 > \theta_0$, so that we have a **right-sided test**. For a left-sided or a two-sided test the discussion is quite similar.

We choose a critical $c > \theta_0$. From a given sample x_1, \dots, x_n we then compute a value:

$$\hat{\theta} = g(x_1, \dots, x_n)$$

with a suitable g .

the choice of g will be a main point of our further discussion; for instance, take $g = (x_1 + \dots + x_n) / n$ in the case in which θ is the mean.

If $\hat{\theta} > c$, we **reject the hypothesis**. If $\hat{\theta} \leq c$, we accept it. Here, the value $\hat{\theta}$ can be regarded as an observed value of the random variable

$$\hat{\Theta} = g(X_1, \dots, X_n)$$

because x_j may be regarded as an observed value of X_j where $j = 1, \dots, n$. In this test there are two (2) possibilities of making an error, as follows.

Type I Error The hypothesis is true but is rejected³⁶ because Θ assumes a value $\hat{\theta} > c$. Obviously, the probability of making such an error equals ³⁶hence the alternative is accepted.

$$P(\hat{\Theta} > c)_{\theta} = \theta_0 = \alpha. \quad (4.23)$$

α is called the **significance level** of the test, as mentioned before.

Type II Error The hypothesis is false but is accepted because $\hat{\Theta}$ assumes a value $\hat{\theta} \leq c$. The probability of making such an error is denoted by β ; Therefore:

$$P(\hat{\Theta} \leq c)_{\theta=\theta_1} = \beta. \quad (4.24)$$

$\eta = 1 - \beta$ is called the **power** of the test. Obviously, the power η is the probability of avoiding a Type II error. Formulas Eq. (4.23) and Eq. (4.24) show that both α and β depend on c , and

		Auxiliary Values	
		$\theta = \theta_0$	$\theta = \theta_1$
Accepted	$\theta = \theta_0$	True Decision $P = 1 - \alpha$	Type II Error $P = \beta$
	$\theta = \theta_1$	Type I Error $P = \alpha$	True Decision $P = 1 - \beta$

Table 4.2: Type I and Type II errors in testing a hypothesis $\theta = \theta_0$ against an alternative $\theta = \theta_1$.

we would like to choose c so these probabilities of making errors are as small as possible. But the important Fig. 4.5 shows that these are conflicting requirements because to let α decrease we must shift c to the right, but then β increases. In practice we first choose α (5%, sometimes 1%), then determine c , and finally compute β . If β is large so that the power $\eta = 1 - \beta$ is small, we should repeat the test, choosing a larger sample, for reasons that will appear shortly. If the alternative is

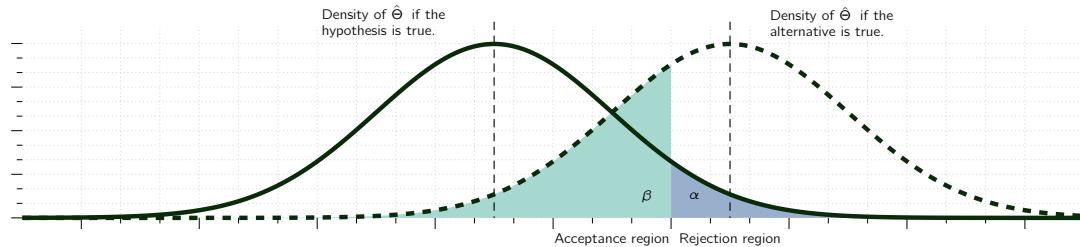


Figure 4.5: Illustration of Type I and II errors in testing a hypothesis $\theta = \theta_0$ against an alternative $\theta = \theta_1$.

NOT a single number but is of the form Eq. (4.20)-Eq. (4.22), then β becomes a function of θ . This function $\beta(\theta)$ is called the operating characteristic (OC) of the test and its curve the OC curve. Clearly, in this case $\eta = 1 - \beta$ also depends on θ . This function $\eta(\theta)$ is called the **power function** of the test.

Of course, from a test that leads to the acceptance of a certain hypothesis θ_0 , it does **NOT** follow that this is the only possible hypothesis or the best possible hypothesis. Hence the terms “not reject” or “fail to reject” are perhaps better than the term “accept”.

Exercise 4.8: Test for the Mean of the Normal Distribution with Known Variance

Let X be a normal random variable with variance $\sigma^2 = 9$. Using a sample of size $n = 10$ with mean \bar{X} , test the hypothesis $\mu = \mu_0 = 24$ against the three (3) kinds of alternatives, namely,

- (a) $\mu > \mu_0$ (b) $\mu < \mu_0$ (c) $\mu \neq \mu_0$

Solution

We choose the significance level $\alpha = 0.05$ as it is customary at this point. An estimate of the mean will be obtained from:

$$\bar{X} = \frac{1}{n} (X_1 + \dots + X_n).$$

If the hypothesis is true, \bar{X} is normal with mean $\mu = 24$ and variance $\sigma^2/n = 0.9$. Therefore we may obtain the critical value c from X .

Right-Sided Test We determine c from

$$P(\bar{X} > c)_{\mu=24} = \alpha = 0.05$$

that is,

$$P(\bar{X} \leq c)_{\mu=24} = \Phi\left(\frac{c-24}{\sqrt{0.9}}\right) = 1 - \alpha = 0.95.$$

Reverse engineering Table ?? by looking for 0.95 percentile gives $(c - 24)/\sqrt{0.9} = 1.645$, and $c = 25.56$, which is greater than μ_0 . If $\bar{x} \leq 25.56$, the hypothesis is **accepted**. If $\bar{x} > 25.56$, it is rejected ■

Left-Sided Test The critical value c is obtained from the equation

$$P(\bar{X} \leq c)_{\mu=24} = \Phi\left(\frac{c-24}{\sqrt{0.9}}\right) = \alpha = 0.05.$$

Reverse engineering Table ?? by looking for 0.95 percentile gives $c = 24 - \sqrt{0.9} \times 1.645 = 22.44$. If $\bar{x} \geq 22.44$, we accept the hypothesis. If $\bar{x} < 22.44$, we reject it ■

Two-Sided Test As the normal distribution is **symmetric**, we choose c_1 and c_2 equidistant from $\mu = 24$, say, $c_1 = 24 - k$ and $c_2 = 24 + k$, and determine k from:

$$P(24 - k \leq \bar{X} \leq 24 + k)_{\mu=24} = \Phi\left(\frac{k}{\sqrt{0.9}}\right) - \Phi\left(\frac{-k}{\sqrt{0.9}}\right) = 1 - \alpha = 0.95.$$

Looking for 0.975 in Table ?? gives $k/\sqrt{0.9} = 1.960$, therefore $k = 1.86$. This gives the values $c_1 = 24 - 1.86 = 22.14$ and $c_2 = 24 + 1.86 = 25.86$. If \bar{x} is not smaller than c_1 and not greater than c_2 , we accept the hypothesis. Otherwise, we reject it ■

4.5 Goodness of Fit

To test for goodness of fit³⁷ means that we wish to test that a certain function $F(x)$ is the distribution function of a distribution from which we have a sample x_1, \dots, x_n . Then we test whether the **sample distribution function** $\tilde{F}(x)$ defined as:

$$\tilde{F}(x) = \text{Sum of the relative frequencies of all sample values } x_j \text{ not exceeding } x, \quad (4.25)$$

fits $\tilde{F}(x)$ **sufficiently well**. If this is so, we shall **accept** the hypothesis that $\tilde{F}(x)$ is the distribution function of the population; else, we shall **reject the hypothesis**.

This test is of considerable practical importance, and it differs in character from the tests for parameters (μ, σ^2 , etc.) considered thus far.

To test in that fashion, we have to know how much $\tilde{F}(x)$ can differ from $F(x)$ if the hypothesis is **true**. Hence we must first introduce a quantity which measures the deviation of $\tilde{F}(x)$ from $F(x)$, and we must know the probability distribution of this quantity under the assumption that the hypothesis is true.

Then we proceed as follows.

We determine a number, lets use c , such that, if the hypothesis is **true**, a deviation greater than c has a small preassigned probability. If, nevertheless, a deviation greater than c occurs, we have reason to doubt that the hypothesis is true and we reject it. On the other hand, if the deviation does **NOT** exceed c , so that $\tilde{F}(x)$ approximates $F(x)$ sufficiently well, we accept the hypothesis.

Of course, if we accept the hypothesis, this means that we have insufficient evidence to reject it, and this does not exclude the possibility that there are other functions that would not be rejected in the test.

In this respect the situation is quite similar to hypothesis testing we talked previously.

The following text-block shows a test of that type, which was introduced by *R. A. Fisher*. This test is justified by the fact that if the hypothesis is true, then χ_0^2 is an observed value of a random variable whose distribution function approaches that of the chi-square distribution with $K - 1$ degrees of freedom³⁸ as n approaches infinity. The requirement that at least five (5) sample values lie in each interval results from the fact that for finite n that random variable has only approximately a chi-square distribution.

If the sample is so small that the requirement cannot be satisfied, one may continue with the test, but then use the result **with caution**.

³⁷In literature, this method also means χ^2 -Test.

Historical Anecdote

During the 19th century, statistical analytical methods were mainly applied to biological data and it was customary for researchers to assume observations followed a **normal distribution**, such as Sir George Airy and Mansfield Merriman, whose works were criticized by Karl Pearson in his 1900 paper.

At the end of the 19th century, Pearson noticed the existence of significant skewness within some biological observations. To model the observations regardless of being normal or skewed, Pearson, in a series of articles published from 1893 to 1916, devised the Pearson distribution, a family of continuous probability distributions, which includes the normal distribution and many skewed distributions, and proposed a method of statistical analysis consisting of using the Pearson distribution to model the observation and performing a test of goodness of fit to determine how well the model really fits to the observations.

³⁸or $K - r - 1$ degrees of freedom if r parameters are estimated.

Theory 4.22: Chi-square Test for $F(x)$ being the Distribution Function of a Population

- Subdivide the x -axis into n intervals I_1, \dots, I_n such that each interval contains at least five (5) values of the given sample x_1, \dots, x_n .

Determine the number b_j of sample values in the interval I_j , where $j = 1, \dots, K$. If a sample value lies at a common boundary point of two (2) intervals, add 0.5 to each of the two (2) corresponding b_j .

- Using $F(x)$, calculate the probability p_j that the random variable X under consideration assumes any value in the interval I_j , where $j = 1, \dots, K$. Then, calculate

$$e_j = np_j.$$

This is the number of sample values **theoretically expected** in I_j if the hypothesis is true.

- Compute the deviation:

$$\chi^2_0 = \sum_{j=1}^K \frac{(b_j - e_j)^2}{e_j}.$$

- Choose a significance level such as 5%, 1%, or the like.

- Determine the solution c of the equation

$$P(\chi^2 \leq c) = 1 - \alpha.$$

from the table of the chi-square distribution with $K - 1$ degrees of freedom **Table ??** to **Table ??**.

If $\chi^2_0 \leq c$, accept the hypothesis. If $\chi^2_0 > c$, reject the hypothesis.

Exercise 4.9: Printed Circuit Boards

The number of defects in printed circuit board is hypothesized to follow a Poisson distribution. A random sample of $n = 60$ printed boards have been collected, and following number of defectswere observed

Number of Defects	Observed Frequency
0	32
1	15
2	9
3	4

Solution

The mean of the assumed Poisson distribution in this example is unknown and must be estimatedfrom the sample data. The estimate of the mean number of defects per board is the sample average,that is:

$$(32 \times 0 + 15 \times 1 + 9 \times 2 + 4 \times 3) / 60 = 0.75$$

From the Poisson distribution with parameter 0.75, we may compute p_i , the theoretical,hypothesized probability associated with the i^{th} class interval. Since each classinterval corresponds to a particular number of defects, we may find the p_i as follows:

$$p_1 = P(X = 0) = \frac{e^{-0.75}(0.75)^0}{0!} = 0.472$$

$$p_2 = P(X = 1) = \frac{e^{-0.75}(0.75)^1}{1!} = 0.354$$

$$p_3 = P(X = 2) = \frac{e^{-0.75}(0.75)^2}{2!} = 0.133$$

$$p_4 = P(X \geq 3) = 1 - (p_1 + p_2 + p_3) = 0.041$$

The expected frequencies are computed by multiplying the sample size $n = 60$ times the probabilities p_i . That is, $e_i = np_i$. The expected frequencies follow:

Number of Defects	Probability	Expected Frequency
0	0.472	28.32
1	0.354	21.24
2	0.133	7.98
3 (or more)	0.041	2.46

Since the expected frequency in the last cell is less than 3, we combine the last two cells:

NOTE: Categories with expected frequency is combined because the Chi-square test would not work if the frequency is less than 5. If the sample size is too small the chi-square value is over-estimated and if it is too large chi-square value is under-estimated. Hence why we combine with the category with the lowest frequency.

Since the expected frequency in the last cell is less than 3, we combine the last two cells:

Number of Defects	Probability	Expected Frequency
0	32	28.32
1	15	21.24
2 (or more)	13	10.44

Now, the chi-square test will have $k - p - 1 = 3 - 1 - 1 = 1$ degree of freedom, because the mean of the Poisson distribution was estimated from the data.

The hypothesis-testing procedure may now be applied using $\alpha = 0.05$,

1. The variable of interest is the form of the distribution of defects in printed circuitboards.
 2. H_0 The form of the distribution of defects is Poisson.
 3. H_1 The form of the distribution of defects is not Poisson.
 4. Test statistic is:
- $$\chi^2_0 = \sum_{j=1}^k \frac{(b_j - e_j)^2}{e_j}$$
5. Reject H_0 if $\chi^2_0 > \chi^2_{0.05,1} = 3.84$.
 6. Time to calculate χ^2_0 :

$$\chi^2_0 = \frac{(32 - 28.32)^2}{28.32} + \frac{(15 - 21.24)^2}{21.24} + \frac{(13 - 10.44)^2}{10.44} = 2.94$$

7. As $\chi^2_0 = 2.94 < \chi^2_{0.05,1} = 3.84$, we are unable to reject the null hypothesis that the distribution of defects in printed circuit boards is Poisson.

4.6 Regression and Correlation

Up to this points, we were only concerned with **random experiments** in which we observed a single quantity³⁹ and got samples whose values were single numbers. In this section we discuss experiments

³⁹In this case it is a random variable.
in which we observe or measure two (2) quantities simultaneously, so we get samples of **pairs** of values:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

Most applications involve one of two kinds of experiments, which are as follows:

Regression one (1) of the two (2) variables, call it x , can be regarded as an ordinary variable because we can measure it without substantial error or we can even give it values we want. x is called the **independent variable**, or sometimes the **controlled variable** as we can **control** it.⁴⁰ The other variable, Y , is a random variable, and we are interested in the **dependence** of Y on x .

⁴⁰set it at values we choose

Examples include the dependence of the blood pressure Y on the age x of a person or, as we shall now say, the regression of Y on x . The regression of the gain of weight Y of certain animals on the daily ratio of food x , the regression of the heat conductivity Y of work on the specific weight x of the rock, etc.

Correlation both quantities are random variables and we are interested in relations between them.

⁴¹we say correlation

Examples are the relation⁴¹ between user X and year Y of the front tires of cars, between grades X and Y of students in mathematics and in physics, respectively, between the hardness X of steel plates in the centre and the hardness Y near the edges of the plates, etc.

4.6.1 Regression Analysis

In regression analysis the dependence of Y on x is a dependence of the mean μ of Y on x , so that $\mu = \mu(x)$ is a function in the ordinary sense. The curve of $\mu(x)$ is called the **regression curve** of Y on x .

Let's look into the simplest case, namely, that of a **straight regression line**:

$$\mu(x) = \kappa_0 + \kappa_1 x. \quad (4.26)$$

Then we may want to graph the sample values as n points in the xY -plane, fit a straight line through them, and use it for estimating $\mu(x)$ at values of x that interest us, so we know what values of Y we can expect for those x .

Fitting line by eye would not be good because it would be **subjective** as people would come up with different estimations, particularly if the points are scattered. So we need a mathematical method

which gives a **unique result** depending **only** on the n points. A widely used procedure is the **method of least squares** by Gauss and Legendre.

For our task we may formulate it as follows.

Theory 4.23: Least Square Principle

The straight line should be fitted through the given points so that the sum of the squares of the distances of those points from the straight line is **minimum**, where the distance is measured in the vertical direction, which is the y -direction.

To get uniqueness of the straight line, we need some extra condition.

To see this, lets look at a small example and take the sample $(0, 1), (0, -1)$. Then all the lines $y = k_1x$ with any k_1 satisfy the principle.

The following assumption will imply uniqueness, as we shall find out.

Theory 4.24: Assumption A1

The x -values x_1, \dots, x_n in our sample $(x_1, y_1), \dots, (x_n, y_n)$ are not all equal.

From a given sample $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ we shall now determine a **straight line** by least squares. We write the line as:

$$y = k_0 + k_1x, \quad (4.27)$$

and call it the **sample regression line** as it will be the counterpart of the population regression line given in Eq. (4.26).

Now a sample point (x_j, y_j) has the vertical distance⁴² from Eq. (4.27) given by:

⁴²distance measured in the y -direction

$$\left| y_j - (k_0 + k_1 x_j) \right| \quad (4.28)$$

This gives the sum of the squares of these distances as:

$$q = \sum_{j=1}^n \left(y_j - (k_0 + k_1 x_j) \right)^2 \quad (4.29)$$

In the method of least squares we now have to determine k_0 and k_1 such that q is minimum. From calculus we know that a necessary condition for this is:

$$\frac{\partial q}{\partial k_0} = 0 \quad \text{and} \quad \frac{\partial q}{\partial k_1} = 0. \quad (4.30)$$

We shall see that from this condition we obtain for the sample regression line the formula

$$y - \bar{y} = k_1 (x - \bar{x}). \quad (4.31)$$

Here \bar{x} and \bar{y} are the means of the x - and the y -values in our sample, that is,

$$\bar{x} = \frac{1}{n} (x_1 + \cdots + x_n) \quad \text{and} \quad \bar{y} = \frac{1}{n} (y_1 + \cdots + y_n). \quad (4.32)$$

The slope k_1 in Eq. (4.31) is called the **regression coefficient** of the sample and is given by:

$$k_1 = \frac{s_{xy}}{s_x^2}. \quad (4.33)$$

Here the **sample covariance** s_{xy} is:

$$s_{xy} = \frac{1}{n-1} \sum_{j=1}^n (x_j - \bar{x})(y_j - \bar{y}) = \frac{1}{n-1} \left[\sum_{j=1}^n x_j y_j - \frac{1}{n} \left(\sum_{i=1}^n x_i \right) \left(\sum_{j=1}^n y_j \right) \right] \quad (4.34)$$

and s_x^2 is given by

$$s_x^2 = \frac{1}{n-1} \sum_{j=1}^n (x_j - \bar{x})^2 = \frac{1}{n-1} \left[\sum_{j=1}^n x_j^2 - \frac{1}{n} \left(\sum_{j=1}^n x_j \right)^2 \right]. \quad (4.35)$$

From Eq. (4.31) we see that the sample regression line passes through the point (\bar{x}, \bar{y}) , by which it is determined, together with the regression coefficient Eq. (4.33). We may call s_x^2 the *variance* of the x -values, but keep in mind that x is an ordinary variable, and **NOT** a random variable.

We shall soon also need:

$$s_y^2 = \frac{1}{n-1} \sum_{j=1}^n (y_j - \bar{y})^2 = \frac{1}{n-1} \left[\sum_{j=1}^n y_j^2 - \frac{1}{n} \left(\sum_{j=1}^n y_j \right)^2 \right]. \quad (4.36)$$

Now, let's try to derive Eq. (4.31) and Eq. (4.33). Differentiating Eq. (4.29) and using Eq. (4.30), we first obtain:

$$\begin{aligned} \frac{\partial q}{\partial k_0} &= -2 \sum (y_j - k_0 - k_1 x_j) = 0, \\ \frac{\partial q}{\partial k_1} &= -2 \sum x_j (y_j - k_0 - k_1 x_j) = 0. \end{aligned}$$

where we sum over j from 1 to n . We now divide by 2, write each of the two sums as three sums, and take the sums containing y_j and $x_j y_j$ over to the right.

Then we get the **normal equations**:

$$\begin{aligned} k_0 n + k_1 \sum x_j &= \sum y_j \\ k_0 \sum x_j + k_1 \sum x_j^2 &= \sum x_j y_j. \end{aligned} \quad (4.37)$$

This is a **linear system** of two (2) equations with two unknowns k_0 , k_1 , with coefficient determinant being:

$$\begin{vmatrix} n & \sum x_j \\ \sum x_j & \sum x_j^2 \end{vmatrix} = n \sum x_j^2 - (\sum x_j)^2 = n(n-1)s_x^2 = n \sum (x_j - \bar{x})^2.$$

and is **NOT** zero due to the first assumption (A1) we made prior. Hence the system has a **unique solution**. Dividing the first equation of Eq. (4.37) by n and using Eq. (4.32), we get $k_0 = \bar{y} - k_1 \bar{x}$.

Together with $y = k_0 + k_1 x$ in Eq. (4.27) this gives Eq. (4.31). To get Eq. (4.33), we solve the system Eq. (4.37) by Cramer's rule or elimination, finding

$$k_1 = \frac{n \sum x_j y_j - \sum x_i \sum y_j}{n(n-1) s_x^2}. \quad (4.38)$$

Which completes the derivation.

Exercise 4.10: Regression Line

The decrease of volume y (in %) of leather for certain fixed values of high pressure x (atmosphere) was measured. The results are shown in the first two columns of the table below.

Given Values		Auxiliary Values	
x_j	y_j	x_j^2	$x_j y_j$
4000	2.3	16,000,000	9200
6000	4.1	36,000,000	24,600
8000	5.7	64,000,000	45,600
10,000	6.9	100,000,000	69,000
28,000	19.0	216,000,000	148,400

Table 4.3: Dataset

Find the regression line of y on x .

Solution

We see that the sample count is $n = 4$ and obtain the values

$$\bar{x} = \frac{28000}{4} = 7000 \quad \text{and} \quad \bar{y} = \frac{19.0}{4} = 4.75,$$

and from Eq. (4.35), Eq. (4.36) and Eq. (4.34)

$$s_x^2 = \frac{1}{3} \left(216,000,000 - \frac{28,000^2}{4} \right) = \frac{20,000,000}{3}$$

$$s_{xy} = \frac{1}{3} \left(148,400 - \frac{28,000 \cdot 19}{4} \right) = \frac{15,400}{3}.$$

Hence $k_1 = 15,400/20,000,000 = 0.00077$ from Eq. (4.33), and the regression line is

$$y - 4.75 = 0.00077(x - 7000)$$

$$y = 0.00077x - 0.64.$$

With both options being valid. Note $y(0) = -10.65$, which is physically meaningless, but typically indicates that a linear relation is merely an approximation valid on some restricted interval ■

4.6.2 Confidence Intervals

If we want to get confidence intervals, we have to make assumptions about the distribution of Y .⁴³ We assume normality and independence in sampling:

⁴³which we have not made so far; least squares is a *geometric principle*, not involving probabilities.

Theory 4.25: Assumption A2

For each fixed x the random variable Y is normal with mean Eq. (4.26), that is,

$$\mu(x) = \kappa_0 + \kappa_1 x \quad (4.39)$$

and variance σ^2 **independent** of x .

Theory 4.26: Assumption A3

The n performances of the experiment by which we obtain a sample

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \quad (4.40)$$

are independent

κ_1 given in Eq. (4.39) is called the **regression coefficient** of the population because it can be shown that, under the assumptions given in A1 to A3, the maximum likelihood estimate of κ_1 is the sample regression coefficient k_1 given by Eq. (4.38).

Following with the assumptions from A1 to A3, we may now obtain a confidence interval for κ_1 , as shown below.

Information: Determination of Regression Coefficient under Assumptions A1 to A3

1. Choose a confidence level γ which can take values of 95%, 99%, or the like.
2. Determine the solution c of the equation,

$$F(c) = \frac{1}{2} (1 + \gamma) \quad (4.41)$$

from the table of the t -distribution with $n - 2$ degrees of freedom (Table A9 in App. 5; n = sample size)

3. Using a sample $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, calculate $(n - 1) s_x^2$ from Eq. (4.35), $(n - 1) s_{xy}$ from Eq. (4.34), k_1 from Eq. (4.33),

$$(n - 1) s_y^2 = \sum_{j=1}^n y_j^2 - \frac{1}{n} \left(\sum_{j=1}^n y_j \right)^2$$

which was described in Eq. (4.36), and

$$q_0 = (n - 1) (s_y^2 - k_1^2 s_x^2)$$

4. Calculate:

$$K = c \sqrt{\frac{q_0}{(n - 2)(n - 1)s_x^2}}.$$

5. The confidence interval is then defined to be:

$$\text{CONF}_\gamma \{ k_1 - K \leq \kappa_1 \leq k_1 + K \}. \quad (4.42)$$

Exercise 4.11: Confidence Interval for the Regression Coefficient

Using the sample in the given table, determine a confidence interval for κ_1 by the method described just previously.

Given Values		Auxiliary Values	
x_j	y_j	x_j^2	$x_j y_j$
4000	2.3	16,000,000	9200
6000	4.1	36,000,000	24,600
8000	5.7	64,000,000	45,600
10,000	6.9	100,000,000	69,000
28,000	19.0	216,000,000	148,400

Table 4.4: the given dataset of measurement.

Solution

1. We start by choosing the confidence level: $\gamma = 0.95$.

2. $F(c) = 1/2(1+\gamma)$ takes the form $F(c) = 0.975$, and Table ?? with $n - 2 = 2$ degrees of freedom gives $c = 4.30$.

3. By using the table we have $3s_x^2 = 20\,000\,000$ and $k_1 = 0.00077$. From Table 4.4 we compute:

$$3s_y^2 = 102.0 - \frac{19^2}{4} = 11.95. \\ q_0 = 11.95 - 20\,000\,000 \cdot 0.00077^2 = 0.092.$$

4. We therefore obtain:

$$K = 4.30 \sqrt{\frac{0.092}{(2 \cdot 20,000,000)}} \\ = 0.000206 \\ \text{CONF}_{0.95} \left\{ 0.00056 \leq \kappa_1 \leq 0.00098 \right\} \blacksquare$$

4.6.3 Correlation Analysis

Time to give an introduction to the basic facts in correlation analysis.

The topic of **correlation analysis** is concerned with the **relation** between X and Y in a two-dimensional random variable (X, Y) . A sample consists of n ordered pairs of values:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n),$$

as before. The interrelation between the x and y values in the sample is measured by the sample covariance s_{xy} in Eq. (4.34) or by the sample **correlation coefficient**:

$$r = \frac{s_{xy}}{s_x s_y} \quad (4.43)$$

with s_x and s_y given in Eq. (4.35) and Eq. (4.36). Here r has the advantage that it does not change under a multiplication of the x and y values by a factor.⁴⁴

⁴⁴Such as the unit changing from g to kg.

Theory 4.27: Sample Correlation Coefficient

The sample correlation coefficient r satisfies $-1 \leq r \leq 1$.

In particular, $r = \pm 1$ if and only if the sample values lie on a straight line which can be seen in Fig. 4.6.

The theoretical counterpart of r is the **correlation coefficient** ρ of X and Y ,

$$\rho = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} \quad (4.44)$$

where:

$$\mu_X = E(X) \quad \mu_Y = E(Y) \quad \sigma_X^2 = E([X - \mu_X]^2) \quad \sigma_Y^2 = E([Y - \mu_Y]^2)$$

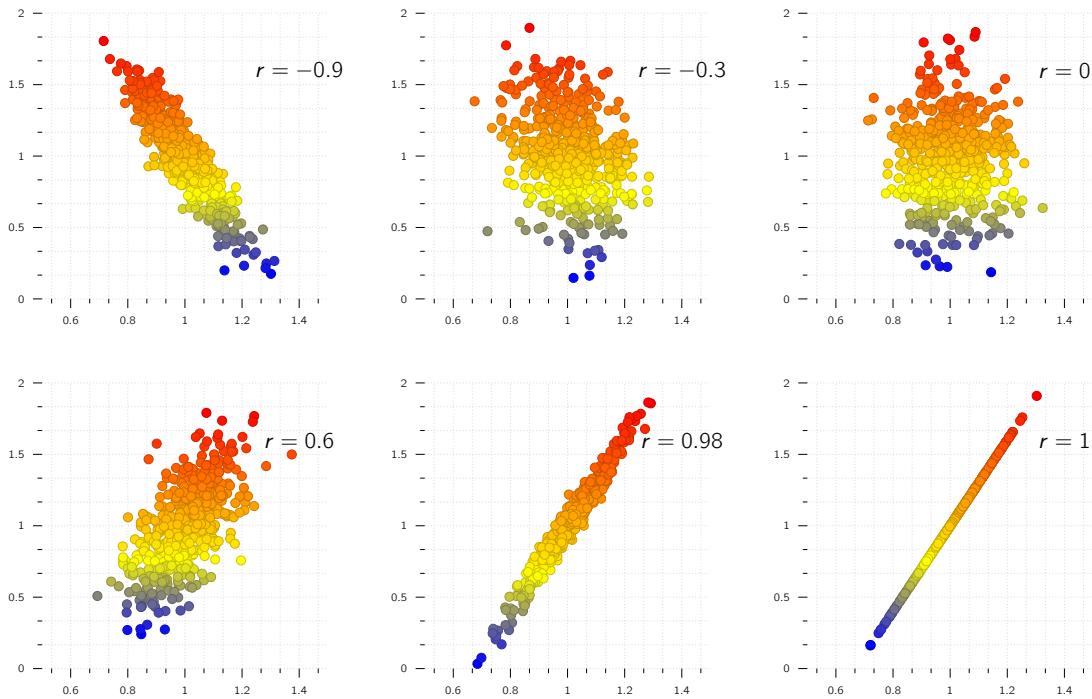


Figure 4.6: Samples with various values of the correlation coefficient r .

which are the means and variances of the marginal distributions of X and Y . The covariance (σ_{XY}), on the other hand, is defined as:

$$\sigma_{XY} = E \left([X - \mu_X] [Y - \mu_Y] \right) = E(XY) - E(X) E(Y) \quad (4.45)$$

Theory 4.28: Correlation Coefficient

The correlation coefficient ρ satisfies $-1 \leq \rho \leq 1$.

In particular, $\rho = \pm 1$ if and only if X and Y are **linearly related**, that is,

$$Y = \gamma X + \delta, X = \gamma_* Y + \delta_*$$

X and Y are **uncorrelated** if $\rho = 0$.

Theory 4.29: Independence and Relation to Normal Distribution

- a. If X and Y are independent, they are uncorrelated.
- b. If (X, Y) is normal, then uncorrelated X and Y are **independent**.

Here the two-dimensional normal distribution can be introduced by taking two (2) independent standardised normal random variables X^* , Y^* , whose joint distribution thus has the density:

$$f^* (x^*, y^*) = \frac{1}{2\pi} e^{-(x^{*2} + y^{*2})/2}$$

and setting

$$X = \mu_X + \sigma_X X^*$$

$$Y = \mu_Y + \rho \sigma_Y X^* + \sqrt{1 - \rho^2} \sigma_Y Y^*$$

This gives the general two-dimensional normal distribution with the density:

$$f(x, y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} e^{-h(x, y)/2} \quad (4.46)$$

where

$$h(x, y) = \frac{1}{1-\rho^2} \left[\left(\frac{x-\mu_X}{\sigma_X} \right)^2 - 2\rho \left(\frac{x-\mu_X}{\sigma_X} \right) \left(\frac{y-\mu_Y}{\sigma_Y} \right) + \left(\frac{y-\mu_Y}{\sigma_Y} \right)^2 \right] \quad (4.47)$$

In Theorem 3(b), normality is important, as we can see from the following example.

Exercise 4.12: Uncorrelated but Dependent Random Variables

If X assumes the value of $-1, 0, 1$ with probability $\frac{1}{3}$ and $Y = X^2$, then $EX = 0$ and in Eq. (4.29)

so that $\rho = 0$ and X and Y are uncorrelated. But they are certainly NOT independent since they are even functionally related.

$$\sigma_{XY} = E(XY) = E(X^3) = (-1)^3 \cdot \frac{1}{3} + 0^3 \cdot \frac{1}{3} + 1^2 \cdot \frac{1}{3} = 0,$$

4.6.4 Test for the Correlation Coefficient

The following text-block shows a test for ρ in the case of the two-dimensional normal distribution. t is an observed value of a random variable that has a t -distribution with $n - 2$ degrees of freedom.⁴⁵

⁴⁵This was shown by R. A. Fisher.

Information: Testing the Hypothesis against the Alternative in case of Two-Dimensional Normal Distribution

1. Choose a significance level α (5%, 1%, or the like).

2. Determine the solution c of the equation:

$$P(T \leq c) = 1 - \alpha,$$

from the t -distribution given in Table ?? to Table ?? with $n - 2$ degrees of freedom.

3. Calculate r from Eq. (4.43), using a sample $(x_1, y_1), \dots, (x_n, y_n)$.

4. Calculate

$$t = r \left(\sqrt{\frac{n-2}{1-r^2}} \right).$$

5. If $t \leq c$, accept the hypothesis. If $t > c$, reject the hypothesis.

Exercise 4.13: Test for the Correlation Coefficient

Test the hypothesis $\rho = 0$ (independence of X and Y , because of Theorem 3) against the alternative $\rho > 0$, using the data when $r = 0.6$ (normal soliding errors on 10 two-sided circuit boards done by 10 workers $x = \text{front}$, $y = \text{back of the bonds}$).

Solution

We choose $\alpha = 5\%$; which makes $1 - \alpha = 95\%$. Since

$n = 10$, and $n - 2 = 8$, the table gives $c = 1.86$. Also,

$$t = 0.6 \sqrt{\frac{8}{0.64}} = 2.12 > c.$$

We reject the hypothesis and search that there is a positive correlation. A worker making few errors on the front side also tends to make few errors on the reverse side of the bond ■

4.7 Bayesian Statistics

The classical methods of estimation that we have studied in this text are based solely on information provided by the random sample. These methods essentially interpret probabilities as relative frequencies.

For example, in arriving at a 95% confidence interval for μ , we interpret the statement:

$$P(-1.96 < Z < 1.96) = 0.95$$

to mean that 95% of the time in repeated experiments Z will fall between -1.96 and 1.96, as:

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$$

for a normal sample with known variance, the probability statement here means that 95% of the random intervals:

$$(\bar{X} - 1.96\sigma/\sqrt{n}, \bar{X} + 1.96\sigma/\sqrt{n})$$

contain the true mean μ . Another approach to statistical methods of estimation is called **Bayesian** methodology. The main idea of the method comes from Bayes' rule.

The key difference between the Bayesian approach and the classical or frequentist approach is that in Bayesian concepts, the parameters are viewed as random variables.

4.7.1 Subjective Probability

Subjective probability is the foundation of Bayesian concepts. In Chapter 2, we discussed two possible approaches to probability, namely the relative frequency and the indifference approaches. The first one determines a probability as a consequence of repeated experiments. For instance, to decide the free-throw percentage of a basketball player, we can record the number of shots made and the total number of attempts this player has made. The probability of hitting a free-throw for this player can be calculated as the ratio of these two numbers. On the other hand, if we have no knowledge of any bias in a die, the probability that a 3 will appear in the next throw will be 1/6. Such an approach to probability interpretation is based on the indifference rule.

However, in many situations, the preceding probability interpretations cannot be applied. For instance, consider the questions What is the probability that it will rain tomorrow? How likely is it that this stock will go up by the end of the month? and What is the likelihood that two companies will be merged together? They can hardly be interpreted by the aforementioned approaches, and the answers to these questions may be different for different people. Yet these questions are constantly asked in daily life, and the approach used to explain these probabilities is called subjective probability, which reflects ones subjective opinion.

Part III

Localisation and Mapping

Chapter 5

Mobile Robot Localisation

Table of Contents

5.1	Introduction	137
5.2	The problems of Noise and Aliasing	139
5.3	Localisation v. Hard-Coded Navigation	145
5.4	Representing Belief	148
5.5	Representing Maps	152
5.6	Probabilistic Map-Based Localisation	160

5.1 Introduction



Figure 5.1: Navigation is one if not the most demanding and complicated task in AMR. However a successful implementation will result in a versatile AMR which can find its way in unknown environments such as exploring other planets [59].

Navigation is one of, if not, the most challenging competencies required of an AMR. Success in navigation requires success at the four (4) building blocks of navigation which are as follows:

Perception the robot must interpret its sensors to extract meaningful data

Localisation the robot must determine its position in the environment

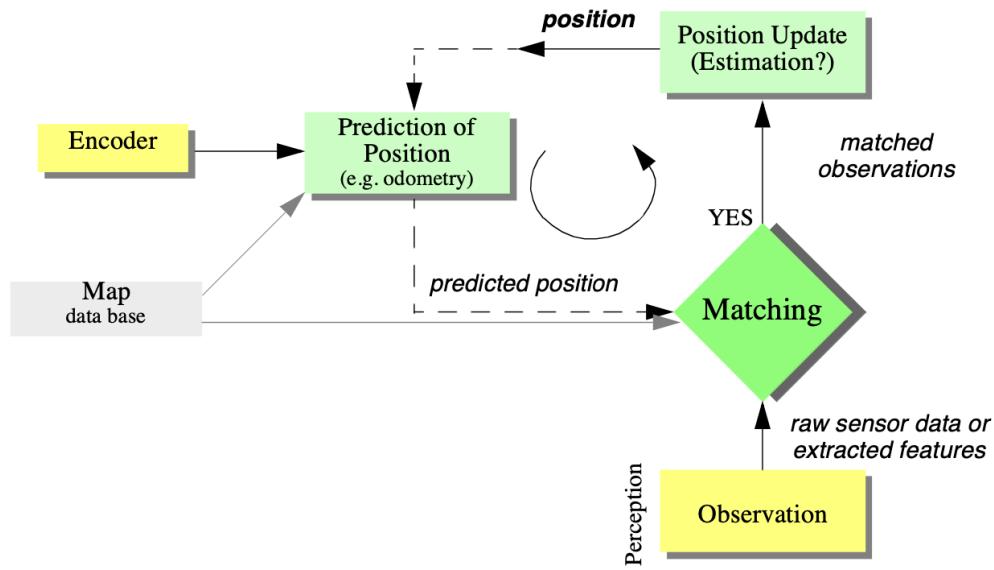


Figure 5.2: General schematic for mobile robot localisation.

Cognition the robot must decide how to act to achieve its goals

Motion control the robot must modulate its motor outputs to achieve the desired trajectory.

Of these four (4) aforementioned components, localisation has received the greatest research attention in the past and, as a result, significant advances have been made on this front [60, 61, 62]. In this chapter, we will explore the successful localisation methodologies and techniques used in academic research and industrial application [63].

The structure of the chapter is as follows:

- We will describe how sensor and effector uncertainty is responsible for the difficulties of localisation in Section 5.2,
- Then, in Section 5.3, we will have a look at the two (2) extreme approaches to dealing with the challenge of robot localisation [64]:
 - Avoiding localisation altogether,
 - Performing explicit map-based localisation
- The remainder of the chapter discusses the question of representation, which we will have a look at different case studies of successful localisation systems using a variety of representations and techniques to achieve AMR localisation.

5.2 The problems of Noise and Aliasing

If one could attach an accurate GPS sensor to an AMR, much of the localisation problem would be obviated. GPS would then inform the robot of its **exact** position and orientation, indoors and outdoors, so the answers to the questions,

Where am I?, Where am I going?, and, How should I get there? [65]

would **always** be immediately available.

Unfortunately, such a sensor is **NOT** currently practical!¹ The existing GPS network provides accuracy to within several cm, which is still not the optimal accuracy for localising human-scale mobile robots as well as miniature mobile robots such as desk robots and the body-navigating nano-robots of the future.

In addition, GPS cannot function indoors or in obstructed areas and are therefore limited in their workspace. But, looking beyond the limitations of GPS, localisation implies more than knowing one's absolute position in the Earth's reference frame.

Consider a robot which is interacting with humans. This robot may need to identify its absolute position, but its relative position with respect to target humans is also equally important. Its localisation task can include:

- identifying humans using its sensor array [66],
- then computing its relative position to the humans.

Furthermore, during operation a robot will select a strategy for achieving its goals. If it intends to reach a particular location, then localisation may not be enough. The robot may need to acquire or build an environmental model;² which aids it in planning a path to the goal. Once again, localisation means more than simply determining an absolute pose in space; it means building a map, then identifying the robot's position relative to that map.

¹Of course, this is a bit of a semi-false statement as we have technology which allows the shrinking of errors down to cm using Real-time kinematic positioning which is used to correct Global Navigation Satellite System (GNSS), which transmits the robot's location by longitude, latitude, altitude, and a timestamp [62].

²i.e., a map representing 2D space or a 3D space if it is navigating rough terrain.

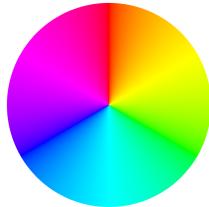
Clearly, the robot's sensors and effectors play an integral role in all the above forms of localisation. It is because of the inaccuracy and incompleteness of these sensors and effectors localisation poses difficult challenges.

5.2.1 Sensor Noise

Sensors are the fundamental robot input for the process of perception, and therefore the degree to which sensors can discriminate world state is critical. Sensor noise produces a limitation on the consistency of sensor readings in the same environmental state and, therefore, on the number of useful bits available from each sensor reading.

Often, the source of sensor noise problems is that some environmental features are not captured by the robot's representation and are thus overlooked.

³For example, this could be indoor office building, or a warehouse.



⁴One of the properties of a color, defined as the degree to which a stimulus can be described as similar to or different from stimuli that are described as red, orange, yellow, green, blue, violet within certain theories of colour vision.

For example, a vision system used for indoor navigation³ may use the colour values detected by its color CCD camera. When the sun is hidden by clouds, the illumination of the building's interior changes due to windows throughout the building. As a result, hue⁴ values are not constant. The color CCD appears noisy from the robot's perspective as if subject to **random error**, and the hue values obtained from the CCD camera will be unusable, unless the robot is able to note the position of the Sun and clouds in its representation.

Illumination dependency is only one example of the apparent noise in a vision-based sensor system [67]. Picture jitter, signal gain, blooming and blurring are all additional sources of noise, potentially reducing the useful content of a color video image.

Consider the noise level of ultrasonic range-measuring sensors, such as sonars, as we discussed previously. When a sonar transducer emits sound towards a relatively smooth and angled surface, much of the signal will coherently reflect away, failing to generate a return echo. Depending on the material characteristics, a small amount of energy may return nonetheless. When this level is close to the gain threshold of the sonar sensor, then the sonar will, at times, succeed and, at other times, fail to detect the object. From the robot's perspective, a virtually unchanged environmental state will result in two (2) different possible sonar readings:

one short, and one long which causes an nondeterministic behaviour.

⁵The propagation phenomenon resulting in signals reaching the receiver by two (2) or more paths. Causes of multipath can be atmospheric ducting, ionospheric reflection and refraction, and reflection from water bodies and terrestrial objects such as mountains and buildings.

⁶Sensor fusion is the process of using information from several different sensors to estimate the state of a dynamic system. The resulting estimate is, in some senses, better than it would be if the sensors were used individually [68].

The poor Signal-to-Noise Ratio (SNR) of a sonar sensor is further confounded by interference between multiple sonar emitters. Often, research robots have between 12 to 48 sonars on a single platform. In acoustically reflective environments, multipath interference⁵ is possible between the sonar emissions of one transducer and the echo detection circuitry of another transducer. The result can be dramatically large errors in ranging values due to a set of coincidental angles. Such errors occur rarely, less than 1% of the time, and are virtually random from the robot's perspective.

In conclusion, sensor noise reduces the useful information content of sensor readings. Clearly, the solution is to take multiple readings into account, employing temporal fusion or multi-sensor fusion⁶ to increase the overall information content of the robot's inputs.

5.2.2 Sensor Aliasing

Aliasing is the second major shortcoming of AMR sensors which cause them to give little information content, further amplifying the problem of **perception** and **localisation**.

Information: The Human Experience

The problem, known as sensor aliasing, is a phenomenon that humans seldom encounter. The human sensory system, particularly the visual system, tends to receive unique inputs in each unique local state within normal usage [69]. In other

words, every different place looks different. The power of this unique mapping is only apparent when one considers situations where this fails to hold.

Consider moving through an unfamiliar building that is completely dark. When the visual system sees only black, one's localisation system quickly degrades. Another useful example is that of a human-sized maze made from tall hedges. Such mazes have been created for centuries, and humans find them extremely difficult to solve without landmarks or clues because, without visual uniqueness, human localisation competence degrades rapidly.

In robots, the non-uniqueness of sensors readings, or sensor aliasing⁷, is the norm and not the exception. Consider a narrow-beam rangefinder such as ultrasonic or infrared rangefinders. This sensor provides range information in a single direction without any additional data regarding material composition such as color, texture and hardness. Even for a robot with several such sensors in an array, there are a variety of environmental states that would trigger the same sensor values across the array. Formally, there is a many-to-one mapping from environmental states to the robot's perceptual inputs. Therefore, the robot's sensors cannot distinguish from among these many states.



⁷Sensor aliasing in multiple types of sensors. One of the most apparent one is usually seen in digital images. For example, in the image above, due to low sampling, moiré pattern starts to be seen [70].

A classical problem with sonar-based robots involves distinguishing between humans and inanimate objects in an indoor setting [71, 72].

When facing an apparent obstacle in front of itself, should the robot say "Excuse me" because the obstacle may be a moving human, or should the robot plan a path around the object because it may be a cardboard box?

With sonar alone, these states are aliased and differentiation is impossible.

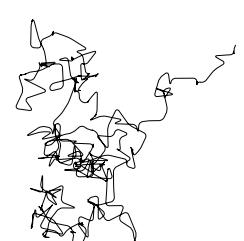
The navigation problem due to sensor aliasing is that, even with noise-free sensors, the amount of information is generally insufficient to identify the robot's accurate position from a single sensor's reading. Therefore techniques need to be employed by the robot programmer which base the robot's localisation on a series of readings and sufficient information to recover the robot's position over time.

5.2.3 Effector Noise

The challenges of localisation does NOT lie with sensor technologies alone. Just as robot sensors are noisy, limiting the information content of the signal, so do the robot effectors.

A single action taken by a AMR may have several different possible results, even though from the robot's point of view the initial state before the action was taken is well-known.

In short, AMR effectors introduce uncertainty about future state.⁸ The simple act of moving tends to increase the uncertainty of a mobile robot. There are, of course, exceptions. Using filters and predictive modelling, the motion can be carefully planned so as to minimise this effect, and indeed sometimes to actually result in more certainty. Furthermore, when the robot actions are taken in concert with careful interpretation of sensory feedback, it can compensate for the uncertainty introduced by noisy actions using the information provided by the sensors.



⁸An over-exaggerated example of effector noise where the motion is severely affected by the uncertainty caused by the deterministic error.

First, however, it is important to understand the precise nature of the effector noise that impacts AMR. It is important to note that, from the robot's point of view, this error in motion is viewed as **error in odometry**, or the robot's inability to estimate its own position over time using knowledge of its kinematics and dynamics. The true source of error generally lies in an **incomplete model** of the environment.⁹ All of these un-modeled sources of error result in:

- inaccuracy between the physical motion of the robot,
- the intended motion of the robot, and the
- proprioceptive sensor estimates of motion.

¹⁰The process of calculating the current position of a moving object by using a previously determined position, or fix, and incorporating estimates of speed, heading (or direction or course), and elapsed time.

In odometry and dead reckoning¹⁰ the position update is based on proprioceptive sensors. The movement of the robot, sensed with wheel encoders and /or heading sensors is integrated to compute position. Because the sensor measurement errors are integrated, the position error accumulates over time. Thus the position has to be updated from time to time by other localisation mechanisms. Otherwise the robot is not able to maintain a meaningful position estimate in long run.

In the following we will concentrate on odometry based on the wheel sensor readings of a differential drive robot only [73].¹¹

There are many sources of odometric error, from environmental factors to resolution:

- Limited resolution during integration¹²
- Misalignment of wheels causing **deterministic** error,
- Unequal wheel diameter, which again, causing **deterministic** error,
- Unequal floor contact, which can cause **slipping** during operation.

¹³To reiterate, deterministic errors are any errors which can be avoided and are generally caused by bad design or poorly calibrated sensors.

Some of the errors might be deterministic¹³ (systematic). However, there are still a number of non-deterministic (random) errors which remain, leading to uncertainties in position estimation over time. From a geometric point of view one can classify the errors into three (3) types:

Range error Integrated path length of the robot movement, as in the sum of wheel motion.

Turn error Similar to range error, but for turns which are difference of the wheel motions.

Drift error difference in the error of the wheels leads to an error in the robot's angular orientation.

Over long periods of time, turn and drift errors far outweigh range errors, as their contribute to the overall position error is non-linear. Consider a robot, whose position is initially perfectly well-known, moving forward in a straight line along the **x** axis. The error in the **y**-position introduced by a move of d meters will have a component of $d \sin \Delta\theta$, which can be quite large as the angular error $\Delta\theta$ grows. Over time, as an AMR moves about the environment, the rotational error between its internal reference frame and its original reference frame grows quickly. As the robot moves away

from the origin of these reference frames, the resulting linear error in position grows quite large. It is instructive to establish an error model for odometric accuracy and see how the errors propagate over time.

5.2.4 An Error Model for Odometric Position Estimation

Generally the pose (position) of a robot is represented by the vector

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

For a differential drive robot the position can be estimated starting from a known position by integrating the movement (summing the incremental travel distances). For a discrete system with a fixed sampling interval Δt the incremental travel distances ($\Delta x; \Delta y; \Delta \theta$) are:

$$\begin{aligned} \Delta x &= \Delta s \cos(\theta + \Delta\theta/2) & \Delta y &= \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta &= \frac{\Delta s_r - \Delta s_l}{b} & \text{and} & \Delta s = \frac{\Delta s_r + \Delta s_l}{2} \end{aligned}$$

As we discussed earlier, odometric position updates can give only a very rough estimate of the actual position. Due to integration errors of the uncertainties of p and the motion errors during the incremental motion (s ; s) the position error based on odometry integration r_l grows with time. In the next step we will establish an error model for the integrated position p' to obtain the of the odometric position estimate. To do so, we assume that at the covariance matrix starting point the initial covariance matrix is known. For the motion increment p (s ; s) we assume the following covariance matrix : r_l where s and s are the distances travelled by each wheel, and k_r, k_l are error constants r_l representing the non-deterministic parameters of the motor drive and the wheel-floor inter- action. As you can see in equation (5.8) we made the following assumption

- The two (2) errors of the individually driven wheels are independent.¹⁴
- The errors are proportional to the absolute value of the travelled distances ($\delta s_r; \delta s_l$).

These assumptions, while **NOT** perfect in the slightest, are nevertheless suitable and will therefore be used for the further development of the error model.

¹⁴If there is more knowledge regarding the actual robot kinematics, the correlation terms of the covariance matrix could also be used.

The **motion errors** are due to unprecise movement because of deformation of wheel, slippage, unequal floor, errors in encoders, ...

The values for the error constants k_r and k_l depend on the robot and the environment and should be experimentally determined by performing and analysing representative movements.

If we assume that p and $\Delta r_l = (\delta s_r; \delta s_l)$ are **uncorrelated** and the derivation of f (equ. (5.7)) can be reasonably approximated by the first-order Taylor expansion (linearization) we conclude, using

the error propagation law (see section 4.2.3):

$$\Sigma_{\mathbf{p}'} = \nabla_{\mathbf{p}} f \cdot \Sigma_{\mathbf{p}} \cdot \nabla_{\mathbf{p}} f^T + \nabla_{\delta_{\mathbf{r}}} f \cdot \Sigma_{\mathbf{f}_\text{ff}} \cdot \nabla_{\delta_{\mathbf{r}}} f^T$$

The covariance matrix is, of course, always given by the $\mathbf{p} \mathbf{p}'$

5.3 Localisation v. Hard-Coded Navigation

Fig. 5.3 depicts a standard indoor environment an AMR is set to navigate. Now, suppose an AMR in question must deliver messages between two (2) specific rooms in this environment:

These are rooms A and B.

In creating a navigation system for this task, it is clear the AMR will need sensors and a motion control system. Sensors are required to avoid hitting moving obstacles such as humans, and some motion control system is required so that the robot can actively move.

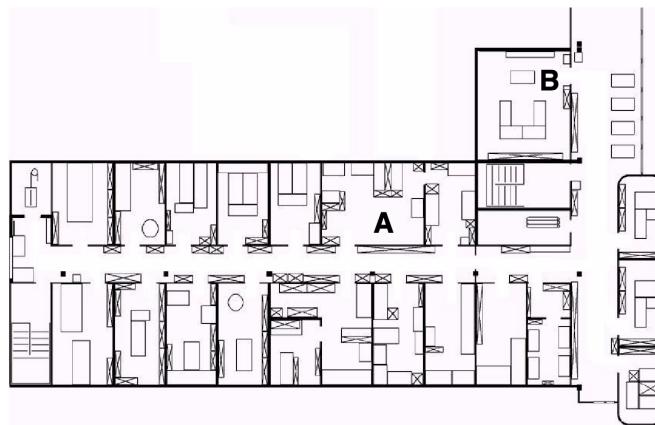


Figure 5.3: A sample environment.

It is less evident, however, whether or not this AMR will require a localisation system. Localisation may seem mandatory to successfully navigate between the two (2) rooms. It is through localising on a map, after all, which the robot can hope to recover its position and detect when it has arrived at the goal location. It is true that, at the least, the robot must have a way of detecting the goal location. However, explicit localisation with reference to a map is **NOT** the only strategy that qualifies as a goal detector.

An alternative, adopted by the behaviour-based community, suggests that, since sensors and effectors are noisy and information-limited, one should **avoid** creating a geometric map for localisation. Instead, they suggest designing sets of behaviours which together result in the **desired robot motion**.

In its essence, this approach avoids explicit reasoning about localisation and position, and therefore generally avoids explicit path planning as well.

This technique is based on a idea, that there exists a procedural solution to the particular navigation problem at hand. For example, in **Fig.** 5.3, the behavioralist approach to navigating from Room A to Room B might be to design a left-wall-following behavior and a detector for Room B that is triggered by some unique queue in Room B, such as the color of the carpet. Then, the robot can reach Room B by engaging the left wall follower with the Room B detector as the termination condition for the program.

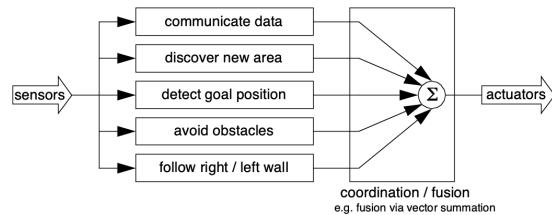


Figure 5.4: An Architecture for Behavior-based Navigation

The architecture of this solution to a specific navigation problem is shown in **Fig. 5.4**. The key advantage of this method is that, when possible, it may be implemented very quickly for a single environment with a small number of goal positions. It suffers from some disadvantages, however.

- The method does not directly scale to other environments or to larger environments. Often, the navigation code is location-specific, and the same degree of coding and debugging is required to move the robot to a new environment.
- The underlying procedures, such as left-wall-follow, must be carefully designed to produce the desired behaviour. This task may be time-consuming and is heavily dependent on the specific robot hardware and environmental characteristics.
- A behaviour-based system may have multiple active behaviors at any one time. Even when individual behaviours are tuned to optimise performance, this fusion and rapid switching between multiple behaviors can negate that fine-tuning. Often, the addition of each new incremental behavior forces the robot designer to re-tune all of the existing behaviors again to ensure that the new interactions with the freshly introduced behavior are all stable

In contrast to the behaviour-based approach, the map-based approach includes both localisation and cognition modules shown in **Fig. 5.5**.

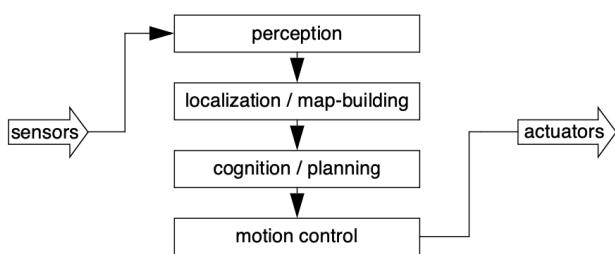


Figure 5.5: An Architecture for Map-based (or model-based) Navigation

In map-based navigation, the robot **explicitly** attempts to localise by collecting sensor data, then updating some belief about its position with respect to a map of the environment. The key advantages of the map-based approach for navigation are as follows:

- The explicit, map-based concept of position makes the system's belief about position transparent.

ently available to the human operators.

- The existence of the map itself represents a medium for communication between human and robot as the human can simply give the robot a new map if the robot goes to a new environment.
- The map, if created by the robot, can be used by humans as well, achieving two uses.

The map-based approach will require more up-front development effort to create a navigating AMR. The hope is that the development effort results in an architecture which can successfully map and navigate a variety of environments, thereby compensating for the up-front design cost over time.

Of course the primary risk of the map-based approach is that an internal representation, rather than the real world itself, is being constructed and trusted by the robot. If that model diverges from reality¹⁵ then the robot's behaviour may be undesirable at best or wrong at worst, even if the raw sensor values of the robot are only transiently incorrect.

In the remainder of this chapter, we focus on a discussion of map-based approaches and, specifically, the localisation component of these techniques. These approaches are particularly appropriate for study given their significant recent successes in enabling AMR to navigate a variety of environments, from academic research buildings to factory floors and museums around the world.

¹⁵As in if the robot gets the wrong idea about its environment and draws the wrong map.

5.4 Representing Belief

The fundamental issue which differentiates different types of map-based localisation systems is the issue of **representation**. There are two (2) specific concepts which the robot must represent, and each has its own unique possible solutions.

- Representation of the environment,
- The map.

What aspects of the environment are contained in this map? At what level of fidelity does the map represent the environment? These are the design questions for map representation.

The robot must also have a representation of its **belief** regarding its position on the map.

Does the robot identify a single unique position as its current position, or does it describe its position in terms of a set of possible positions? If multiple possible positions are expressed in a single belief, how are those multiple positions ranked, if at all?

These are the design questions for belief representation. Decisions along these two (2) design axes can result in varying levels of architectural complexity, computational complexity and overall localisation accuracy.

We will start by discussing belief representation. The first major branch in a taxonomy of belief representation systems differentiates between single hypothesis and multiple hypothesis belief systems.

- The former covers solutions in which the robot postulates its unique position,
- The latter enables a AMR to describe the degree to which it is uncertain about its position.

A sampling of different belief and map representations is shown in figure 5.9.

5.4.1 Single Hypothesis Belief

The single hypothesis belief representation is the most direct possible postulation of an AMR's position [74].

Given some environmental map, the robot's belief about position is expressed as a single unique point on the map.

In **Fig.** 5.6, three (3) examples of a single hypothesis belief are shown using three different map representations of the same actual environment shown in **Fig.** 5.6a. In 5.10b, a single point is geometrically annotated as the robot's position in a continuous two-dimensional geometric map. In 5.10c, the map is a discrete, tessellated map, and the position is noted at the same level of fidelity

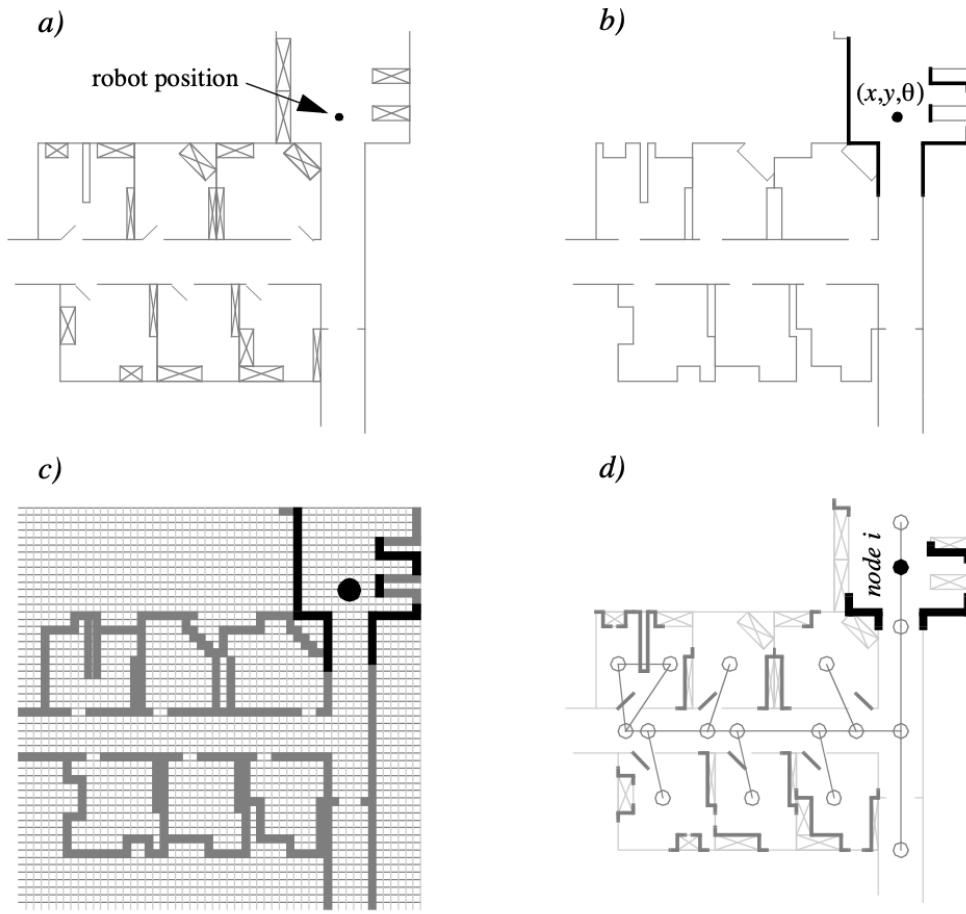


Figure 5.6: The three (3) examples of single hypotheses of position using different map representation. a) real map with walls, doors and furniture b) line-based map -> around 100 lines with two parameters c) occupancy grid based map -> around 3000 grid cells sizing 50x50 cm d) topological map using line features (Z/S-lines) and doors -> around 50 features and 18 nodes

as the map cell size. In 5.10d, the map is not geometrical at all but abstract and topological. In this case, the single hypothesis of position involves identifying a single node i in the topological graph as the robot's position.

The principal advantage of the single hypothesis representation of position stems from the fact that, given a unique belief, there is no position ambiguity. The unambiguous nature of this representation facilitates decision-making at the robot's cognitive level (e.g. path planning). The robot can simply assume that its belief is correct, and can then select its future actions based on its unique position.

Just as decision-making is facilitated by a single-position hypothesis, so updating the robot's belief regarding position is also facilitated, since the single position must be updated by definition to a new, single position. The challenge with this position update approach, which ultimately is the principal disadvantage of single-hypothesis representation, is that robot motion often induces uncertainty due to effector and sensory noise.

Forcing the position update process to always generate a single hypothesis of position is challenging and, often, impossible.

5.4.2 Multiple Hypothesis Belief

In the case of multiple hypothesis beliefs regarding position, the robot tracks **NOT** just a single possible position but a possibly **infinite set of positions**. In one simple example originating in the work of Jean-Claude Latombe [5, 89], the robot's position is described in terms of a convex polygon positioned in a two-dimensional map of the environment.

This multiple hypothesis representation communicates the set of possible robot positions geometrically, with no preference ordering over the positions. Each point in the map is simply either contained by the polygon and, therefore, in the robot's belief set, or outside the polygon and thereby excluded. Mathematically, the position polygon serves to partition the space of possible robot positions. Such a polygonal representation of the multiple hypothesis belief can apply to a continuous, geometric map of the environment or, alternatively, to a tessellated, discrete approximation to the continuous environment.

It may be useful, however, to incorporate some ordering on the possible robot positions, capturing the fact that some robot positions are likelier than others. A strategy for representing a continuous multiple hypothesis belief state along with a preference ordering over possible positions is to model the belief as a mathematical distribution. For example, [42, 47] notate the robot's position belief using an X,Y point in the two-dimensional environment as the mean plus a standard deviation parameter , thereby defining a Gaussian distribution. The intended interpretation is that the distribution at each position represents the probability assigned to the robot being at that location. This representation is particularly amenable to mathematically defined tracking functions, such as the Kalman Filter, that are designed to operate efficiently on Gaussian distributions.

An alternative is to represent the set of possible robot positions, not using a single Gaussian probability density function, but using discrete markers for each possible position. In this case, each possible robot position is individually noted along with a confidence or probability parameter (See Fig. (5.11)). In the case of a highly tessellated map this can result in thousands or even tens of thousands of possible robot positions in a single belief state.

The key advantage of the multiple hypothesis representation is that the robot can explicitly maintain uncertainty regarding its position. If the robot only acquires partial information regarding position from its sensors and effectors, that information can conceptually be incorporated in an updated belief.

A more subtle advantage of this approach revolves around the robot's ability to explicitly measure its own degree of uncertainty regarding position. This advantage is the key to a class of localisation and navigation solutions in which the robot not only reasons about reaching a particular goal, but reasons about the future trajectory of its own belief state. For instance, a robot may choose paths

that minimise its future position uncertainty. An example of this approach is [90], in which the robot plans a path from point A to B that takes it near a series of landmarks in order to mitigate localisation difficulties. This type of explicit reasoning about the effect that trajectories will have on the quality of localisation requires a multiple hypothesis representation.

One of the fundamental disadvantages of the multiple hypothesis approaches involves decision-making. If the robot represents its position as a region or set of possible positions, then how shall it decide what to do next? Figure 5.11 provides an example. At position 3, the robot's belief state is distributed among 5 hallways separately. If the goal of the robot is to travel down one particular hallway, then given this belief state what action should the robot choose?

The challenge occurs because some of the robot's possible positions imply a motion trajectory that is inconsistent with some of its other possible positions. One approach that we will see in the case studies below is to assume, for decision-making purposes, that the robot is physically at the most probable location in its belief state, then to choose a path based on that current position. But this approach demands that each possible position have an associated probability.

In general, the right approach to such a decision-making problems would be to decide on trajectories that eliminate the ambiguity explicitly. But this leads us to the second major disadvantage of the multiple hypothesis approaches. In the most general case, they can be computationally very expensive. When one reasons in a three dimensional space of discrete possible positions, the number of possible belief states in the single hypothesis case is limited to the number of possible positions in the 3D world. Consider this number to be N . When one moves to an arbitrary multiple hypothesis representation, then the number of possible belief states is the power set of N , which is far larger: 2^N . Thus explicit reasoning about the possible trajectory of the belief state over time quickly becomes computationally untenable as the size of the environment grows. There are, however, specific forms of multiple hypothesis representations that are somewhat more constrained, thereby avoiding the computational explosion while allowing a limited type of multiple hypothesis belief. For example, if one assumes a Gaussian distribution of probability centered at a single position, then the problem of representation and tracking of belief becomes equivalent to Kalman Filtering, a straightforward mathematical process described below. Alternatively, a highly tessellated map representation combined with a limit of 10 possible positions in the belief state, results in a discrete update cycle that is, at worst, only 10x more computationally expensive than single hypothesis belief update.

In conclusion, the most critical benefit of the multiple hypothesis belief state is the ability to maintain a sense of position while explicitly annotating the robot's uncertainty about its own position. This powerful representation has enabled robots with limited sensory information to navigate robustly in an array of environments, as we shall see in the case studies below.

5.5 Representing Maps

The problem of representing the environment in which an AMR moves is a dual of the problem of representing the robot's possible position or positions. Decisions made regarding the environmental representation can have impact on the choices available for robot position representation.

Often the fidelity of the position representation is bounded by the fidelity of the map.

There are three (3) fundamental relationships which must be understood when choosing a particular map representation:

1. The precision of the map must appropriately match the precision with which the robot needs to achieve its goals.
2. The precision of the map and the type of features represented must match the precision and data types returned by the robot's sensors.
3. The complexity of the map representation has direct impact on the computational complexity of reasoning about mapping, localisation and navigation.

Using the aforementioned criteria, we identify and discuss critical design choices in creating a map representation. Each such choice has great impact on the relationships, and on the resulting robot localisation architecture. As we will see, the choice of possible map representations is broad, if not expansive. Selecting an appropriate representation requires understanding all of the trade-offs inherent in that choice as well as understanding the specific context in which a particular AMR implementation must perform localisation.

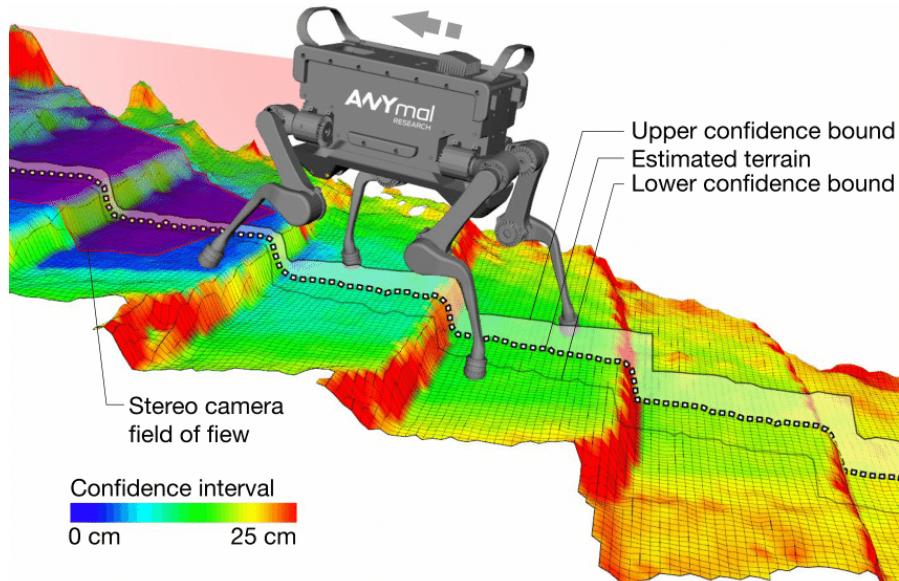


Figure 5.7: The presented robot-centric mapping framework enables mobile robots to create consistent elevation maps of the terrain. Mapping does not necessarily need to be done only in 2D as robots which will be used in outdoor environment would need the height of the map as well [75].

5.5.1 Continuous Representation

A continuous-valued map is one method for [exact](#) decomposition of the environment. The position of environmental features can be mapped precisely in continuous space.

AMR implementations to date use continuous maps only in two (2) dimensional representations, as increasing the number of dimensions can result in high computational load on the AMR navigation computer.

A common approach is to combine the exactness of a continuous representation with the compactness of the closed world assumption. This means that one assumes the representation will specify all environmental objects in the map, and that any area in the map which is devoid of objects has no objects in the corresponding portion of the environment. Therefore, the total storage needed in the map is proportional to the density of objects in the environment, and a sparse environment can be represented by a low-memory map.

One example of such a representation, shown in [Fig. 5.8](#), is a 2D representation in which polygons represent all obstacles in a continuous-valued coordinate space. This is similar to the method used by Latombe [5, 113] and others to represent environments for mobile robot path planning techniques. In the case of [5, 113], most of the experiments are in fact simulations run exclusively within the computer's memory. Therefore, no real effort would have been expended to attempt to use sets of polygons to describe a real-world environment, such as a park or office building.

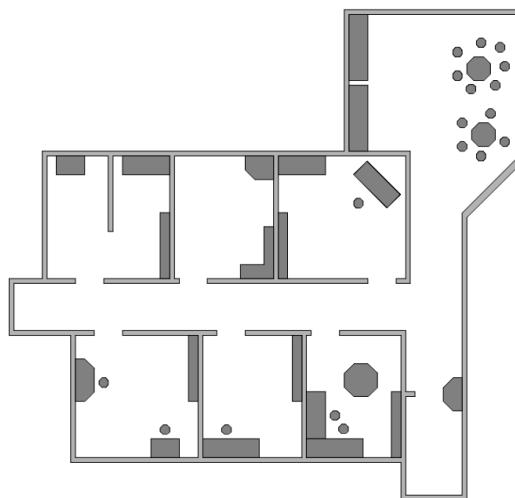


Figure 5.8: A continuous representation using polygons as environmental obstacles.

In other work in which real environments must be captured by the maps, there seems to be a trend towards [selectivity](#) and [abstraction](#). The human map-maker tends to capture on the map, for localisation purposes, only objects that can be detected by the robot's sensors and, furthermore, only a subset of the features of real-world objects.

It should be immediately apparent that geometric maps can capably represent the physical locations of objects without referring to their texture, colour, elasticity, or any other such secondary features that do not relate directly to position and space.

In addition to this level of abstraction, an AMR map can further reduce memory usage by capturing only **aspects of object geometry** which are **immediately relevant** to localisation. For example all objects may be approximated using very simple convex polygons,¹⁶ sacrificing map felicity for the sake of computational speed.

¹⁶A convex polygon is any shape that has all interior angles that measure less than 180 degrees

One excellent example involves **line extraction**. Many indoor AMR rely upon laser range-finding devices to recover distance readings to nearby objects. Such robots can automatically extract best-fit lines from the dense range data provided by thousands of points of laser strikes. Given such a line extraction sensor, an appropriate continuous mapping approach is to populate the map with a set of infinite lines. The continuous nature of the map guarantees that lines can be positioned at arbitrary positions in the plane and at arbitrary angles. The abstraction of real environmental objects such as walls and intersections captures only the information in the map representation that matches the type of information recovered by the AMR's rangefinding sensor.

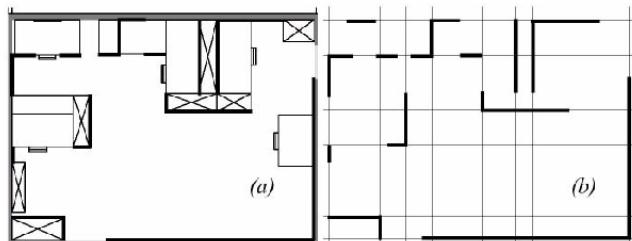


Figure 5.9: Example of a continuous-valued line representation of EPFL. left: real map right: representation with a set of infinite lines.

Fig. 5.9 shows a map of an indoor environment at EPFL using a continuous line representation. Note that the only environmental features captured by the map are straight lines, such as those found at corners and along walls. This represents not only a sampling of the real world of richer features, but also a simplification, for an actual wall may have texture and relief that is not captured by the mapped line. The impact of continuous map representations on position representation is primarily positive. In the case of single hypothesis position representation, that position may be specified as any continuous-valued point in the coordinate space, and therefore extremely high accuracy is possible. In the case of multiple hypothesis position representation, the continuous map enables two types of multiple position representation. In one case, the possible robot position may be depicted as a geometric shape in the hyperplane, such that the robot is known to be within the bounds of that shape. This is shown in Figure 5.30, in which the position of the robot is depicted by an oval bounding area. Yet, the continuous representation does not disallow representation of position in the form of a discrete set of possible positions. For instance, in [111] the robot position belief state is captured by sampling nine continuous-valued positions from within a region near the robot's best known position. This algorithm captures, within a continuous space, a discrete sampling of possible robot positions. In summary, the key advantage of a continuous map representation is the potential for high accuracy and expressiveness with respect to the environmental configuration as well as

the robot position within that environment. The danger of a continuous representation is that the map may be computationally costly. But this danger can be tempered by employing abstraction and capturing only the most relevant environmental features. Together with the use of the closed world assumption, these techniques can enable a continuous-valued map to be no more costly, and sometimes even less costly, than a standard discrete representation.

5.5.2 Decomposition Methods

In previous section, we discussed one method of simplification, in which the continuous map representation contains a set of infinite lines which approximate real-world environmental lines based on a two-dimensional slice of the world.

Basically this transformation from the real world to the map representation is a filter that removes all non-straight data and furthermore extends line segment data into infinite lines that require fewer parameters.

A more dramatic form of simplification is abstraction:

a general decomposition and selection of environmental features.

In this section, we explore decomposition as applied in its more extreme forms to the question of map representation. Why would one radically decompose the real environment during the design of a map representation? The immediate disadvantage of decomposition and abstraction is the loss of fidelity between the map and the real world. Both qualitatively, in terms of overall structure, and quantitatively, in terms of geometric precision, a highly abstract map does not compare favourably to a high-fidelity map.

Despite this disadvantage, decomposition and abstraction may be useful if the abstraction can be planned carefully so as to capture the relevant, useful features of the world while discarding all other features. The advantage of this approach is that the map representation can potentially be minimised. Furthermore, if the decomposition is hierarchical, such as in a pyramid of recursive abstraction, then reasoning and planning with respect to the map representation may be computationally far superior to planning in a fully detailed world model.

A standard, lossless form of opportunistic decomposition is termed exact cell decomposition. This method, introduced by [5], achieves decomposition by selecting boundaries between discrete cells based on geometric criticality.

Figure 5.14 depicts an exact decomposition of a planar workspace populated by polygonal obstacles. The map representation tessellates the space into areas of free space. The representation can be extremely compact because each such area is actually stored as a single node, shown in the graph at the bottom of Figure 5.14.

The underlying assumption behind this decomposition is that the particular position of a robot within

each area of free space does not matter. What matters is the robot's ability to traverse from each area of free space to the adjacent areas. Therefore, as with other representations we will see, the resulting graph captures the adjacency of map locales. If indeed the assumptions are valid and the robot does not care about its precise position within a single area, then this can be an effective representation that nonetheless captures the connectivity of the environment.

Such an exact decomposition is not always appropriate. Exact decomposition is a function of the particular environment obstacles and free space. If this information is expensive to collect or even unknown, then such an approach is not feasible.

An alternative is fixed decomposition, in which the world is tessellated, transforming the continuous real environment into a discrete approximation for the map. Such a transformation is demonstrated in Figure 5.15, which depicts what happens to obstacle-filled and free areas during this transformation. The key disadvantage of this approach stems from its inexact nature. It is possible for narrow passageways to be lost during such a transformation, as shown in Figure 5.15. Formally this means that fixed decomposition is sound but not complete. Yet another approach is adaptive cell decomposition as presented in Figure 5.16.

The concept of fixed decomposition is extremely popular in AMRics; it is perhaps the single most common map representation technique currently utilised. One very popular

version of fixed decomposition is known as the occupancy grid representation [91]. In an occupancy grid, the environment is represented by a discrete grid, where each cell is either filled (part of an obstacle) or empty (part of free space). This method is of particular value when a robot is equipped with range-based sensors because the range values of each sensor, combined with the absolute position of the robot, can be used directly to update the filled/empty value of each cell. In the occupancy grid, each cell may have a counter, whereby the value 0 indicates that the cell has not been "hit" by any ranging measurements and, therefore, it is likely free space. As the number of ranging strikes increases, the cell's value is incremented and, above a certain threshold, the cell is deemed to be an obstacle. By discounting the values of cells over time, both hysteresis and the possibility of transient obstacles can be represented using this occupancy grid approach. Figure 5.17 depicts an occupancy grid representation in which the darkness of each cell is proportional to the value of its counter. One commercial robot that uses a standard occupancy grid for mapping and navigation is the Cye robot [112].

There remain two main disadvantages of the occupancy grid approach. First, the size of the map in robot memory grows with the size of the environment and, if a small cell size is used, this size can quickly become untenable. This occupancy grid approach is not compatible with the closed world assumption, which enabled continuous representations to have potentially very small memory requirements in large, sparse environments. In contrast, the occupancy grid must have memory set aside for every cell in the matrix. Furthermore, any fixed decomposition method such as this imposes a geometric grid on the world *a priori*, regardless of the environmental details. This can be inappropriate in cases where geometry is not the most salient feature of the environment. For these reasons, an alternative, called topological decomposition, has been the subject of some exploration

in AMRics. Topological approaches avoid direct measurement of geometric environmental qualities, instead concentrating on characteristics of the environment that are most relevant to the robot for localisation. Formally, a topological representation is a graph that specifies two things: nodes and the connectivity between those nodes. Insofar as a topological representation is intended for the use of a AMR, nodes are used to denote areas in the world and arcs are used to denote adjacency of pairs of nodes. When an arc connects two nodes, then the robot can traverse from one node to the other without requiring traversal of any other intermediary node. Adjacency is clearly at the heart of the topological approach, just as adjacency in a cell decomposition representation maps to geometric adjacency in the real world. However, the topological approach diverges in that the nodes are not of fixed size nor even specifications of free space. Instead, nodes document an area based on any sensor discriminant such that the robot can recognise entry and exit of the node. Figure 5.18 depicts a topological representation of a set of hallways and offices in an indoor

environment. In this case, the robot is assumed to have an intersection detector, perhaps using sonar and vision to find intersections between halls and between halls and rooms. Note that nodes capture geometric space and arcs in this representation simply represent connectivity. Another example of topological representation is the work of Dudek [49], in which the goal is to create a AMR that can capture the most interesting aspects of an area for human consumption. The nodes in Dudek's representation are visually striking locales rather than route intersections. In order to navigate using a topological map robustly, a robot must satisfy two constraints. First, it must have a means for detecting its current position in terms of the nodes of the topological graph. Second, it must have a means for traveling between nodes using robot motion. The node sizes and particular dimensions must be optimised to match the sensory discrimination of the AMR hardware. This ability to "tune" the representation to the robot's particular sensors can be an important advantage of the topological approach. However, as the map representation drifts further away from true geometry, the expressiveness of the representation for accurately and precisely describing a robot position is lost. Therein lies the compromise between the discrete cell-based map representations and the topological representations. Interestingly, the continuous map representation has the potential to be both compact like a topological representation and precise as with all direct geometric representations. Yet, a chief motivation of the topological approach is that the environment may contain important non-geometric features - features that have no ranging relevance but are useful for localisation. In Chapter 4 we described such whole-image vision-based features. In contrast to these whole-image feature extractors, often spatially localised landmarks are artificially placed in an environment to impose a particular visual-topological connectivity upon the environment. In effect, the artificial landmark can impose artificial structure. Examples of working systems operating with this landmark-based strategy have also demonstrated success. Latombe's landmark-based navigation research [89] has been implemented on real-world indoor AMRs that employ paper landmarks attached to the ceiling as the locally observable features. Chips the museum robot is another robot that uses man-made landmarks to obviate the localisation problem. In this case, a bright pink square serves as a landmark with dimensions and color signature that would be hard to accidentally reproduce in a museum environment [88]. One such museum landmark is shown in Figure (5.19). In summary, range is clearly not the only measurable and useful environmental value for a AMR. This is particularly true due to the advent of color vision as well as

laser rangefinding, which provides reflectance information in addition to range information. Choosing a map representation for a particular AMR requires first understanding the sensors available on the AMR and second understanding the AMR's functional requirements (e.g. required goal precision and accuracy).

5.5.3 Current Challenges

The sections above describe major design decisions in regards to map representation choices. There are, however, fundamental real-world features that AMR map representations do not yet represent well. These continue to be the subject of open research, and several such challenges are described below. The real world is dynamic. As AMRs come to inhabit the same spaces as humans, they will encounter moving people, cars, strollers and the transient obstacles placed and moved by humans as they go about their activities. This is particularly true when one considers the home environment with which domestic robots will someday need to contend. The map representations described above do not, in general, have explicit facilities for identifying and distinguishing between permanent obstacles (e.g. walls, doorways, etc.) and transient obstacles (e.g. humans, shipping packages, etc.). The current state of the art in terms of AMR sensors is partly to blame for this shortcoming. Although vision research is rapidly advancing, robust sensors that discriminate between moving animals and static structures from a moving reference frame are not yet available. Furthermore, estimating the motion vector of transient objects remains a research problem. Usually, the assumption behind the above map representations is that all objects on the map are effectively static. Partial success can be achieved by discounting mapped objects over time. For example, occupancy grid techniques can be more robust to dynamic settings by introducing temporal discounting, effectively treating transient obstacles as noise. The more challenging process of map creation is particularly fragile to environment dynamics; most mapping techniques generally require that the environment be free of moving objects during the mapping process. One exception to this limitation involves topological representations. Because precise geometry is not important, transient objects have little effect on the mapping or localisation process, subject to the critical constraint that the transient objects must not change the topological connectivity of the environment. Still, neither the occupancy grid representation nor a topological approach is actively recognizing and representing transient objects as distinct from both sensor error and permanent map features. As vision sensing provides more robust and more informative content regarding the transience and motion details of objects in the world, AMRicists will in time propose representations that make use of that information. A classic example involves occlusion by human crowds. Museum tour guide robots generally suffer from an extreme amount of occlusion. If the robot's sensing suite is located along the robot's body, then the robot is effectively blind when a group of human visitors completely surrounds the robot. This is because its map contains only environment features that are, at that point, fully hidden from the robot's sensors by the wall of people. In the best case, the robot should recognise its occlusion and make no effort to localise using these invalid sensor readings. In the worst case, the robot will localise with the fully occluded data, and will update its location incorrectly. A vision sensor that can discriminate the local conditions of the robot (e.g. we are surrounded by people) can help eliminate this error mode. A second open challenge in AMR localisation involves the

traversal of open spaces. Existing localisation techniques generally depend on local measures such as range, thereby demanding environments that are somewhat densely filled with objects that the sensors can detect and measure. Wide open spaces such as parking lots, fields of grass and indoor atriums such as those found in convention centers pose a difficulty for such systems due to their relative sparseness. Indeed, when populated with humans, the challenge is exacerbated because any mapped objects are almost certain to be occluded from view by the people. Once again, more recent technologies provide some hope for overcoming these limitations. Both vision and state-of-the-art laser rangefinding devices offer outdoor performance with ranges of up to a hundred meters and more. Of course, GPS performs even better. Such long-range sensing may be required for robots to localise using distant features. This trend teases out a hidden assumption underlying most topological map representations. Usually, topological representations make assumptions regarding spatial locality: a node contains objects and features that are themselves within that node. The process of map creation thus involves making nodes that are, in their own self-contained way, recognizable by virtue of the objects contained within the node. Therefore, in an indoor environment, each room can be a separate node, and this is reasonable because each room will have a layout and a set of belongings that are unique to that room. However, consider the outdoor world of a wide-open park. Where should a single node end and the next node begin? The answer is unclear because objects that are far away from the current node, or position, can yield information for the localisation process. For example, the hump of a hill at the horizon, the position of a river in the valley and the trajectory of the sun all are non-local features that have great bearing on one's ability to infer current position. The spatial locality assumption is violated and, instead, replaced by a visibility criterion: the node or cell may need a mechanism for representing objects that are measurable and visible from that cell. Once again, as sensors improve and, in this case, as outdoor locomotion mechanisms improve, there will be greater urgency to solve problems associated with localisation in wide-open settings, with and without GPS-type global localisation sensors.

We end this section with one final open challenge that represents one of the fundamental academic research questions of robotics: sensor fusion. A variety of measurement types are possible using off-the-shelf robot sensors, including heat, range, acoustic and light-based reflectivity, color, texture, friction, etc. Sensor fusion is a research topic closely related to map representation. Just as a map must embody an environment in sufficient detail for a robot to perform localisation and reasoning, sensor fusion demands a representation of the world that is sufficiently general and expressive that a variety of sensor types can have their data correlated appropriately, strengthening the resulting percepts well beyond that of any individual sensor's readings.

Perhaps the only general implementation of sensor fusion to date is that of neural network classifier. Using this technique, any number and any type of sensor values may be jointly combined in a network that will use whatever means necessary to optimise its classification accuracy. For the AMR that must use a human-readable internal map representation, no equally general sensor fusion scheme has yet been born. It is reasonable to expect that, when the sensor fusion problem is solved, integration of a large number of disparate sensor types may easily result in sufficient discriminatory power for robots to achieve real-world navigation, even in wide-open and dynamic circumstances such as a public square filled with people.

5.6 Probabilistic Map-Based Localisation

5.6.1 Introduction

As stated earlier, multiple hypothesis position representation is advantageous because the robot can explicitly track its own beliefs regarding its possible positions in the environment. Ideally, the robot's belief state will change, over time, as is consistent with its motor outputs and perceptual inputs. One geometric approach to multiple hypothesis representation, mentioned earlier, involves identifying the possible positions of the robot by specifying a polygon in the environmental representation [113]. This method does not provide any indication of the relative chances between various possible robot positions. Probabilistic techniques differ from this because they explicitly identify probabilities with the possible robot positions, and for this reason these methods have been the focus of recent research. In the following sections we present two classes of probabilistic localisation. The first class, Markov localisation, uses an explicitly specified probability distribution across all possible robots positions. The second method, Kalman filter localisation, uses a Gaussian probability density representation of robot position and scan matching for localisation. Unlike Markov localisation, Kalman filter localisation does not independently consider each possible pose in the robot's configuration space. Interestingly, the Kalman filter localization process results from the Markov localisation axioms if the robot's position uncertainty is assumed to have a Gaussian form [28 page 43-44]. Before discussing each method in detail, we present the general robot localisation problem and solution strategy. Consider a mobile robot moving in a known environment. As it starts to move, say from a precisely known location, it can keep track of its motion using odometry. Due to odometry uncertainty, after some movement the robot will become very uncertain about its position (see section 5.2.4). To keep position uncertainty from growing unbounded, the robot must localise itself in relation to its environment map. To localise, the robot might use its on-board sensors (ultrasonic, range sensor, vision) to make observations of its environment. The information provided by the robot's odometry, plus the information provided by such exteroceptive observations can be combined to enable the robot to localise as well as possible with respect to its map. The processes of updating based on proprioceptive sensor values and exteroceptive sensor values are often separated logically, leading to a general two-step process for robot position update. Action update represents the application of some action model Act to the mobile robot's proprioceptive encoder measurements o and prior belief state s to yield a new belief s' representing the robot's belief about its current position. Note that throughout this chapter we will assume that the robot's proprioceptive encoder measurements are used as the best possible measure of its actions over time. If, for instance, a differential drive robot had motors without encoders connected to its wheels and employed open-loop control, then instead of encoder measurements the robot's highly uncertain estimates of wheel spin would need to be incorporated. We ignore such cases and therefore have a simple formula:

$$s'_t = \text{Act}(o_t, s_{t-1}) \quad (5.1)$$

Perception update represents the application of some perception model See to the mobile robot's exteroceptive sensor inputs i and updated belief state s' to yield a refined belief s'' representing

the robot's current position:

$$s_t = \text{See}\left(i_t, s'_{t-1}\right) \quad (5.2)$$

The perception model See and sometimes the action model Act are abstract functions of both the map and the robot's physical configuration.¹⁷

17

In general, the action update process contributes uncertainty to the robot's belief about position: encoders have error and therefore motion is somewhat nondeterministic. By contrast, perception update generally refines the belief state. Sensor measurements, when compared to the robot's environmental model, tend to provide clues regarding the robot's possible position. In the case of Markov localisation, the robot's belief state is usually represented as separate probability assignments for every possible robot pose in its map. The action update and perception update processes must update the probability of every cell in this case. Kalman filter localisation represents the robot's belief state using a single, well-defined Gaussian probability density function, and thus retains just a single parameterization of the robot's belief about position with respect to the map. Updating the parameters of the Gaussian distribution is all that is required. This fundamental difference in the representation of belief state leads to the following advantages and disadvantages of the two methods, as presented in [44]:

Part IV

GNU/Linux Operating System

Chapter 6

Welcome to Linux

Table of Contents

6.1 Learning the Linux Command Line	165
6.2 Installation	172

6.1 Learning the Linux Command Line

Working with a text-based [Command Line Environment](#) (CLI), without the graphical user interface can be intimidating at first glance, as most are accustomed to using a graphical user interface (GUI). But understanding the command line environment will show how powerful and efficient it is.

```
1 echo "Hello, Linux!"                                C.R. 1  
1 Hello, Linux!                                         bash
```



```
1 Hello, Linux!                                         text
```

Most senior programmers in the industry and veteran Linux [system administrators](#) will exclusively use CLI as their day to day interaction with the computer. The reason is, the GUI being designed for simplifying human interaction with computers rather than improving the computer's efficiency.

This document aims to introduce the fundamentals of working with the **Linux command line** using a very common shell called Bash as it will be important in the future when working with [ROS](#) (Robot Operating System) or in any future endeavour the reader may pursue in the fields related to computer science.

- Work on what the command line is and how it works,

- Look at working with files and folders,
- How Linux protects files from unauthorised access with permissions,
- Common commands to be familiar with and how to connect commands together with pipes,
- Introduction to some complex command line tasks.

This part of the lecture-book aims to give practical knowledge on working with the widely used Bash shell, in case you choose to extend your learning into user management, network configuration, programming and development, or system administration.

6.1.1 A History of Command Line Interface

The command-line interface came from a form of dialogue once conducted by humans over [teleprinter](#) (TTY) machines, in which human operators remotely exchanged information instead of a human communicating with another human over a teleprinter. Early computer systems often used teleprinter machines as the means of interaction with a human operator. The computer became one end of the human-to-human teleprinter model.

The mechanical teleprinter was replaced by a [terminal](#), a keyboard and screen emulating the teleprinter. "Smart" terminals permitted additional functions, such as cursor movement over the entire screen, or local editing of data on the terminal for transmission to the computer.

As the microcomputer revolution replaced the traditional systems, hardware terminals were replaced by terminal emulators - PC software that interpreted terminal signals sent through the PC's serial ports. These were typically used to interface an organization's new PC's with their existing mini- or mainframe computers, or to connect PC to PC. Some of these PCs were running Bulletin Board System software.



Figure 6.1: Hughes telegraph, an early (1855) teleprinter built by Siemens and Halske. The centrifugal governor to achieve synchronicity with the other end can be seen ([Original](#))

Early operating system CLIs were implemented as part of resident monitor programs, and could not easily be replaced. The first implementation of the shell as a replaceable component was part of the Multics time-sharing operating system. In 1964, MIT Computation Center staff member Louis Pouzin developed the RUNCOM tool for executing command scripts while allowing argument substitution.

Pouzin coined the term "shell" to describe the technique of using commands like a programming language, and wrote a paper about how to implement the idea in the Multics operating system. Pouzin returned to his native



France in 1965, and the first Multics shell was developed by Glenda Schroeder. At Nokia Bell Labs headquarters the first Unix shell, the V6 shell, was developed by Ken Thompson in 1971 and was modelled after Schroeder's Multics shell. The Bourne shell was introduced in 1977 as a replacement for the V6 shell. Although it is used as an interactive command interpreter, it was also intended as a scripting language and contains most of the features that are commonly considered to produce structured programs.

The Bourne shell led to the development of the KornShell (ksh), Almquist shell (ash), and the popular Bourne-again shell (or Bash). Early microcomputers themselves were based on a command-line interface such as CP/M , DOS or AppleSoft BASIC. During the 1980s and 1990s, the introduction of the Apple Macintosh and of Microsoft Windows on PCs saw the command line interface as the primary user interface replaced by the Graphical User Interface. The command line remained available as an alternative user interface, often used by system administrators and other advanced users for system administration, computer programming and batch processing.

```

Terminal
-rwxr-xr-x 1 bin      18296 Jun  8 1979 fsck
-rwxr-xr-x 1 bin      1458  Jun  8 1979 getty
-rw-r--r-- 1 root     49  Jun  8 1979 group
-rwxr-xr-x 1 bin      2482  Jun  8 1979 init
-rwxr-xr-x 1 bin      8484  Jun  8 1979 mkfs
-rwxr-xr-x 1 bin      3642  Jun  8 1979 mknod
-rwxr-xr-x 1 bin      3976  Jun  8 1979 mount
-rw-r--r-- 1 root     141  Jun  8 1979 passwd
-rw-r--r-- 1 bin      366  Jun  8 1979 rc
-rw-r--r-- 1 bin      266  Jun  8 1979 ttys
-rwxr-xr-x 1 bin      3794  Jun  8 1979 umount
-rwxr-xr-x 1 bin      634  Jun  8 1979 update
-rw-r--r-- 1 bin      40 Sep 22 05:49 utmp
-rwxr-xr-x 1 root     4520  Jun  8 1979 wail
# ls -l /*unix*
-rwxr-xr-x 1 sys      53802 Jun  8 1979 /hptunix
-rwxr-xr-x 1 sys      52850 Jun  8 1979 /hptmunix
-rwxr-xr-x 1 root     50990 Jun  8 1979 /rkunix
-rwxr-xr-x 1 root     51982 Jun  8 1979 /rl2unix
-rwxr-xr-x 1 sys      51790 Jun  8 1979 /rphtunix
-rwxr-xr-x 1 sys      51274 Jun  8 1979 /rpptmunix
# ls -l /bin/sh
-rwxr-xr-x 1 bin      17810 Jun  8 1979 /bin/sh
#

```

Figure 6.3: Bourne shell interaction on Version 7 Unix ([Original](#)).

Shells in other Operating Systems

Windows In November 2006, Microsoft released version 1.0 of Windows PowerShell, which combined features of traditional Unix shells with their proprietary object-oriented .NET Framework. MinGW and Cygwin are open-source packages for Windows that offer a Unix-like CLI. Microsoft provides MKS Inc.'s ksh implementation MKS Korn shell for Windows through their Services for UNIX add-on.

Macintosh Since 2001, the Macintosh operating system macOS has been based on a Unix-like operating system called Darwin. On these computers, users can access a Unix-like command-

line interface by running the terminal emulator program called Terminal, which is found in the Utilities sub-folder of the Applications folder, or by remotely logging into the machine using ssh. Z shell is the default shell for macOS (as of macOS Catalina) with bash, tcsh, and the KornShell also provided. Before macOS Catalina, bash was the default.

6.1.2 Linux is a Nutshell

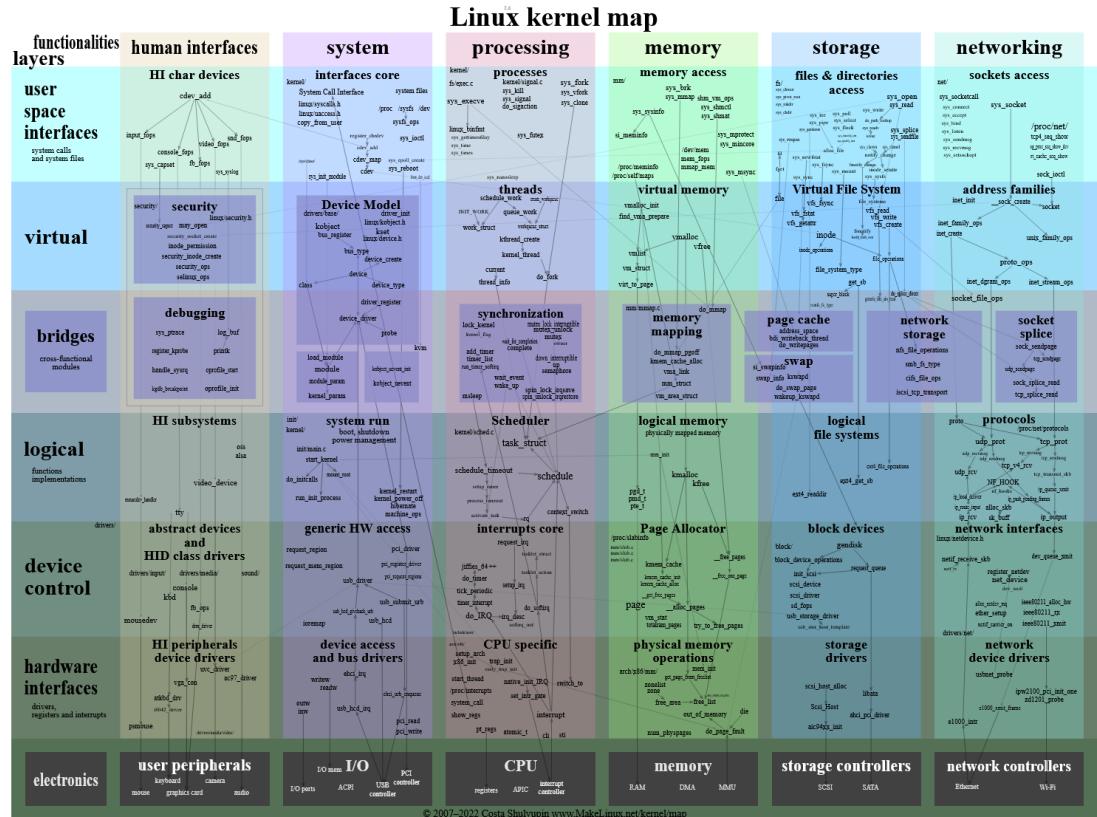


Figure 6.4: The kernel mapping of the Linux operating system.

A Brief Description of What Linux Does

Linux is a general purpose computer operating system, originally released in 1991 by Linus Torvalds and began as a personal project of him [76]. It was to create a new **free** operating system kernel which the resulting kernel has been marked by constant growth throughout its history¹.

Linux is defined by its kernel, called the **Linux kernel**, which is the core component of the system. This kernel interacts with the computer hardware to allow software and other hardware to exchange information, which you can see in **Fig. 6.4**.

Imagine the kernel as the middle-man between your software and the hardware

¹There were alternative OSs on the market such as MINIX but it was under a proprietary license which was later became open-source in 2000

As Linux is an open-source project and is probably one of the greatest collaborative software work in history, it has a rich history. It was inspired by MINIX which, in turn, was inspired by UNIX with UNIX being the first portable operating system ever designed [77] as it was mostly written with the C programming language [78].

Open Source v. Closed Source

In programming there are two (2) main approaches when it comes to sharing code:

- It can be closed source, which means, you are not allowed to edit the code the program is running on,
- Open source which you are free to edit and share the code as you see fit.

Linux is based on a philosophy of software and operating systems being **free**; free of cost and freely modifiable. The software license which allows this, in the case of the Linux kernel, is called the GNU General Public License². This emphasis of freedom, both, of cost and modification has helped Linux to become popular for many different applications and purposes from tinkering programming to being used in massive databases of major companies.

Linux has popped up everywhere from the majority of the servers that run web services we all use, to super computers, to Wi-Fi routers, in cars, mobile phones, and everywhere in between. Odds are that you are closed to a device that uses some part of the Linux kernel. In the midst of all these different kinds of Linux installations, the most important distinction you'll need to be aware of is one of the genealogy of Linux.

²are a series of widely used free software licenses, or copyleft licenses, that guarantee end users the freedoms to run, study, share, or modify the software.

6.1.3 Linux Distributions

While the Linux kernel is more or less the same across nearly all installations of Linux, the software that surrounds the kernel that provides capabilities like *software package management*, *control of services*, and the *location of configuration files* differs between them. Many of the tools that come packaged with Linux come from the GNU Project and aren't actually a part of Linux and, taken together, the combination of the kernel and these common tools is often referred to as **GNU Linux**. Different groups of software and configuration choices that are maintained by individuals or groups of people are called distributions, or distro's. Most major distributions of Linux fall into categories based on the original distribution from which they were derived. These are:

Depending the readers future work or study area, it is likely to end up learning to use the command line on a system that inherits from one of these distributions. Most likely, it will be a distribution derived from Debian or Red Hat. Linux Mint, Ubuntu, Elementary OS, and Kali Linux are all derived from Debian. CentOS, Fedora, and Red Hat Enterprise Linux are derived from Red Hat.

The history of all of these different distributions of Linux is beyond the scope of this document. But,

Distribution	Advantages	Disadvantages
Linux Mint	Superb collection of "minty" tools developed in-house, hundreds of user-friendly enhancements, inclusion of multimedia codecs, open to users' suggestions	The project does not issue security advisories
Ubuntu	Fixed release cycle and support period; long-term support (LTS) variants with five years of security updates; novice-friendly; wealth of documentation, both official and user-contributed	Lacks compatibility with Debian; frequent major changes tend to drive some users away; non-LTS releases come with only nine months of security support
Arch Linux	Excellent software management infrastructure; unparalleled customisation and tweaking options; superb on-line documentation	Occasional instability and risk of breakdown
Gentoo	Highly flexible, endlessly customizable, able to use a range of compile-time configurations, init systems and run on many architectures	Requires a higher degree of knowledge to use, upgrading packages via source can be time consuming
Slackware Linux	Considered highly stable, clean and largely bug-free, strong adherence to UNIX principles	Limited number of officially supported applications; conservative in terms of base package selection; complex upgrade procedure
Debian	Very stable; remarkable quality control; includes over 30,000 software packages; supports more processor architectures than any other Linux distribution	Conservative - due to its support for many processor architectures, newer technologies are not always included; slow release cycle (one stable release every 2 - 3 years); discussions on developer mailing lists and blogs can be uncultured at times
Fedora	Highly innovative; outstanding security features; large number of supported packages; strict adherence to the free software philosophy; availability of live spins featuring many popular desktop environments	Fedora's priorities tend to lean towards enterprise features, rather than desktop usability; some bleeding edge features, such as switching early to KDE 4 and GNOME 3, occasionally alienate some desktop users
openSUSE	Comprehensive and intuitive configuration tool; large repository of software packages, excellent web site infrastructure and printed documentation, Btrfs with boot environments by default	Its resource-heavy desktop setup and graphical utilities are sometimes seen as "bloated and slow"
Red Hat	Long-term, commercial support of ten years or more. Stability.	Lacks latest Linux technologies; small software repositories; licensing restrictions
FreeBSD	Fast and stable; availability of over 24,000 software applications (or "ports") for installation; very good documentation; native ZFS support and boot environments	Tends to lag behind Linux in terms of support for new and exotic hardware, limited availability of commercial applications; lacks graphical configuration tools

Table 6.1: Most popular distributions used according to distrowatch.com

what this means at its core is the need to be aware of what system is in use and the need to adapt to account for differences in distributions. As we begin working with Linux, through the command line, it will be apparent, most of what can be done is the same across the major distributions.

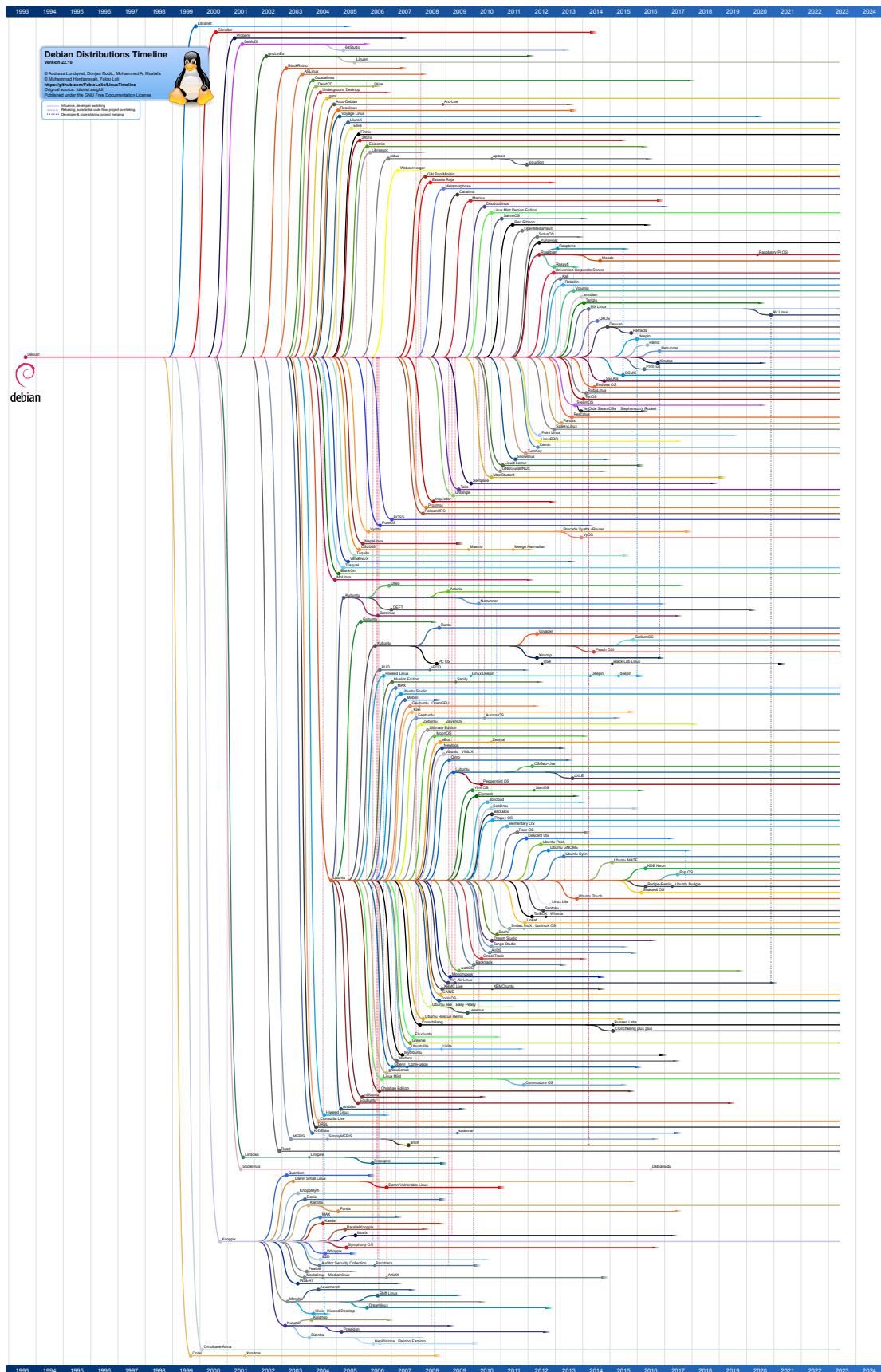


Figure 6.5: A Family tree of the debian branch of linux [79].



Figure 6.6: The docker logo

6.2 Installation

6.2.1 Docker

Docker is an open-source platform that has completely changed the way we develop, deploy, and use apps. The application development lifecycle is a dynamic process, and developers are always looking for ways to make it more efficient. Docker enables developers to package their work and all of its dependencies into standardized units called containers by utilizing containerization technology.

By separating apps from the underlying infrastructure, these lightweight containers provide reliable performance and functionality in a variety of environments. Because of this, Docker is a game-changer for developers because it frees them up to concentrate on creating amazing software rather than handling difficult infrastructure.

Regardless of your level of experience, Docker provides an extensive feature set and a strong toolset that can greatly enhance your development process. In this tutorial, we will provide you with a thorough understanding of Docker, going over its main features, advantages, and ways to use it to develop, launch, and distribute apps more quickly and easily.

Dockerfile

A Dockerfile is a text document in which you can lay down all the instructions that you want for an image to be created. The first entry in the file specifies the base image, which is a pre-made image containing all the dependencies you need for your application. Then, there are commands you can send to the Dockerfile to install additional software, copy files, or run scripts. The result is a Docker image: a self-sufficient, executable file with all the information needed to run an application.

Dockerfiles are a compelling way to create and deploy applications. They help in creating an environment consistently reproducibly, and in an easier way. Dockerfiles also automate the deployment process.

A Dockerfile is used to create new custom images prepared individually according to specific needs. For instance, a Docker image can have a particular version of a web server or, for example, a

database server.

```

1 # DOCKERFILE FOR LINUX LECTURES -----
2 FROM ubuntu
3 RUN apt-get update
4 RUN apt-get update && apt-get install -y \
5     unzip \
6     wget \
7     sudo
8 RUN yes | unminimize
9
10 ARG
11   → ZIPFILE="https://github.com/dTmC0945/L-MCI-BSc-Mobile-Robotics/raw/refs/heads/main/datasets/linux-tutorials.zip"
12
13
14 # Create some user directories to simulate a desktop environment
15 RUN mkdir -p \
16     /home/student/Downloads \
17     /home/student/Desktop \
18     /home/student/Desktop
19
20 # Download the tutorial files to their correct place
21 RUN cd "/home/student/Desktop" && \
22     wget -O "linux.zip" "$ZIPFILE" && \
23     unzip "linux.zip" && \
24     rm "linux.zip"

```

Important Instructions

A Dockerfile is a text document that includes all the different steps and instructions on how to build a Docker image. The main elements described in the Dockerfile are the base image, required dependencies, and commands to execute application deployment within a container.

The essential instructions of a Dockerfile are illustrated below

FROM This instruction sets the base image on which the new image is going to be built upon. It is usually the first instruction in a Dockerfile.

```
RUN apt-get update && apt-get install -y \
```

RUN This will be an instruction that will be executed for running the commands inside the container while building. It typically can be utilized to install an application, update libraries, or do general setup.

Part V

Robot Operating System

Chapter 7

Installation

Table of Contents

7.1	Introduction	177
7.2	Installing ROS Humble Hawksbill	178

7.1 Introduction

Setting up the Locale

Make sure you have a locale which supports UTF-8¹. If you are in a minimal environment (such as a docker container), the locale may be something minimal like POSIX. We test with the following settings. However, it should be fine if you're using a different UTF-8 supported locale.

```
1  locale # check for UTF-8
2
3  sudo apt update && sudo apt install locales
4  sudo locale-gen en_US en_US.UTF-8
5  sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
6  export LANG=en_US.UTF-8
7
8  locale # verify settings
```

C.R. 1
bash

¹UTF-8 is a character encoding standard used for electronic communication. Defined by the Unicode Standard, the name is derived from Unicode Transformation Format - 8-bit. Almost every webpage is stored in UTF-8.

7.2 Installing ROS Humble Hawksbill

Before we can begin working with ROS, we must install all the necessary files and dependencies required. For these lectures we will install ROS 2 Humble Hawksbill which is currently available for Ubuntu Jammy (22.04 LTS).

While currently there are more up-to-date version of ROS 2 available, due to its long term support and established compatibility, we will be using ROS 2 Humble.

It is **heavily** recommended to be on Ubuntu 22.04 LTS as ROS 2 Humble is only officially supported on this version. It is possible to compile ROS 2 on other Ubuntu or Linux distributions, however, you need to compile the binaries yourself.

7.2.1 Set locale

- ²i.e., a docker container. Make sure you have a locale which supports **UTF-8**. If you are in a minimal environment², the locale may be something minimal like **POSIX**. We test with the following settings. However, it should be fine if you're using a different **UTF-8** supported locale. To start with installation, if you are using a Graphical User Interface (GUI) open up your terminal (**Ctrl+Alt+T**).

1 `sudo apt install software-properties-common`
2 `sudo add-apt-repository universe`

C.R. 2

bash

Information: UTF-8

A variable-length character encoding standard used for electronic communication.

Information: POSIX

A family of standards specified for maintaining compatibility between operating systems.

Information: Docker

A set of platform as a service products that use OS-level virtualization to deliver software in packages.

7.2.2 Setup Sources

You will need to add the ROS 2 `apt3` repository to your system.

First ensure that the Ubuntu Universe repository is enabled.

1 `sudo apt install software-properties-common`
2 `sudo add-apt-repository universe`

C.R. 3

bash

Now add the ROS 2 GPG key with apt.

```
1 sudo apt update && sudo apt install curl -y
2 sudo curl -sSL \
3     https://raw.githubusercontent.com/ros/rosdistro/master/ros.key \
4     -o /usr/share/keyrings/ros-archive-keyring.gpg
```

C.R. 4
bash

Then add the repository to your sources list.

```
1 echo "deb [arch=$(dpkg --print-architecture) \
2     signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] \
3     http://packages.ros.org/ros2/ubuntu \
4     $(. /etc/os-release && echo $UBUNTU_CODENAME) main" | \
5     sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
```

C.R. 5
bash

echo A command that outputs the strings that are passed to it as arguments.

7.2.3 Install ROS 2 packages

Update your apt repository caches after setting up the repositories. ROS 2 packages are built on frequently updated Ubuntu systems. It is always recommended that you ensure your system is up to date before installing new packages.

```
1 sudo apt update && sudo apt upgrade
```

C.R. 6
bash

Desktop Install (Recommended): ROS, RViz, demos, tutorials. Development tools: Compilers and other tools to build ROS packages

```
1 sudo apt install ros-foxy-desktop python3-argcomplete
```

C.R. 7
bash

If you have read the document before doing copy and pasting, you can use the `rosinstall.sh` provided to you to automatically install everything required for this course.

```
1 sudo apt install ros-foxy-desktop python3-argcomplete
```

C.R. 8
bash

7.2.4 Setting up the Environment

Set up your environment by sourcing the following file in your terminal (`Ctrl+Alt+T`).

```
1 # Replace ".bash" with your shell if you're not using bash
2 # Possible values are: setup.bash, setup.sh, setup.zsh
```

C.R. 9
bash

```
3 source /opt/ros/humble/setup.bash
```

C.R. 10

bash

Alternatively, if you prefer to automate this action, simply type the following code in your terminal:

The aforementioned command simply writes the command to a document called `bashrc` which is a script running when the OS boots up.

The `.bashrc` file is a script file that's executed when a user logs in. The file itself contains a series of configurations for the terminal session. This includes setting up or enabling: coloring, completion, shell history, command aliases, and more.

It is a hidden file and simple `ls` command won't show the file.

```
1 #!/bin/bash
2
3 # First check your Ubuntu Version
4 # For maximum compatibility with ROS it needs to be 22.04 LTS
5
6 # Creating log for troubleshooting
7 echo "##### BEGIN ATTEMPT #####">>install_log.txt
8
9 echo "Welcome to ROS 2 Automated Installation"
10 echo ""
11 echo ""
12
13 # Accessing the Ubuntu version using AWK and piping it to grep for Regex
14 version=$(
15     awk '/VERSION_ID/' IGNORECASE=1 /etc/*release |
16         grep -Eo "[[:digit:]]+([.][[:digit:]]+)?"
17 )
18
19 # Checks version for ROS Compliance
20 if [[ "${version}" == *"22.04"* ]]; then
21     echo "Version is supported."
22     sleep 1
23     echo "Continuing installation..."
24     sleep 1
25 else
26     echo "Your version: ${version}, What is needed: 22.04"
27     echo "I am sorry but your version is not supported."
28     echo "This install script will terminate"
29     exit
30
31 fi
32
33 echo ""
34 echo "Installing UTF-8 Compliance ..."
35
36 {
37     locale # check for UTF-8
```

C.R. 11

bash

```

38
39     sudo apt update
40     sudo apt install locales
41     sudo locale-gen en_US en_US.UTF-8
42     sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
43     export LANG=en_US.UTF-8
44
45     locale # verify settings
46 } &>install_log.txt
47
48 echo ""
49 echo "Enabling Ubuntu Universe Repositories..."
50
51 {
52     sudo apt install software-properties-common
53     echo | sudo add-apt-repository universe
54 } &>install_log.txt
55
56 echo ""
57 echo "Adding ROS 2 GPG Keys ..."
58
59 {
60     sudo apt update
61     sudo apt install curl -y
62     sudo curl -sSL \
63         https://raw.githubusercontent.com/ros/rosdistro/master/ros.key \
64         -o /usr/share/keyrings/ros-archive-keyring.gpg
65 } &>install_log.txt
66
67 echo ""
68 echo "Adding ROS 2 to repository ..."
69
70 {
71     echo "deb [arch=$(dpkg --print-architecture) \
72 signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] \
73 http://packages.ros.org/ros2/ubuntu \
74 $(. /etc/os-release && echo $UBUNTU_CODENAME) main" \
75         | sudo tee /etc/apt/sources.list.d/ros2.list >/dev/null
76 } &>install_log.txt
77
78 echo ""
79 echo "Getting Updates ..."
80
81 {
82     sudo apt update
83     echo yes | sudo apt upgrade
84 } &>install_log.txt
85
86 echo ""
87 echo "Installing ROS ..."
88
89 {
90     echo yes | sudo apt install ros-humble-desktop

```

C.R. 12

```
91     yes | sudo apt install ros-dev-tools  
92 } &>install_log.txt  
93  
94 {  
95     sudo apt install dbus-x11  
96 } &>install_log.txt  
97  
98 echo ""  
99 echo "Sourcing ROS file ..."  
100 sleep 1  
101  
102 echo "source /opt/ros/humble/setup.bash" >~/.bashrc  
103  
104 echo ""  
105 echo "Removing unnecessary files ..."  
106 sleep 1  
107 {  
108     yes | sudo apt autoremove  
109 } &>install_log.txt
```

List of Acronyms

AMR Autonomous Mobile Robotics. ix, 27–30, 32–37, 39–43, 45, 49, 55–57, 59, 61, 137–142, 145, 147, 148, 152–154, 156–159

CCD Charge Coupled Device. viii, 28, 32, 47, 51–56, 140

CMOS Complimentary MOS. viii, 47, 51, 52, 55, 56

DoF Degrees of Freedom. vii, 7, 10–12

GNSS Global Navigation Satellite System. 139

GPS Global Positioning System. 38, 139

GUI Graphical User Interface. 178

LED Light-Emitting Diode. 44

LHS Left Hand Side. 86

LIDAR Light Detection and Ranging. 43

PSD Position Sensing Device. 47, 48

RHS Right Hand Side. ix, 86, 117

SNR Signal-to-Noise Ratio. 140

ToF Time-of-Flight. 40, 41, 43

Bibliography

- [1] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.
- [2] Michael LaBarbera. "Why the wheels won't go". In: *The American Naturalist* 121.3 (1983), pp. 395–408.
- [3] Julian FV Vincent et al. "Biomimetics: its practice and theory". In: *Journal of the Royal Society Interface* 3.9 (2006), pp. 471–482.
- [4] Fran ccois Druelle et al. "Convergence of bipedal locomotion: why walk or run on only two legs". In: *Convergent Evolution: Animal Form and Function*. Springer, 2023, pp. 431–476.
- [5] Damian M Lyons and Kiran Pamnany. "Rotational legged locomotion". In: *ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005*. IEEE. 2005, pp. 223–228.
- [6] G Schweitzer. "ROBOTRAC-a Mobile Manipulator Platform for Rough Terrain". In: *Proc. of Int. Symp. on Advanced Robot Technology*. 1991, pp. 411–416.
- [7] Mathias Thor et al. "A dung beetle-inspired robotic model and its distributed sensor-driven control for walking and ball rolling". In: *Artificial Life and Robotics* 23 (2018), pp. 435–443.
- [8] Z. P. Square R. Jones. *All Praise The Humble Dung Beetle*. 2018. URL: <https://www.smithsonianmag.com/science-nature/the-humble-dung-beetle-180967781/>.
- [9] Sharp Photography. *Common ostrich (Struthio camelus australis) male running (composite image), Damaraland, Namibia*. 2018. URL: [https://commons.wikimedia.org/wiki/File:Common_ostrich_\(Struthio_camelus_australis\)_male_running_composite.jpg](https://commons.wikimedia.org/wiki/File:Common_ostrich_(Struthio_camelus_australis)_male_running_composite.jpg).
- [10] Porges. *Zebra in Wellington Zoo*. 2025. URL: https://commons.wikimedia.org/wiki/File:Zebra_sideview.jpg.
- [11] Encyclopaedia Britannica. *Ant*. 2025. URL: <https://cdn.britannica.com/42/223142-050-7033F421/Red-ant-on-a-green-branch.jpg>.
- [12] Zhanbing Song et al. "The Impact of Exercise Play on the Biomechanical Characteristics of Single-Leg Jumping in 5-to 6-Year-Old Preschool Children". In: *Sensors* 25.2 (2025), p. 422.
- [13] Sven Böttcher. "Principles of robot locomotion". In: *Proceedings of human robot interaction seminar*. 2006.
- [14] Andrzej Krzywinski, Anna Niedbalska, and L Twardowski. "Growth and development of hand reared fallow deer fawns." In: *Acta theriologica* 29.29 (1984), pp. 349–356.
- [15] John Brackenbury. "Caterpillar kinematics". In: *Nature* 390.6659 (1997), pp. 453–453.

- [16] José L Pons. *Wearable robots: biomechatronic exoskeletons*. John Wiley & Sons, 2008.
- [17] William P Zyhowski, Sasha N Zill, and Nicholas S Szczechinski. "Adaptive load feedback robustly signals force dynamics in robotic model of *Carausius morosus* stepping". In: *Frontiers in Neurorobotics* 17 (2023), p. 1125171.
- [18] David A Winter. *Biomechanics and motor control of human gait: normal, elderly and pathological*. 1991.
- [19] Francesco Lacquaniti, Yuri P Ivanenko, and Myrka Zago. "Patterned control of human locomotion". In: *The Journal of physiology* 590.10 (2012), pp. 2189–2199.
- [20] Hyunglae Lee and Neville Hogan. "Investigation of human ankle mechanical impedance during locomotion using a wearable ankle robot". In: *2013 IEEE International Conference on Robotics and Automation*. IEEE. 2013, pp. 2651–2656.
- [21] R McN Alexander. "Optimization and gaits in the locomotion of vertebrates". In: *Physiological reviews* 69.4 (1989), pp. 1199–1227.
- [22] MIT. *The Raibert Hopper*. 1984. URL: http://www.ai.mit.edu/projects/leglab/robots/3D_hopper/3D_hopper.html.
- [23] Citizendum. *Asimo*. 2005. URL: <https://citizendum.org/wiki/ASIMO>.
- [24] Seshashayee S Murthy and Marc H Raibert. "3D balance in legged locomotion: modeling and simulation for the one-legged case". In: *ACM SIGGRAPH Computer Graphics* 18.1 (1984), pp. 27–27.
- [25] Marc H Raibert, H Benjamin Brown Jr, and Michael Chepponis. "Experiments in balance with a 3D one-legged hopping machine". In: *The International Journal of Robotics Research* 3.2 (1984), pp. 75–92.
- [26] Ben Brown and Garth Zeglin. "The bow leg hopping robot". In: *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*. Vol. 1. IEEE. 1998, pp. 781–786.
- [27] Robert Ringrose. "Self-stabilizing running". In: *Proceedings of International Conference on Robotics and Automation*. Vol. 1. IEEE. 1997, pp. 487–493.
- [28] Flamingo. *Spring Flamingo*. 2000. URL: http://www.ai.mit.edu/projects/leglab/robots/Spring_Flamingo/Spring_Flamingo.html.
- [29] Ilon Bengt Erland. "Rad fuer ein laufstabiles, selbstfahrendes fahrzeug". In: *German Patent No. DE2354404A1* (1974).
- [30] Kevin Dowling et al. "NAVLAB An Autonomous Navigation Testbed". In: *Vision and Navigation: The Carnegie Mellon Navlab*. Springer, 1990, pp. 259–282.
- [31] ackerman. *Ackerman Steering Mechanism*. 2025. URL: <https://www.dubizzle.com/blog/cars/ackerman-steering-mechanism/>.
- [32] Farnell. *Rotary Encoder, Module, Optical, Incremental, 500 PPR, 0 Detents, Vertical, Without Push Switch*. 2025. URL: <https://at.farnell.com/en-AT/broadcom-limited/aedb-9140-a13/encoder-3channel-500cpr-8mm/dp/1161087>.

- [33] Flyrobo. *GY-26 Digital Electronic Compass Sensor Module*. 2025. URL: <https://www.flyrobo.in/gy-26-digital-electronic-compass-sensor-module>.
- [34] FindLight. *Optical Gyroscopes: Measuring Rotational Changes With Sagnac Effect*. 2025. URL: <https://www.findlight.net/blog/optical-gyroscopes-measuring-rotations/>.
- [35] PiHut. *HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi*. 2025. URL: <https://the-phihut.com/blogs/raspberry-pi-tutorials/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi>.
- [36] Reinhold. *Structured light sources on display at the 2014 Machine Vision Show in Boston*. 2014. URL: https://commons.wikimedia.org/wiki/File:Structured_light_sources.agr.jpg.
- [37] Andrzej. *CCD image sensor SONY ICX493AQA 10,14 (Gross 10,75) M pixels APS-C 1.8" 28.328mm (23.4 x 15.6 mm) from module IS-026 from digital camera SONY DSLR-A200 or DSLR-A300 sensor side*. 2014. URL: https://commons.wikimedia.org/wiki/File:CCD_SONY_ICX493AQA_sensor_side.jpg.
- [38] TeledyneCCD. *How a Charge Coupled Device (CCD) Image Sensor Works*. 2020. URL: https://www.teledyneimaging.com/media/1300/2020-01-22_e2v_how-a-charge-coupled-device-works_web.pdf.
- [39] K. Hirakawa and T.W. Parks. "Chromatic adaptation and white-balance problem". In: *IEEE International Conference on Image Processing 2005*. Vol. 3. 2005, pp. III-984. DOI: [10.1109/ICIP.2005.1530559](https://doi.org/10.1109/ICIP.2005.1530559).
- [40] Fstoppers. *Is There a Difference Between Color Temperature and White Balance?* 2022. URL: <https://fstoppers.com/natural-light/there-difference-between-color-temperature-and-white-balance-596031>.
- [41] Adrian Ilie and Greg Welch. "Ensuring color consistency across multiple cameras". In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 2. IEEE. 2005, pp. 1268–1275.
- [42] Teledyne. *Saturation and Blooming*. 2025. URL: <https://www.photometrics.com/learn/imaging-topics/saturation-and-blooming>.
- [43] Zhen Zhang et al. "Analysis and simulation to excessive saturation effect of CCD". In: *2nd International Symposium on Laser Interaction with Matter (LIMIS 2012)*. Vol. 8796. SPIE. 2013, pp. 96–102.
- [44] Baumer. *Operating principle and features of CMOS sensors*. 2025. URL: <https://www.baumer.com/int/en/service-support/function-principle/operating-principle-and-features-of-cmos-sensors/a/EMVA1288>.
- [45] econ Systems. *CMOS camera module*. <https://www.directindustry.com/prod/e-con-systems/product-168594-2365044.html>. URL: <https://www.directindustry.com/prod/e-con-systems/product-168594-2365044.html>.
- [46] TS Holst and GC Lomheim. *CMOS/CCD Sensors*. JCD publishing, 2007.
- [47] Mdf. *A photon noise simulation*. 2010. URL: <https://commons.wikimedia.org/wiki/File:Photon-noise.jpg>.

- [48] Mark-j. *Open Camera Blog*. 2018. URL: <https://sourceforge.net/p/opencamera/blog/2018/09/focus-bracketing-with-open-camera/>.
- [49] David C Hoaglin, Frederick Mosteller, and John W Tukey. *Understanding robust and exploratory data analysis*. Vol. 76. John Wiley & Sons, 2000.
- [50] leonbloy (<https://math.stackexchange.com/users/312/leonbloy>). *Maximum Likelihood Estimate and Second derivative test?* Mathematics Stack Exchange. URL:<https://math.stackexchange.com/q/2241123> (version: 2017-04-19). eprint: <https://math.stackexchange.com/q/2241123>. URL: <https://math.stackexchange.com/q/2241123>.
- [51] Rust John. "Maximum likelihood estimation of discrete control processes". In: *SIAM journal on control and optimization* 26.5 (1988), pp. 1006–1024.
- [52] Yacine Aït-Sahalia and Robert Kimmel. "Maximum likelihood estimation of stochastic volatility models". In: *Journal of financial economics* 83.2 (2007), pp. 413–452.
- [53] Kubrom Hisho Teka. "Parameter estimation of the Black-Scholes-Merton model". In: (2013).
- [54] William CL Stewart and Elizabeth A Thompson. "Improving estimates of genetic maps: a maximum likelihood approach". In: *Biometrics* 62.3 (2006), pp. 728–734.
- [55] Aravinda Chakravarti, Laura K Lasher, and Jillian E Reefer. "A maximum likelihood method for estimating genome length using genetic linkage data." In: *Genetics* 128.1 (1991), pp. 175–182.
- [56] Jerzy Neyman. "On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection". In: *Breakthroughs in statistics: Methodology and distribution*. Springer, 1992, pp. 123–150.
- [57] Jerzy Neyman. "Outline of a theory of statistical estimation based on the classical theory of probability". In: *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences* 236.767 (1937), pp. 333–380.
- [58] Jerzy Neyman and Egon Sharpe Pearson. "IX. On the problem of the most efficient tests of statistical hypotheses". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231.694-706 (1933), pp. 289–337.
- [59] UoW. *How Mars rovers use artificial intelligence*. 2023. URL: <https://online.wlv.ac.uk/how-mars-rovers-use-artificial-intelligence/>.
- [60] Tim Bailey. "Mobile robot localisation and mapping in extensive outdoor environments". PhD thesis. Citeseer, 2002.
- [61] Sean Campbell et al. "Where am I? Localization techniques for mobile robots a review". In: *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*. IEEE. 2020, pp. 43–47.
- [62] Victoria J Hodge. "Sensors and data in mobile robotics for localisation". In: *Encyclopedia of Data Science and Machine Learning* (2023), pp. 2223–2238.
- [63] Omar Jaradat et al. "Challenges of safety assurance for industry 4.0". In: *2017 13th European Dependable Computing Conference (EDCC)*. IEEE. 2017, pp. 103–106.

- [64] David Filliat and Jean-Arcady Meyer. "Map-based navigation in mobile robots:: I. a review of localization strategies". In: *Cognitive systems research* 4.4 (2003), pp. 243–282.
- [65] John J Leonard and Hugh F Durrant-Whyte. *Directed sonar sensing for mobile robot navigation*. Vol. 175. Springer Science & Business Media, 2012.
- [66] Nicola Bellotto and Huosheng Hu. "People tracking and identification with a mobile robot". In: *2007 International Conference on Mechatronics and Automation*. IEEE. 2007, pp. 3565–3570.
- [67] Mohan Sridharan and Peter Stone. "Color learning and illumination invariance on mobile robots: A survey". In: *Robotics and Autonomous Systems* 57.6-7 (2009), pp. 629–644.
- [68] Diego Galar and Uday Kumar. "Chapter 1 - Sensors and Data Acquisition". In: *eMaintenance*. Ed. by Diego Galar and Uday Kumar. Academic Press, 2017, pp. 1–72. ISBN: 978-0-12-811153-6. DOI: <https://doi.org/10.1016/B978-0-12-811153-6.00001-4>. URL: <https://www.sciencedirect.com/science/article/pii/B9780128111536000014>.
- [69] David R Williams. "Aliasing in human foveal vision". In: *Vision research* 25.2 (1985), pp. 195–205.
- [70] maksim. *An example of a poorly sampled brick pattern*. 2006. URL: https://commons.wikimedia.org/wiki/File:Moire_pattern_of_bricks_small.jpg.
- [71] Gaddi Blumrosen, Ben Fishman, and Yossi Yovel. "Noncontact wideband sonar for human activity detection and classification". In: *IEEE Sensors Journal* 14.11 (2014), pp. 4043–4054.
- [72] Angelo M Sabatini and Valentina Colla. "A method for sonar based recognition of walking people". In: *Robotics and Autonomous Systems* 25.1-2 (1998), pp. 117–126.
- [73] Parag H Batavia and Illah Nourbakhsh. "Path planning for the Cye personal robot". In: *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*. Vol. 1. IEEE. 2000, pp. 15–20.
- [74] J Reuter. "Scan-and featurebased multiple hypothesis tracking for mobile robot localization: a data fusion approach". In: *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)*. Vol. 4. IEEE. 1999, pp. 714–719.
- [75] Péter Fankhauser, Michael Bloesch, and Marco Hutter. "Probabilistic terrain mapping for mobile robots with uncertain localization". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3019–3026.
- [76] Oded Koren. "A study of the Linux kernel evolution". In: *ACM SIGOPS Operating Systems Review* 40.2 (2006), pp. 110–112.
- [77] Maurice J Bach. "The Design of the UNIX". In: *RTM. Operating system Prentice Hall* (1986), pp. 312–329.
- [78] Steven C Johnson and Dennis M Ritchie. "UNIX time-sharing system: Portability of C programs and the UNIX system". In: *The Bell System Technical Journal* 57.6 (1978), pp. 2021–2048.
- [79] A. Lundqvist. *Timeline of Debian Linux and related projects*. 2012. URL: <https://commons.wikimedia.org/wiki/File:DebianFamilyTree1210.svg>.

