Request an intellectual property (IP) licence

Date: **13 December 2024**
Time: **15:23**
This form has been sent to the Metropolitan Police via the Single Online Home reporting service.

MIP-619-24-0100-000

Your details

Your details

Title
Mr

First name
tshingombe

Surname
tshitadi

Company name
tshingombe

Email address
tshingombefiston@gmail.com

Phone number
0725298946

Your request

Your request

Select the option that most applies to you
Request an intellectual property (IP) licence to use a trademark belonging to the Met or Mayor's Office for Policing and Crime (MOPAC) for any purpose

Details of your enquiry

ATLATIC INTERNATIONAL UNIVERSITY

1. Circulum design format offline.
-Name : tshingombe Tshitadi

-course title| course objectives|| course description||| activity to carry out |||| ID source of date ||||| bibliography.
-

course title course objectives course description activity to carry out ID source of ibliography

1- Proposal of thesis content / final project
Content

1.Name of thesis : implementation and framework national qualification and national trade examination circulum experimental job theoretical pratical college and government policy LMS in engineering studies science electrical businesses module: case studies rsa in dhet,saqa , St peace college
2. Index: topic achieve research advance field basic field , essential filling research circulum, fundation intermediate,elementaire
3.Introduction : the core and research advanced field experience of sciences engineering electrical study and implement programme in social education and industrial trade vocational career productu sector in energy electrical and science engineering field system need to learn and re implement system information management system sector opportunity and through activities investment horizontal creation of equitable distribution: transformer science engineering and electrical product method learn capacity generative intelligence systems of linear regression models machine learning model for specific results reported that they haveA Mon other aspirations Isreal parameter real power factor and Imagineer power factor ,, need to resolved system exper and artificial intelligence system rural development system residential dispatch deployment system and framework qualification mean regulation humain resource and material work trade design career center to make system LMS factor adaptation between robot science trade elementary work trainer training phase products and systems industrial generator entrepreneurs in same order phase assessment news field and compensation.problem ask rural development need new training order framework to qualicafition requested requalification redesign equivalents system , occupation framework system between national framework qualifications instituts and national trading sector licensed theory and practical in nature and creative abilities,
-typical evry country or landscape will be in a constant state of design system in ,,,,
Large measure unpredictable and this city or village at different paint of time ,, implementation the Grove years of failed turound ..
4.desceiption :at the heart of solutions to framework qualicafition and national trade implementation sub sector training trainer experiemental work place industrial more student and instituts college trade years external internal work value increase price macro economics instability Crete ,.sice accentuated by advertising shortage high inflation levek rising unemployment capacity industrial trademarks society system and materials adequately support trade training QMS system information commissioner,to under utilities in the address desterious policy design implementation ,
5. General analysis: in order to break the successful it has become social contract principle in

-6 current information:
In working to formatted a trade framework qualicafition and national framework and career skill sector trade seta in same system in order to resolve problem impact real to dispatch electrical system real ,work trade design
For the turnaround ,the following
- objective.
- the diagnosis the fundamental strategies instituts framework qualicafition national equivalent national trade international sector approval occupation council trade council engineering sector portal career design to synchronise system adaptative sector LMS learner engineering competition grade post senior principal, engineering electrical ,tradesman wire ,cadet minim system up date successful system in design grade operational, framework award qualifition research undertake material test week conductor atom technical engineering innovation learn teach research mark method marks need to implement adaptative system , research topics circulum regulation irregularity material script, backlog system , combination system ,printer and system need to make synchronise system deploy generative job framework undercover job in next generation must going

- to discern and isolate the sicio economic environment engineering system trade safety security police , commissioner trade need to meet requirements qualicafition framework and the framework must also show in the social successful but framework it increases by outage loadshedding and social down to declined empirical experiemental in other contemporary ,the regret filled job no successful for time table printer system or computers system experiemental make design advanced research ,
-7. discussion the objective is to explore that strategies and situation where Rapide performance import. Trade theory..
- conclusion:
Whilst the field of strategy has be explored extensively in vast to trade framework qualifications need to requalification system was temporarily qualify expire system in job work sector training and regulations system industrial system need cpd to continue system and subject short and gate more skill job was slow operational field basic in basic was poorly no attandance system advance essential field job make support frame commissioner no meeting system trade retrade was not in the same ways Orders orientation industrial, imperative hard, largely ,the research interest and how a fruit full common,ground can be established.
- one of the critical virtues of the proposal thesis that it Engineering electrical science make in order to stabilize thought transfer the vei ld consensus building in ,,
- the thesis is ,, model design
Policy commissioner vs learn vs teacher vs ,, framework national trade vs company property intellectuel business electrical system need to meeting...wrong model design topic ,, research rural energy design framework , and orientation system learner teach career mentor faciltor purpose framework,leaver school need to meeting,
Design two g city design systeme economic revenue bank system portal need sector trade to work in place electrical designer b Poste trade case research job workplace resulted was recruited need printer pool position rank no waiting

- 8 bibliography:

- tshingombe 2023_2924 < Poe's published,,educ technology, magazine net database, St peace college.
Record book completed
- web TVET dhet ,saqa wab
- alu

Graduation procedure form . congratulations programme , diploma .
-1 data verification.
- grade | description| point | numeracy

2

-2. Basic questionnaire exam test
Class

AIU .
-Academic evaluation questionnaire , videoconference:
-A.I.U|education|| domination|||emphasis|||| specifications||||| professional.

3.curculum course ,
Assessment
-3.1.title of the subject :
engineering electrical master
-3 2 terminanal objective of the course :
Engineering electrical master basic advance field studies assignment to able capable to define to design creativity fundamental system master low skills and knowledge value compete with each section shall be responsible for delivering the best regards in electrostatic electrodynamics electromagnetic and value of power systems.
- 3.3..brief description : the course electrical power system use or business in trade theory pratical system to master system value more stability of movement quantum mechanics transformation of electrostatic dynamic low stability,relativity of charge celerity basic and advance in trade theory electrical low Commissioning and approval: low change rules change phenomenon fundamental by stress of movement rupture breaking electrical system synchronise system asynchronous linearization system,in trade theory electrical and industrial electronics basic advance power
3.4.synopsis of content: the stability design projection system trade marketing board information system electrokinematic dynamic physical state engineering science introduction used to trade theory electrical ,manufacture process inventory low stamp system low stable loadshedding week manufacture industrial technology linearization system.
-3.5 activities of course :
Activity engineering electrical electrical experiemental subject completed log Engineering studies work 3dimension multidisciplinary approach logic of this claim: information management system in education and learn trade facilitation

Discussion log : completed theory pratical physic experiemental panel trade ,, experiemental input and output system
Activity: manipulation: test electrostatic
Conductivity expension linearization system ,dynamic system test insulation conductivity low rules , derivatives limited integrally sum resulted test system evaluation framework.
Critical source
3.5 .source of data :
Experiemental topics St peace college tshingombe ,web PG
3.6 bibliography:
Tshingombe .

4.Assignment :
Title page: engineering electrical master
Electrostatic electrokinematic electrodynamics electromagnetic, stability power systems ,,,process control ,,in trade theory pratical manufacture process. Inventory claim
- index :
- page :
Cover the ,7 basic
Question course
Wath means
- diagrams: scheme correlative matrices and comparative matrices :
Answer:
- deepening of the subject : engineering electrical master low phenomenology studies vibration system.
- pratical example and cases .: engineering electrical cases study city power scheneider Eskom. Loadshedding power and industrial dtic trade career hr

- justification:

- level experience :

- how the treated subject is seen at the local regional
-advantage and disadvantages,.
Poor efficiency and poor distribution of system ,, in trade close tendered system
Big system most important consumers system in trade increase coat award ..

No master number real system imaging

---

5. Topics.
Table of contents:
5.1: Introduction purpose of topics
Definition rationale:
5.2 description:
Components of the topics
5.3.general analyse :
- 5.4. actualization : case study.
5.5 . discussion:
5.6 general recommendation .
5.7 : suggestions.
Conclusion news perspective

- 6 topics in electrical engineering,MS ,MSEE..
- topic 6.1: digital telephonic
Introduction purpose of definition

- topic 6.2: space control system.

- topic 6.3 . advanced telecommunication.
-topic 6.4: wireless telecommunications systems.
- topic 6.5: neural networks.
-topic 6.6: computation and biologic
-topic 6.7: knowledge base system in electrical.
- topic 6.8: principle of internetworking.
- topics 6.9: optical fibre ,
- topics 6.10: signal detection and estimation theory .
- topics 6.11: digital control system.
Topics 6.12 microprocess system .
- topics 6.13 introduction to stochastic process : movement aleatoi ,signal redresseur assessvisa system band etroite , signal note .
-topic6,14 optical and ultrasound ,tomographic ,,supersoun u

Propagation linear celerity movement incidence ..
Topic : 6:15 industrial power systems process ,,
Signal input output functions power
Topics : 6:16 . signal detection and estimation theory digital images reconstruction and medical imagine
- topic 6:17, process integration
- topics 6;18.parallels computer architecture .
Topic.6:19. architecture computer
-
Topic 6:20 . power systems control stability.
Topic 6.21: electromagnetic
Topic 6,22 mathematics ,statistic probability,, calculus ,,binary
Physic ,..

Orientation course.
- topics 6:22.communicatiin , investigation comphrensive
- topics6:23.. organization's theory Portofilio
-topics 6.24. experiemental learning , autobiography.
- topic 6.25 ,academic questions evaluation evaluation .
- topic ,6,25 fundamental of knowledge integration.
- topics fundamental principles phylosophie education.
- professional evaluation development evaluation
- development of graduation studiy
Master skill development long

approfondis kinematics system phase transition phase education system specialist personal care education facilities,, phenomenon city

Topic
. Topics.
Table of contents:

3

**- defensive scope process , applicability
Claim system
Thesis
Overview; education trade
Key**

**-brigades vs private security public safety police goverment student police army order public police CA safety ,vs student portofy police metropolitan student student case government thermie vs securite gov study .
-* overview ,key topics prospectus university operational task requirements criteria college university natural summarise key trade abstract phylosophie concept trade concept definitely extension trade and understand trade design comphrensive trade design comphrensive trade concept vs trade theory college requirements basic task construction partie trade explanation low rules university trade overview idea univer , industrial thesis work undragogie concept.
* Applied trade to resolve trade, applied sciences math work operational applied vocational national framework sciences math work operational applied continue university institute trade low rules.
- key , overview abstract trade concept trade theory electric conception ,idea phylosophie education trade undragogie idea axiom argument resonement univer summary application vs college scope.
- trade submission mission applied trade to supply.
- abstraction , metaphysical metaform transformer trade university vs College purpose that requirements basic principle installation that career vs university.
- Vs e cpd diploma trade continue Scotland continue diploma trade certificate master degree construction master degree , professional supplemtaire continue vs diploma graduate continue integration and master degree short not professional skills development degree discovery career center master tlc technical learner college diploma and master degree diploma building electrical master businesses please can see satellite, combination cpd training job the don't want to vocational cashier and ncv and relate ,and hr w.
* Distance learning courses is for people don't have time no distance learning is for people have time credit distance the do authority thing don't have class place I your things after thing the teach university e.
- work distance home programme workplace place the is not space to make things .
- research master degree engineering electrical trade CVS in research master degree Education technologie cad Education technologie not education master degree ,AIU not outcom engineering electrical.
+Framework saqa engineering is not Education technologie Education technic pedagogie career AIU Education Microsoft one note
-esucation technology circulum educator framework educator week modules years subject technology fundamental power education phenomenology AIU no allowed Master stability static education degree no stability static engineering creation linear stability in education trade.
- technical matric and education technology trade ncv matric educator.
- technologie manufacture research not Engineering matric engineering trade CVS .
- lecture facilitator trainer moderator assessor career education technology after di master engineering thesis degree Honore must complete master degree educator technic form thesis TVET and
- the master trade technologie and master master education technology are Cree humanity orientation cycle technologie creation humanity didn't overview concept humain key humanity
- technologies engineering humain vocational technical phase master humanity and component.
-is degree Batchelor is degree honorable master translate Sens possible appoint n engineering and Education in labour Education relation labour in security defense posted for understanding university undergraduate work sars sarb level master Eaton Scheineder master principle engineering engineering 12 years staff master ,12 years the appoint seniore training power city the appointment .
12 years experience job duty if the train senior advance technology you pass if not must work**

orientation TVET or master ,2 years .
- 12 years stables office work engineering power trade sign report draw design ups building is no stable is there building everyday,only one building the trade lay is not master office road is notaster office road public work stability
- have 1000 building new installation ,100 building city japon China ,100 entrepreneurship author chine in Congo e,3 years after years wiring engineering ,1000building USA rebuild computer wiring ,1000 architecture.
----
12 years experience cadet minim junior senior semmester experiemental duty training college and job trade drilling foreman experiemental after ejunior engineering staff engineering engineering engineering staff engineering job cpd engineering categories engineering cadet ,grade ,12 N1 junior level ecoxustrure Microsoft training cadet function duty grade,A,B,C,d job in your trade e ,N1,N2,N3,N4,N5,N6, subject module experiemental duty editing type career transmission generation power do it trainer do saps duty office doing cpd ,doing type career doing transmission generation power do it trainer do saps operation power do in your thesis advance field diploma do it seniore and principal engineering director duty core ,b

On Tue, 10 Dec 2024, 16:48 tshingombe fiston, wrote:
16.hydraulic components:fluid driver actuator or cylinder .
*** Vibratory mechanism , A system that produces oscillator or vibration, oft used in applications like material.**
2 . deriving equation for a pneumatic hydraulic system the dynamic described using Newton second low and the principles of fluid mechanics
_1 force balance the net force acting on the system,express as \F-{\text { net }}=F-{\ text { pneumatic}}++ ,
F-{\text { hydraulic }}- , F {\ text { damping }}-F{\ text (inertial }}
2.* Pneumatic force .the force generated by a pneumatic actuator.

To derive the relationship force ,motion. ,power ,energy .
\[F= m\ CDOT a \ ] where.
.\(F\) = force (N) , | \ (m \)= mass (kg) |\(a)= acceleration ,(m/s.s)
*** Work done by a force : work (\(w\) is defined as the force applied to an object time distance (\(d\) over which the force**
Is applied in the direction force .
|[W= F \ CDOT d \ CDOT \ cos(\thita \]
.\(w\)= work ,joule
.\(f\)= force ,N
.(d\)= Distance,m
.\(\ theta \ ) = angle between .

,3 energy :
Kinetic energy ,(k.E) is the energy of an object du it's motion .
\[K.E= \frac {I}{2}.m.V^2\..
Where .
\(V\)= velocity (m/s)..to analizing the concept of magnetic electromagnet and electrodynamics,system in relation silence ,or damping and solenoids

- understanding the concept.
- solenoid ,a coil of wire generate a magnetic field an electrical current pass through it.
*** Magnetic moment ,A measure of the strength and director of a magnetic source**
*** Electromagnetic induction , a measure of the strength and direction of a magnetic source .**
*** Electromagnetic.iduction .the process by changing magnetic.field induce and electromotive force ,EMF ,in a conductor .**

* Electocinectic ; refer to the motion of charged | particle a fluid under the inference of an electric field magnetic moment of solenoid..- the magnetic of solenoid.
-the magnetic moment (\(m\)) of a solenoid, \[m=n\cdot \cdot A \]
Where . \+Cn\)= number of turns per unit length ,turns / m
\(\)= current throughout the solenoid ,A
\(A\) = cross - sectional area of the solenoid,mm. Electromagnetic induction
- according to Faraday los electromagnetic the induce

16.3.The term Quotient intellectual calculus is term in mathematics or intellectual ass.
- intellectual Quotient ,(iQ) ,the ,IQ is a measure of a personal intellectual abilities in relation to standardise test that assess various cognitive skill .
- IQ \[\ text {IA}=\ left (\ frac { text { mental age }}{\ text { chronological age }}\ rigth) \ time ,100\]

- mental age : the age level at which a person perform intellectual.
- chronological age : the actual >
2.Quotient in calculus.
If you have two function \ (f(x) and \(g(x)\).the quotient \ [A(X)=\ frac {f(x){g(x)}
3.calxukating the derivative of a quotient , [fract = d}{dx{\ left (\ fract { f (x) }{g(x)\ rigth)

- to analyse psychometric variance ,variance in electrical psychometric field of study concerned with theory of psychopedagogie measurements knowledge ability attides and personality traits in this psychometric test analysed staatiscally ..
2. Calculating variance is statistics measure that represent the degree of spread in set of value in the of electrical measurements.for variance : the variance (\(} sigma ^2\)) of a set of values \ (x_1,x_2\ isots ,x_n\) is calculated using formula
€[| sigma ^2= \ frac { 1}{n}\ sum _ {I=1}^{x_i_\ mu)^2\]
- \(\ sigma ^ 2\)= variance
.(X_1\) = Each inductive . observations
- formulation

In electrical engineering under is crucial for analyse data especially.
1) variance : measure how a set of value differ from the mean of set it quantite the spread of the data paint .
-for a set of \(n\) observt it quantt the spread of the data .
Point formula for variance.
For a set of \(n\) observations \(x_1,x_2,\ isots ,x_ n \)
|[| Sigma ^2=\ frac {1}{n} sum_
{I=1}^{n}(x_1-\mu)^2
Where.\(\ sigma ^ 2))= variance.
.\(n\)= number of observations .
.\(xi)= each individual observation .
.\(| mu\)= mean of thicd ..
---\[|my= \ frac {1}{n} sum _ { I= 1}{n}, x _1\]

2.covariance measure the degree to which two the degree to which two random variables change together indicate the direction of the linear relationship between the variable : { foetus set of observations \ (x=(x_1,x-2\dot ,x_ n)\) and \ + y= (y-@,y-2,\ idots ,y-n

3. Calcul the electrical installation requirements for a building term .
- understanding power and energy .
* Power ,O * measure in kilowatt (kW) it represent the rate at which electrical energy is consumt products .
* Energy ,(E) : measured in kilowatt hour ,kWh it represents.

.\[E= O\ times \ ]
.\(E\)= energy in kWh
.\(P\)= power in kW
.\(t\) = time in hours .
- 2 calculating total power demand to calculate the total power for a building.

- list of electrical load ligthning ,10 fixtures a ,15 watt each ,HVAC : 3 kW , appliances ,2 kW other equipment ,1 kW ..
2 calculate total power demand ligthning \(10\ text{ fixtures \ times

On Tue, 10 Dec 2024, 15:40 tshingombe fiston, wrote:
14*. Mass balance equation : the general mass balance equation can expressed as :
\[\ text { input }-\ text { output }+\ { generation } \ text consumption}=\ text { accumulation \ ]
- for a steady state process (where accumulation is zero the equation simplified to \[\ text { input}\ text { output}\tezt Generation}-\ text { consumption}=\]
4.example calculation consider a simple chemical reactions input : A= 100kg / h , B =50 kg /h ,output ,,C= 120kg /h ..
.product..

14.1 to calculate the derivative and integral related an electromechanical systems we typically analysis the system behaviour using differential equations that describes the dynamic of the system structure approach to derive the master equation and performance the necessary.
14.1. master derivatives : electrical derivatives for a simple electrical circuit with an induction ,\(L\) and a resistor \(R\/ the voltage across the inductance can be by : \ [V-L { L\ frac {Di}{St}\]
Where \(v-L)= voltage accross the inductor.
.\(|\){ current through the inductor .
- b mechanical derivatives:for a mechanical system the relationship between torque \ (\tau\) and angular velocity \(\ omega \) can be described by .
\ you = |/ frac { d\ omega }{ St}\]
- where .\ (\ tau\) = torque.
\ (\\)= moment of inertia
\(\ omega \ ) = angular velocity master

14.2 definition: isostatic system a system that hasjus enough support to maintain equilibrium without any redundantly it has exactly as many constraints as necessary

- hyperstatic for equilibrium leading to redundancy in constraint.
* Stability : refers to the ability of a system to return to its original state after disturbance.
* Stability analysis: for stability analysis ,we typically use method.
Eingenvalue analysis for a system represented by a matrix the eingenvalue can indicate stability ,if all aigenvalue have negative real part the involved finding a lyapunov ,(function \(V(x)\) , such that \(V(X)>0\) and ,,\(\dot{(V)(X)<O\) for stability.
14.4 transformation to linear system to transform a hyperstatic system into a linear system , we can use the following step , modelling a motion ..

14.6 creating a programme for a artificial intelligence ,AI , system that focuses on operational metering in electric system involves several steps , including defining the object design the architecture implementation . Algor designed the architecture implementation algoris below .
- 1 define objective
- purpose : the AI system should monitoring analyse and Optimizer electric metering operations.
14.7.
Key features:.
- real time data collection from electric meter .

- data analysis for consumption patterns.
- anomaly detection for identifying irregularity.
- predictive maintenance for meter reporting and visualisation of data.

14.8. system architecture:.data source electric meter and sensor ,Day ,SQL no sQL) to store historical data .
*** Processing layer , implement data processing and analysis using AI algorithm.**
*** User interface development a dashboard for user to visualisation data and insights.**
*** Data collection / use API ,,direct connection to gather data from electric meters,example shifter for data collection ,(python)**
*** Python,import request,def ,collect meter dentK meter data storage.**
- r esponse request get ,(f" http:// API electricity meter comparable ,/ { meter _ I'd "} return response .jsob ( )

14.9.Creating on expert system for network involved several steps . < Including defining the objective designed the architecture. Implementating the algorithm below is a structure approach to developing.

-* define objective :
Purpose .the expert system shouand ld assist in network management troubleshooting and optimisation.
*** Key features: network monitoring and performance analysis troubleshooting and diagnostic capabilities.**
- recommendations for network configuration.
User friendly interface for networking administratir.
2. System architecture , knowledge base a repository of network knowledge including rules ,fact and heir interference engine the core Logica knowledge base derive,user interface

- implementation step : knowledge base developm.protocols configuration common issues and solutions
-plain text .
If network _speed
Acceptable _ level then
Recommended _ check _ hardware.
- inference Engine implement the inference Engine to process user queries and apply the rules from from the knowledge base.
Ex code snippet , python.
Python
Class expert system
Def _ initi_ self
Self . knowledge base
- to analyse a pneumatic hydraulic vibratory system equation governing the system and performance integrals
1. Understanding the system ,A pneumatic - hydraulic
Vibrator system typically consist of
*** Pneumatic components : air driven actuator or cylinder.**
*** Hydraulic components : fluid driven actuator or cycle**

On Tue, 10 Dec 2024, 07:27 tshingombe fiston, wrote:
 # apparent power calculay the apprent power can be calculated using the following formula .

\[S=\sqt{P^2+Q^2}\]
.value .real power (\(P\))=500w,
.reactive power(\Q\))=300VAR
- calculate apparent power (s)\[,S=\sqt{P^2t
- to calculate the characteristics of AC and DC machine we typically look at paramt such a peed torque and electromotive force ,(EMF) calculate these ,
Characteristics for both type machines .
-where .
|(a-O=| frac {1}{T} int- 0^ f(t)\dt)
.\+a-n = frac {2}{T} int_O^Y f (t) \ cos\ left (\ frac { 2\ Pi .n t}{ t } rigth \, DT have simple square wave function.

- to calculate the transformation and conservation of signal in the context of electrical signal we.
1 .signal transformation Fourier transform.
- the Fourier transform is used to convert a time domain signal into it frequency domain represent formula : transform \ F (\ omega)\(of a continuous signal \ (f(t) , e^{- j \ omega }\ dt\]
- where . \(f+ \ omega)\) = Fourier transform of the signal.
.\(f(t)\)= time - domain signal .
.\ (\ omega \) = angular frequency in Radia per second ..\ (j\) = imaginary unit .
BB* Laplace transform
- the la place trans is another transformation used to analyse linear time _ invariant system formula for the la place transform \ F(S)\) of function \ (f(t)\) is \[f(s)=\ into -{O} infty } , f(t) ,, e{-st } St \]
- \(f(s)\)= Laplace transform of the signal .
- \ f(ft c)l)= time domain signal.

- to calculate and understand synchroun and synchronous system, particularly in context of linearization .
1. Synchronous systems.
- in a coordinated, governed a common clock signal , in electrical synchronise system are used in digital circuit and communication system.
- example : lineare system the state space representatation .
\[| dot { x}(t}=Ax(t)+By(t)\]\[y(t)=(x(t)+du(t)\]
Where :
-|(x(t)\)= state vector
.|(u(t)\)= Input vector
.\ (y(t) \)= output vector .
.(A\)= System .
.\(B\)= input matrix .
.|(C\) = Output matrix .
.\(D \ ) = feed forward.

2.asynchronous system as asynchronous system operate without a global click signal operate independently and may not be synchronised this common in certain types of digital circuit and communication system .
- example equation for an asynchronous ,for an asynchronous linear systems the state space representatation .
|[| Dot { x} (t}= Ax (t) + B(t) \ ]\[y(t).
&
To calculate the integral of an amplified signal , detection of a signal and the probability of a radon signal aleatoire.
Integral of plidie signal
If you have a signal \ f(t)\) that is amplified by a constant factor \(A\) the amplifier signal can ,be represented as \(af (f(t)\). The integral of this amplified signal over a time interval \[a,b])\[\ int_ a^baf(t)\, St = A\ into _ a^ b ,f (t) \ St

**Exampt say \ f(t) = t ^ 2\) and \(A= 2\) we want to calculate the integral from \ (0\) to \ (1 }):
\[\ into _0^/2t^\, St = 2\ in_ 0^ t^2\,dt\]
Calculating the integral
\[into 0^1t 2^2\,St=\ left[\ fract { t^3}{3}\ rigth]-0^1=\
Frac {1^3}{3}_\ frac { 0^ 3}{3}=\ frac { 1}{3}\}
Thus \[\ into _ 0^ 1 ,2 T ^ 2\,dr..**

**To calculate or design a program for artificial intelligence ,AL within an operational framework we can outline the key component and steps involved .
Program
- 1 define the operational framework : an operational framework for an all program typically includes the following components.
* Objective : clearly defined the purpose of the AI program classification predict optimisation
* Data source : identify the data source requirements for training and testing the AI model database ,APU real time data ,
* Algorithm : choose the appropriate AI algorithm based on the problem type ,supervised learning , unsupervised learning reinforced
1. Data collection and preprot
Data collection gather data from identified source this could involve wab departing using APIs or accessing database.
* Data cleaning : remove duplicate handle missing value and correct inconsistent in the data.
* Feature ent : select and transfy relevant feat that will be used in the modej .
3* model development.
* Select model choose the AI model based on the problem type for .
- for classification decision tree random ,forest ,support vector ,machine ,neural networks.
-* for regression linear regression polynomial regression neural networks .
- training train model using the data set .
- to calculate a physical chemical plant balance we typically use the principles of mass and energy balance this,involves accounting for all input out son, accumulation of material and energy systems.structure approach to performing a mass balance physical chemical process .
2 define system : identify the boundaries of the system your are analizing this could be reactor distillation column any other unit operation in a chemical plan .
3. Identify input and output : list all the input and output system ,input can include raw material solve energy source while output / and was**

**On Mon, 09 Dec 2024, 18:27 tshingombe fiston, wrote:
1. DC machines : speed (n) the speed of DC motor can be calculated using formula \[N=\ frac { V-1\ CDOT -R}{ CDOT \ Phi }\]
- where \ (N\ ) = speed in Rpm (revolution perminute .
- \ (N\)= supply voltage (v) ,
-\(i\)= armature current (A)
\ (R\)= armature resistance ,(ohm .
\(K\)= a constant that depends on .
.\(\ Phi\) = flux per pole ,(WB)
b ,torque ,(T) ,the torque procedure by DC .
T= k \ CDOT } phi \ CDOT \] ,
Where .
.\(T\)= torque ,(N.m)
\(k\)= A constant that depends .
\(phi\)= flux per pole WB .
\(|\)= armature current .**

**To calculate de gradient of a function and derive the integral of a Senegal,
1. Calculating the gradient of a function**

The gradient of a function \(f(x,y)\) is a vector that contains all of its partial derivatives for a functionalite of two variables the gradient is given.
\[\ nabka , f=\ ,left(\fract { \ partial f) {\ partial x } , \ frac \ partial f }\ partial y \ right.
f(x,y) = x^2+x^2\]
.step ,1 calculate the partial derivatives.
.\(\ frac {\ partial f}{ \ partial x }=2x \)
.\(\ frac {\ partial f }{\ partial y}= 2y , I)
. Step 2: write the gradient \[\nabla ,f =2x,2y)\]
2. Deriving the integral of a signal ,we typically,use the fundamental theorem of calculus ,if we have a continuous function \(f(t,,)\ the integral from \(a)to \(b\)
Is given by : \[\ into$a^ b f(f) \,dt\]
-___
To calculate derivation ,both partial total double ,triple, relate ,to signal detection.
1. Partial derivatives : partial derivatives are used dealing with functionalite of multiple variable , for a functionalite.
\(f(x,y)\) the partial derivatives with respect to \(x\)
Is denoted as \(\ frac{\ partial f }{ partial x}\) and with ,
Respect to \ (y\) as \(\ frac {\ partial f }{\ partial ,y }\)
Examp : \[funct \[f(x,y)=x^2y+3xy^3\]
* Calculate partial derivatives .\(\ frac \ partial f){ partial ,x }= 2xy+3y^2\)
.\(\ fract \ partial ,f }{ partial y }{= x^2+6x6xy\)
* Total derivatives : the total derivatives account how a functionalite change with respect to all it variable.for a functionalite.
\(f(x,y)\), the total derivatives \ (DF \) is given by : \[DF = \ frac { \ partial f }{\ partial x }sx +\ frac{ \ partial }{ \ partial y } St \ ]
Using the previous : \ [DF = 2xy+ 2y^2)

- to calculate the Laplace and Fourier series Fourier a random vibrational signal ,signal aleatoire vibratoire in the context break down into a few steps.
1. Fourier series : the Fourier series and cosine function for periodic function \ (ft)\) with period (t) the Fourier series is.
f(t)= a 0+\ sum ${n= 1}{\ infty}
\ left + a_ n \ cos \ left \ left + } frac { 2\ pi not }{T}
\ right) + b _ n \ sunleft (\ track { 2\2 Pu ,not { T}\, \ rigth \ rigth)\]
- pratical exercise related to electrical engineering
Exercises calculate the total resistance in a circuit problem statement have ,3 resistance.
Resistor ,R1= 100 ohm resistor ,R2= 20 ohm , resistor ,R3= 309 ohm
Formula for total resistance.
.\[R-{\ text t{ total }}= R1+R2+R3\]
. Substituting the values \[R - { \ text { total }}= 10\.\ text , { ohms }+ 20\, text{ ohms}+30\, text {ohms}\], calculating ,\[R - { \ text { total /}} = 60\\ text { ohm }\]

- to calculate the supply trade theoty impedance and resonance in a electrical circuit ,we typically deal with RLC ,(resistant inductor , overview ; of impedance and resonance.
1. Impedance ,Z in a RLC circuit the total impedance is combination of resistance ,(R) inductive reactance ,x l and capacitive reactance ,(x- c) the formula for impedance in a series RLC ,circuit is
[Z =\ sqrt{R^ 2+(x-L-X_C)^2}\]
Where _\(x_L= 2\ pi f L \ Pi .f.c\} (capacitive reactance ,
-\(X_C)=\ frac ,{1}{2\ pi.f.c}\}(capacitive reactance.
-|(fl) is the frequency in Hertz (z)
-\(L/) is the inductance in Hertz (Hz) ,
| (L /) Is the inductance in Henry(h)
-\(cl) is the capacitance in farad (f)
2* resonance occurred in an RLC circuit when the inductive reactance equals the capacitive

reactance (\ (X_L= X_ C) at resonance the impedance is purely resistive and the formulation for resonance frequency form resonance frequency (\ (f - I)) is \
To the calculate the fundamental system electric power factor we need to understand relationship between real power reactive power and apparent power in electrical how to define.
1 .real power ,P owner reactive power and apparent power in electrical how to to definition real power the actual power consumed by the load measure in watt ,w
2) reactive power ,q the power the oscillator between the source and the load measured in volt amperage reactive ,varv
3 ,apparent power ,s the total power in the circuit , measure volt ampere ,Va is the combination a real and reactive power ,
- power factor calculation:
The power factor ,of is defined as the ratio of real power to apparent power , \ [\ text power factor ,of)}=\ frac ,{P}{s}\]
Where : .\(P\)= real power (w)
.\(S\\)= apparent power ,(VA)
apparent power calculation

On Mon, 09 Dec 2024, 17:06 tshingombe fiston, wrote:
-14. measure in true.
*1 types of measure errors measure : systematic these are considering repeatabt errors that occurred measurements system they.
*Random error unpredictable and can vary from one measure.
- gross errors : the are large errors that occure to human .
* Calibration of instruments ,calibrat is the process of adjusting instrument to ensure its measure are accurate step for calibration.
1. Select a standard: use a reference standard.
2. Measure with the instrument take measures using the instrument.
3. Compare measurements , compare the instruments .
4. Calculate errors the errors can \[text { text error }= text { measured values \ text { true value } \]
5. Adjust the instrument if system error are found adjust .
- to perform conversion between binary hexadecimal.
Conversion between number systeme.
* To convert a binary number to decimal ,use the formuler , \ [\ text { Decima} = sun -{ I= 0}^{n}b-1\colot 2^i/] ,where \(b_1\) is the binary digital (00r1) and \(n\) is the position of the difit from the rigth starting at 0 convert ,\ (1011_21) to decimal \ [= 1\ colot ,2^) to decimal \[= 1\ colot ,2^3+0\ colotv,2^2+1\ CDOT,
2^1+1\ colot ,2^O= 8+0+2+1=11_{10}\]
-decimal to binary : to convert a decimal number to binary divide the number by ,2 and record the remainder , repeat until the Quotient record the remainder ,repeat until the Quotient is ,On
Exp : convert \ (1@-{10}\) to binary .
\ [11\div ,2=5\ qual\ text { remainder ,r= \\ ,5 \ div ,2=2\
-to calculate the size of a memory accumulator in a binary system.
1) understanding binary representation:
In a binary system ,data is represented using bits ,binary digital where bit can either 00r,1 the number determine the range ,of value that can store .
2. Memory size calculation : the size a memory accumulator based number of bit it the total number of unique represented by an \(n\) bit binary number .
\ [text { number of values } = 2^n\]
Where . \(n\)= numbers of bits.
* Example calculation: determine the size of the accumulation.
2) calculate the number of value ,\[\ text { number of values }= 2^ 8 = 256\]
This mean the accumulator can hold values from \ (01) to (255\)(decimal ,### memory size in bytes
* memory size is of expressed in byte since ,1 byte = 8 bit ,size of the accumulator in bytes is \ [text , { sizer

To calculate thevenin , equivalent of a network ,short circuit current and voltage value ,
1. Thevenin theorem.
*** Overview: thevenin theorem state that any linear Electrical net with voltage source and resistance can be replaced by an equivalent circuit consisting of single voltage source (1(V-{the}\)) in series with with a single resistor (\ (R-{ the}\)).**
2. Step to find the in equivalent.
a identify the portion of the circuit select the portion the circuit for which
b calculate thevening voltage (\(V - (the }\))
1.open - circuit voltage, calculate the voltage across the terminal where the load was connected this is the thevenin voltage (\(V-*{the}\))
-2 method : you voltage division nodal analysing
- calculate thevenin resistance (\(R-{the}\))
- deactivated all independent source : replace independent field.
- to calculate amplification in circuits involving diodes transmission diode transistor ,and triacs understand each a analyse characteristics.
 # diode amplification diode are typically not used for amplification in the Sens performance signal modulation rectification signal signal modulation rectification diode current calculation.
\[I-D= L.S \ left (e^{\ frac { V-D}{ NV - T}} -1 rigth .
. \(1-D\)= diode current (A).
.\(I-S\)= reverse saturation current.
. \(V-D\)= voltage across the diode ,V
.\(n\) = ideality factor (typical between ,1 and ,2
. \ (V-T\)= thermal voltage (\approx 26\ MV) at room .
2. Transistor application transistor can use common collector thermostat common ,is common emitter amplifier .
1. Voltage gain (\ (A-C\))\[A-V=

To analyse and calculate parameter in a control system we typically focus on aspect such systems stability response.
""""""
2. Basic concepts in controle systeme.
*** Open - loop control system : systeme that does not use feedback to determine if it's output has achieved the desired goal .**
*** Closed loop control system systeme that uses feedback to compare the actual output to the desired output.**
,2 transfer function
The transfer function.
The transfer function \(H(S)\) of a control system relates the output (V(s)\<) to the input *\ (x,(s)\) in the Laplace domain : \ [H(s)= \ frac { y(s)}{x(s)}
3. Stability analysis , to determine the stability of a control system we can use the characteristics equation derivative the transfer function the characteristics equation is obtained by setting the denominator of the transfer function to zero
- for a transfer functions , ,\[H(S)=\ frac {k}{S^2+3S+2}\]
The characteristics equation is .
\[S^2+3s+2=0\] to find the root we can use ,
S=\ frac {-b\ pm \ start ,{ b^ 2-4ac}}{2a}

On Mon, 09 Dec 2024, 08:02 tshingombe fiston, wrote:
13. * Winding on rewinding transformers and machines both DC and AC involves understanding the specification of the windings the types of machine ,and the desired ,
13.1.* Understanding the types of machines .* DC machines these include DC motor and generator which typically have -armature windings and field windings.
- AC machine : these include ,AC motor ,like induction synchronous motor and transformser .

-2 key parameters for windings.
a winfmding specifications;
1. Number of turns (N) the number of turns in the winding affect the voltage and current characteristics.
2. Wire gauge : the thickness of the wire affect the resistance and current carrying capacity.
3.* Winding configuration Serie parallels combination depending machine type calculating wing parameters.
*** For DC machine .**
1.calculate the number turns the number of turns can be calculated based on the desired magnetic field strength for DC motor,back EMF(\(E\))\[E=\frac{frac{N\cdot\Phi\cdot\cdotZ}
-

- to calculate aspect related to telephonic cellular telecommunication supply .
- understanding cellular telecommunication supplies cellular telecommunication supply involves the infrastructure and resource required to provide mobile communication .
*** Base station : equipment that connect mobile devices to the network's.**
*** Backhaul the connection between base station and the core networks.**
*** Core network the central par of the telecommunication network that manage data and voice traffic ,##2* key calculation.**
a* coverage area calculation the coverage ,area of cellular tower can estimate using the following ,\[A=\PIr^2\]
-\(A\)=coverage area (in square kilometres)
-\(r)=radius of coverage (in kilometres)
Example : if a tower has a coverage radius of ,5 km
\[A=\pi(5^2)\ approx ,,78.54\ text { km } ^2\].
b. Capacity calculation: the capacity of cellular network calculated based on the number of channels available ,traffic per channel the Erlang ,B formula is commonly ,\[c=\frac{(A^B).}.{B.} \ big / sum ..

*** To calculate the component of a cellular telephone system and derive relevant integral, .**
1. Components of a cellular telephone system ,
A: cellular telephone system typicay consist of the components.
*** Mobile station (ms) * the users device ,smart phone tower that communication with mobile stations.**
*** Mobile switching centre ,MSC , manages the communication base station and the core networks .**
*** Core network,handles data routing billing and other ,**
,2 calculating key metrics.
a.coverage ,Area calculation the coverage area of a base station be estimated using .
\[A=\pir^2\]
-where .\(A\)= coverage area in square kilometres .
.\(r\)= radius of coverage in kilometres ,ex : if a base station has a coverage radius of ,3 km .
\ [A=\ pi (3^2)\ approx ,28.27\,\text ,{km} ^2\]
.b capacity calculation.

To calculate the spatial transmission characteristics of a system particularly in telecommunication.
##/ understanding spatial: transmission, spatial transmission refer to how signal private ,space transmission refer to how signal propagation space ,factor distance obstacle ,and the environment.
*** Free space path loss (fspl) the loss of signal strength ast travel through free space.**
*** Multiple path propagation the phenomenon where sign effect : the change in frequency of wave in relation to an observe moving relative to source of the wave .**

2 calculating free space path loss (fspl) the free space path loss can be calculated using .
\ [text \ text { fspl } = 20\ log - { ,10}{d} 20 log { 10}(f)+32,44\] where : \ d = distance between the transmit and receiver , kilometres ,\(; f\)= frequency of the signal ,in megahertz ,example calculation ,if the distance ,\ d (d) is ,10 Kim and the frequency, (f\) is ,900MHz ,\[\ text ,{ fspl}

To calculate the properties of material used and conductor insulator and magnetic material in electrical and stereo ,system ,we can analyse their characteristic.
1. Conductor : are material resistance common conductor..
- resistivity calculation the resistivity,
(\(\ rho}) of conductor is a measure of now strongly it resist the flow of electric current the resistance ,
(\(R)) Of conductor can ,
}[R=\ rho \ frac { L}{A}\]
.\(R\)= resistance,(ohms) .
.(\rho)= Resistivity (ohm metre)
.(Li)= Length of the conductors meter
./ (A\)= Cross - sectional area ,saaremeter .
Ex . Calculation for copper wire with a length of ,2 meter and a cross ,section area of \+|\ mm^2\)
(Which is / (i\ times ,10 ^{ - 6} , m ^\)) And using the resistivity of copper (\+\ rho \ approx ., 1,67\ times . 10^ ,=-8},| omega.| Cost m.})) ,
R= 1.68 } times , 10^
To calculate the size of a winding for stepper motor .
1. Understanding stepper motors.
- a stepper motor is a types of DC motor that decides a full rotation into a number of equal step winding configuration and size are crucial for the motor .
*** Number of phase : most stepper motor are either ,2 phase ,5 phase..**
*** Number of steps per revolution ,common value are ,200 steps ,(1.8 degree per step or ,400 steps (0.9 degree per sleep .**
*** Windt configuration the arrangement winding unipolar wire gauge : the thickness of wire used for the winding effects resistance ,**
3. Calculating the size of the winding : determine the number of turns s , the number of turns in each winding ,calculated based motor specifications : for example ,\[N= \ frac { v} { L csoti }\ ]

On Sun, 08 Dec 2024, 17:32 tshingombe fiston, wrote:
12. *Transmission system learning about the design and operation of transmission system for radio and television inclut antennas , modulation technique and signal processing.
***12.1 Broadcasting technologie exploring the technologie used in broadcasting such as satellite communication digital broadcasting and stream.**
*** 12.2.Sound engineering : understanding the eof accoustict ,inclust sound wave w sound design and audio technology ,includ application in audio engineering noise control and sound system design.**
*** 12.3.Optic : learning about the behaviour of lights including .**
- broadcasting e , exploring the technologie used in broadcasting such as satellite communication digital broadcasting and the princit of accoustict including sound wave progration sound design and audio technology application in audio e noise control sound system design.
*** 12.4. Optics : learning about the behaviour of ligth including reflection refraction and diffraction.you ' ll study optical system lenses mirrors and fibre optic which are essential in various technologies including images systeme .**
*** Application , exploring how sound and can be integrated into Engineering solutions such imagine (ultrasound) optical communication system and sensor technology.**
***12.5. Electrical machines: understanding the principles and operations of electric machines including motor generator,and transformers ,you ll learn about their design control and applicat in various industries**
*** 12.6.Electrotech : this field focusy on the study of electrical system and their components including circuit design power distribution and electrical safety you'll gain knot about the standard and regulations governing electrical installation , electrotechnology this encompasses the applicat of electrical and electronics technologie in various field including automation control system and**

reneu energy systems explore technology are used to improve efficiency and performance in Engineering applicat.
*** 12.7. Radio wave propagation understand how radio wave travel through different engineering environment including factor that affect their range and quality such terroir , wether and frequency .**
***12.8; communication system : learning about the design and operation of radio communication system including ,AM ,,FM and digital radio broadcasting.**
*** Antenna desii: exploring the principles of antenna theory and design including different types of antenna and their application in various communication system .**
***12.9 signal processing: gainit knowledge in technique for processing and analizing radio signals to improve communication quality and efft.**
12.10 . Radiotecht play a cruct role in telecommunication broadcasting and many modern technologies if you many modern technologies.
- random signals understanding the characteristics and analysis of signal that have a random or stochastic nature ,this include studying noise statistically .
- *12.11vibratory signal : learning about signal related to vibration which can be crucial in field like mechanical engineering structural health monitoring and accoustict you 'll study how technology interpretation vibrator signals ,to access the conditt of structural of machinery .
***Application , exploring how both random and vibration my signal are used in various applications such as in telecommunication audio engineering.**
***12.12. probability theory: understanding the principles of probability including random variables probabit distribution and the low of large number thesis knowledge is essential for modelling uncertainty engineering systems.**
***12.13 statistical methods: learning about about variously statistict technique for data analysis including hypothesis testing regression eand statistics inference ,these methods are to .**
***12.14 . building electrical system and materials are essential component in electrical engineering.**
- built electrical system: understanding the design installation of electrical system understanding the design installation of electrical system in building includy ligthi power distribution and emergency system you ll about codes and standards that government electrical installation.
12.15.electrical material study the various.material used in electriy system conductor , insulator , semiconductor you explore their property how they affect the performance and safety of electrical system.
***12.16. sustainable practices : learning about energy efficiency design and reneu energy integration in built design and renewable energy integration in building systems include solar power and smart grid .**

-*12.17. construction electrical refer to the electrical system and installation that are integral to building.
***12.18. electrical design : understanding how to design electrical system for buit including power distribution ligthning and communication system how to design electrical system for building inclidiy power learn about load calculations circuit design and systeme.**
*** Installation practt learning about the best practices installation electrical syst in construction.lroject wiring panel installation and safety protot.**
*** Buit code and standards formiliaring with the local and natit code that government electrical installation in construction.**

On Sun, 08 Dec 2024, 15:41 tshingombe fiston, wrote:
Topics ,are
- 11. project management : gaining skill in managing electrical construction project including buildings budgeting schedule.
*** 11.1 .Entrepreneurs , management design management their .**
***11.2. business planning understanding how to create comphrensive business plan that outline goal strategies and financial projections is crucial for securing funding guiding busiy operation .**
*** 11.3. projection management learning about tools and techniques for managu project including schedules resource allocation and risk Mt helps entrepreneurs budget.**

- financial management : gaming knowledge in managing in managing finance include budgeting accounting analyse this is business decist and ensure profitability.
* Marketing and sales strategies : exploring effects marketing techniques and sale strau to attract and ret custt includes digital e.
*Technology integration , understand how to leverage technology and software solutions to streamline operational improves efficiency.
Low commercial regulation refere to minimal government intervention and oversight in commercial activities
- impact on businesens operation , understanding how low regulation can create armored flexible environment for businesses allowit for easier entry into
-11.3.-market and dynamics : analysing how regulation effect competition innovation consumer chaise can lead to increased entrepreneurship but also raise can lead ,to increased entrepreneurship but also raise .

- 11.4 .legal framework learnings about the legal aspects of commercial regulation including contract trade practice and consumer protection low even low , regulation environment business must navigation countries approach commercial regulation and the .
-11.5.implication for internatt : trade and investment , mining geotechnical engineering is a specialized ,field that focuses on the behaviour of earth material in mining operations.
* Geotechnical analysis: understanding the properties of soil and rock behvot under various conditions this is crucial design .
* Slope stability learn about the analyse and design of slopes in open ,pi mining and undersgroun , excavation to parent landslides.
* Ground support system , exploring the design .
* Global perspective , exploring how different countries approach commercial regut and the impliy for international trade and investment .
* Mining geotechnical engineering is a specialized field that focuses on the behaviour of earth material in geotechnical analyse , understanding the properties of soil and rock material including their strength stability and behaviour under various conditions ,this is crucial for design safe and efficient mining .
* Slope stability : learning about the analysis and ..design of slopes in open mining and underound excavation to prevent land slide and ensure the safety of workers and equipment.
* Ground support systems:.exxplot the design and implementation explore ground support system ,such as Rick bolts shot Crete and mesh to stabilize excavation collapse.
* Environmental considerations: understanding the environmental impact of mining activities and how to mitigate risk associated with with ground .
* Site investigation risks associated with ground .
Site investigation gaining skills in conducting site investigation to assess geological and inform mining design and planing.
- 11.5electrical stability understanding stability of electrical system including voltage stability frequency stability and transient stability this involves analizing how systeme response to distribution and ensuring they can return to stable operating conditions.
-transformer operation including how they step down level in power systems ,you 'll study design effict and perft characteristics.
* 11.6.1Transformer conservation , exploring method method for conserving energy in transformer operational including to, management tools management maintenance,practice and the use of energy efficiency .
*11.6.6 Transformer crucial for reducing losses and improving overall system efficiency.
* Conditt : monitoring gaming knowledge in tech monitory the health and performance of transformer including temperature monitoring insulation testing dusgnostt .
* Spatial Caltrain concept in various fields.
-11.7. spatial control system : understanding how to design implementation control system that montage the position and movement of object in a three dimensional space crucial in application .
* 11.6.Robotic and automation learning about the principles of controlling robotics system includ

kinematics dynamic , exploring how to integration sensor GPS lidar camera into .

On Sun, 08 Dec 2024, 14:42 tshingombe fiston, wrote:
-9.1 simulation and modelling gaining in simulating control system to analyse their performance and Optimizer their design .
- 9.1.1.satellite communication understanding the principles of satellite communication system including hour satellite transmitted and ret signals the different types of satellite geostationary low earth arbitrary application in broadcasting.
-9.1.2. fiver optic technology learning about fiver optic communication uses light to transmit data over long distances with minimal loss you studies installation.
- 9.1.3.integration of techniques exploring how satellite and fibre optic technologie , exploring how satellite and fibre optic technologies can be integrated to provide comprehensive communication solution such as using satellites for connection in remote areas where .
-9.1.4. network design and Optimizer communication network utlize both satellite and fibre optic technologie data transmission and connectivity .
- 9.1.5 energy electro energies ,likely ref various forms of electrical energy their application in .
- *to electro energy systems . understanding the generator energy includes studying power plants , renewable energy source , like solar ,winds hydro and the electrical .
9.1.6 .energy conversion. Learning about the process involves in converting different form of energy mechanical thermal chemical. Into electrical energy and this includes studying devices like generator motor .
- *energy efficiency .exploring method .
- energy efficiency exploring method to improve the efficiency of electrical energy use in various applications including industrial processes.
*** Smart grids gaining knowledge in the device management of smart grid technologies that enhances reliability and efficiency.**
*** Network engineering how to design efficiency and cable network including local area network LANs wide area network ,Wan's and cloud network,you learn about network topologies protocols ,and architecture.**
-*network security , learning about the principles of security studying firewalls ,instrusion detection system and encryption.
-*network management gaming skill in managing and monitoring network performance , including troubleshooting issue optimizig traffic flow and ensi reliability.
- emerging technologies exploring new trend in network software ,
- definitely networks (SDN) , network functionalite virtualisation (NFV) and the internet of things .
***Certification and standards: familiart yourself with industry standard ,/ certification such .**
- certificate network associate ,(CCNA) or competition network which can enhance your career.
-;electrical trade theory is an essential aspect of electrical of engineering and vocational training that focuses on the principles , practice and standard related to electrical work .
*** Fundamentals principal understanding the basic concept of electricity including ohm s low Kirchhoff's low,and other principle of circuits ,voltage current and resistance.**
*** Electrical code and standards learning about the regulation and stars that govern electrical installation and safety practices such national electrical code ,(NEC) local building.**
*** Installation practices gaining knowledge in the proper technique for installing electrical systems including wiring circuit breakers outlets and ligthning fixture while ensuring compliance safety standards.**
*** Troubleshooting and maintenance developing skills in diagnosing and repairing electrical issues including understanding common problem and implementation effective solution.**
*** Safety practices , emphasising the importance of safety in electrical work practice and understand electrical hazard, instruments measurements and controle in electrical engineering is a critical reaction focused on teachiques and tools used to measure and control electrical ..**
- Measurements techniques technique understand various for measuring electrical quantities such as voltage current resistance power and energy this include multimeter oscilloscope and power

analyser .
- control systems . learning about the about the principles of control system, including loops control algorithm and systeme stability to design and implementation control systems to regulate electrical process.
*** Sensors and transducer ,: exploring the type of sensor and transducer used to convert physical quantities.**
- temperature , pressure and flow into electrical signal for application.
- data acquisition system,gaming knowledge in the designated and implementation of data acquisition system that collected and analizing data from various sensor and instruments for monitoring and control purposes.
- automation and process control understanding how, measurements controle systeme are Applied in industrial automation including programmable logic controller ,PLC and supervisory control and data acquisition , SCADA..

*** 10. Banknote processing machines specialized device used in the banking and financial store to handle ,.. in the banking sector to handle sort.**

***Currency authentication understanding the technologie used in bank note processing machines to verify the authenticity of currency note ,this includes features like ultraviolet ,UK ligth detection magnetic character recognise and infrared IR , scanning.**

10.1 telecommunication systems understanding the principles of communication system signal processing and networking design .
- Power systeme learning about the generation transmission and distribution of electrical power as well as renewable energy sources.
10.2 .neural ,
*** Medical imaging using neural neural networks for image analysis in MRI ,CT scans and x- rays it improves diagnostic.**
*** Predictive analytics : developing model to predict patient outcomes or disease progression base on medical data**

***10.3 Wearable technology : integrating neural networks into devices that monitor health metrics in real time .**
*** 10.4. Mathematics : advanced topics such as linear algebra calculus differential equations and status which are essential for modelling and solving Engineering problem .**
***10.4.1. Physics : concept related to electromagnetic circuit theory and signal processing which or .circuciak understanding electrical systeand their applications.**
-10.4.2andragogy focused on the methods and principle used in adult educay emphasising the unique need of adult contest of your master program understanding andragie help you design effive learning experience.
-10.4.3 educational philosophy involves the study of the fundamental naturel and purpose of education it can guide your approach to learning and teaching help your approach to learning and teaching help you to develop a personal philosophy that design with your goal in Engineering Education.
***10.4.5Professional theory this include the ethical legal and social implications of engineering practices as well as the responsibility of the Engineering in society it prepares you to make informed decisions in your professional career.**
***10.5.4..Trade theory ,this focuses on the technical skill and knt requirements in specific engineering trade it often include hand , on training and pratical application of Engineering concepts.**
***10.5.4. Industrial electronics this invot the study of electronics systeme used in industrial applications including automation control system ,and robotics sensor , actuator and the integration of electronic systems in manufacturing process.**
*** 10.5.5Digital system : focused on digital circuit design microcontroller and digital signal**

processing digital technology is applied variance field .
***Advanced circuit theory : building on basic circuit principle to explore complex circuit network theories and analysis technique.**
***10.5.6 Electro magnetic including Maxwell equation wave propagation and field theory which are crucial for many.**
- 10.5.5control system : delving into advanced control theory include feedback system stability analizing and control design techniques.
- 10.5.6electromechanical mechatronics is an exciting interdisciplinary field that comine mechanical engineerin.
electronics computer.
-19.5.6 mechatronics systems systems understanding how mechanical systems integrate with electronics control and software to create intelligent system the include robotics ,Utomation and smart device .
- 10.3.controle systeme learning about the principles of controle electromechanical system loops ,sensor .
***10.4 Design and analysis ,gaining skill in design and e mechatronics focusing on their functionality efficiency.**
- computer architecture.
,- 10.5. parallels computing understanding how multiple processors or core work together to perform task more efficiently including concept like parallel algorithms concurrency and synchronisation.
-10.6. computer architecture learnings about the design and organisations of compulator system including CPU memory hierarchy input ./ Output system.
- 10.7. performance evaluation , analysing the performance parallel system including metric .
-10.8 . policy development understands how to create implementation and maintenance policies that govern organisation practice especially in Engineering projects .
-*10.8. compliance and risk management learning how to ensure that police align with legal and regulatory requirements. As well how to assess and mitigate risk, association with engineering practice.
*** Information system exploring how technology management policies documents management system workflow automy data analytics to tract compliance..**
*** 10.9 .Security systemes , understanding the design and implementation of system that protect information and asset including cybersecurity measure encryption and secure communication protocol**
*** 10.10.Safety engineering: learning about principle of designing system that ensure the safety of user and the environment ,including risk assessment hazard analysis ,and safety management systems..**
***10.11. defense system : exploring technologie and strategic used in national defense , including surveillance systems threat detection and response mechatronics.**
TV radio .
-* 10.1 media frequency : understanding the electromagnetic spectrum and how different frequencies are used for various forms of communication including any and FM radio television broad casting

On Sun, 08 Dec 2024, 08:00 tshingombe fiston, wrote:
5.motoring electrical vehicles.
***5.1 overview: this area focus on the design development and operations of electric vehicle (EVS) and their components . including electric motor , batteries and chargers system.**
- *5.2. key topic : you might explore electric motor design battery technology power electronics and vehicle dynamics understanding the integration of renewable energy source and renewable energy.
-*5.3 . substation: overview , substation are design protection system design protection system, controle and maintenance practice , understanding the role substy in smart grid technology and

renewable energy integration,is .

*** 5.4 . Nanotechnology:**
***5.5 overview: involved manipulating matter at the nanoscale billion of meter to create material and devices with unique properties field has application across various industries electronics medicine and in the context of electrical engineering study nanoscale component as transistor sensor and energy storage devices nanotechnology enhance ,perfy .**

5.6 cellular components :
5.6.1 overview this refer to the study of t structure and function of cells structures on function cell which are the basic application in biotechnology and cellular signak memoire brand dynamic and role of protein and nuclei acids.
5.6.1 azure and machine learning Microsoft Azure is a cloud computing platform that provides a wide range of services including machine learning data storage and development to buit deployment and application machine development to build deployment and application machine python use task such as analysing medical .
***5.6.2. Assess moderator :**
*** Overview is responsible for overseeing and ensuring the quality and fairness of assessment in Education settings this role often involves evaluation effectiveness .**
*** Key topics: focus on asssessment evaluation effect.**
***Key topic : focus an assessment evaluation criteria and best practices for ensuring reliability and validity in testing.**
*** 5.6.3. Education,didactic :**
*** overview : didactic in the science of teaching and learning it involved understanding how to effectively learning experience.**
*** Key topics: study instruction design curriculum development and teaching strategies ,styles .**
5.6.4. psychopedagogy ,
***Overview: this field combines psychology to understand how psychological principle can be applied to Education practice ..**
.5.6.5.: role is some who guide and supporter ,a group or individual in achieving their goals often in Educational or professional setting this role involves creating learning collaboration.
***Key skill : effective commy ,active listening conflict resolution and the ability to faster engagement skill for a facilitation .**

***5.6.6 Personality care in montesory:**
-overview: education setting per .
- care focuses on nurturing the individual child's development including their emotional.
*** Key principles: Montessori educy emphasising respect for the child fastening independent and creating a supportive environment that encourages exploration a supportive environment that exploration and self directed learning personality care involves understanding each childs..**

***6.1 Marine Engineering overview marine engineering focused construction and maintenance of ships boats and other marine vessel combined with electrical engineering it involves the electrical system that power and control .- key topics : in this field marine propulsion system electrical power generation and t ontrok system for navigation and automation engineering.**
- key topics in this field marine propulsion system electrical power navigation and automation engineering.
- key topics: in this field you might study field you study marine propulsion electrical power generation and distribution control system for navigation and automation and safety systems marine systems marine electrical engineering ensure that the electrical system on vessel are efficient reliable and compliant with maritime.

- 6.2. labour machinery low.

*** Overview : this area focuses on the legal regulation standards governing the use of machinery in the workplace safety ,labour rights and operationel standards.**
*** Key topics: you might study occupation safety.regulation machine stardard and compliance understanding low is crucial for ensuring , environment.and protecting workers .**
***6.3. Bargaining:**
***Overview bargaining typically refers to the negotiations process between employer workings conditions wage and other.**
- key topics : you might explore collective bargaining agreement negotiation strategies and labour relations under.dynamic of bargaining is essential for mastering positive workplace.

*** How to make a self assessment ,exam creating a self , assessment exam can help you evaluation your understanding of biophysics engineering concept.**
1. Identify key topics, list the main topics concept you want to assess for biophysics engineering area like biomechanics medical.
2.create questions : development variety of questions types multiple choice provide several optt for each question true false simple statement that the responsibility must .
- identify as true or false.
*** ______________**
-6.4 marking topics for electrical assignment exam Portofilio.
- preparing your Portofolio for an electronical assignment exam .
1.select relevant topics : choose topics that Lign with the course objectives and your interest the could area like circuit design power systems control power or renewable energy.
- organisation : your work structure your Portofolio logically you might include section for.
*** Introduction :**
***Of the topics cover.**
*** Projection: detailed description of project you completed including objective methods.**
*** Assignment: including key assignment that demonstrates your understanding of the material..**

7.1 .Sorting and counting learning about the mechanism that allowed these machines to sort and count bank note efficiently involves understanding the sensor and algorithm used to detect different denomination and conditions of note ,new worn or damages.
-quality controle exploring how bank note processing machine ensure that only acceptable note are circulate, removal of counterfeit or damaged not are circulate removal of counterfeit on damaged notes from.
- integration with banking system gaining knowledge integrate with bank systeme for invatory management cash flow analysis and reporting.
- maintenance and trout , understanding maintenance requirements and common issue that can arise .
- with banknoy processing machine ,.

________________'''
- 7.2 chemical engineering engineering and science are distinct yet interconnected field with engineering that focuses chemical engineering.
***Overview: this field involve the design optimization and operations of process that convert row material into valuable product chemical fuels pharmaceutical and dad,**
***You might study thermodynamics , reaction engineering ,process design and separation process chemical engineering also focus on safety sustainability environment impose**

7.3. physics engineering:.- overview physic t applies principle of physic development new technology and solve engineering problem.often overlap with field like electrical mechanical and materials science.
- key topic: explore topics electromagnetic thermodynamics and quantum physics engineering work project involving.
.- science engineering.

*** Overview : is Broder term that can encompasses various engineering disct that apply science principle to solve pratical problt include interdisciplinary approach .**
*** Key topics on focus study area science biomedical engineering often work on research and project requirements a strong foundation.**

-7.4.biophysical Engineering is an interdit field that combines principles of physic biolt and enito understand and development technologie related to .
- biomaterials: understanding the properties and applications of material used in medical device implants and tissue engineering this include studying how these materials interact with biological.
*** Biomechanics: learning about the mechanical principle govern biological systems including the movement of organism and the force acting on biology tissue this knowledge is crucy for designing .**
- medical imaging exploring technologie used.
- biological structure and functt MRI ,CT and ultrasound ,physic being imagi technique and theirs application in medicine.
- bioinformatics: gaining knowledge in the the computational tools and techniques used to analyse biological data including genetic sequence and protein structure this is essential for understanding complex brigicak system ,systeme biology understanding how biological systems functionalite as interaction between genes ,protein metabolism pathways this knowledge can infot the design of targeted therapies and biotechnological application

---

-7.9. biophysical engineering and total productive maintenance ,tpm are important concepts in the field .
1.biophyscal engineering:
- overview: this interdisciplinary field combines principles of biological physics and engineering to develop technologie and process that improve healthcare and bioloy design of medical device biomaterials and bioprocesses .
-key topics : you might study area a biomechanics bioinformatics medical imat and tissues engineering physical Engineering work on project that involve the application of physic principle to biologist system ,such as developing prosthesis ..

- 9.10. total productive maintenance (tpm)
*** Overview: tpm is a maintenance philosophy aimed at maximizing the productivity of equipment by minimising downtime and ensui that machines operate at peak efficiy .it involve all employee in the maintenance process ,from a operator to manai**
***- key topics : explore concepts such as automouse maintenance ,planned maintenance and continuous improvements ,tpm ,focuses on proactive maintenance strategies including regular inspection preventive maintenance..**

*** 10. Relavtiviy ,**
*** Overview : relativity primarily associated with Albert Einstein include the theories of soeciat relativity and general relativity thesis theories revolutionised our understanding of space time ,and gravity .**
- key topics : in engit you might study the implicai of relativity in field like astrophysics GPS technt and high speed particle physic , understanding relativity is crucial for application involve high velocities or strong gravitational.
- hydraulic and pneumatic system uses liquid,while pneumatic uses gases both system are widt used in industrial applications machinery and automation.
- key topics: you might study fluid mechanics system design control system and the component.

On Sat, 07 Dec 2024, 21:03 tshingombe fiston, wrote:

3. Workshop lab: aspect of trade e in electrical engineering trade theory often involves the pratical application of theoretical concept in a workshop settings.
***1. Fundamentals of electrical theory , understanding ohm low ,Kirchhoff's low and other foundation principle that government electrical circuit.**
***2. Hands - on circuit assembly , student typically engagement assembling and testing various electrical circuits applying theoretical knowledge to practical scenario.**
3.troubleshooting technique , workshop often include exercise diagnosis and fixing uses in electrical system , which is crucial.
4.safety practice : emphasising safety protocol when working with electrical components and systems is vital part of any workshop.
5.usr of tools and equipment familiarisation with tools such as multimeter , oscilloscope and soldet equipment ,which are essential for electrical engineering task .
6.project based learning ,student may work on specify project that requires them to apply traditional theory concepts such a designing a simple electrical device or system .
7. Collaboration and teamwork , encourage
-information on workshop lab that cover trademarks panel wiring electrical switch one way and two ,way relay motor .
- panel wiring : basic of panel wiring learning how to wire electrical panel including understanding circuit diagrams and layout planning.
*** One-way switches hands- on practice with one - way switch which control a ligth or device from a single location.**
*** Two way switch work with two way switch that allows control of a ligth or Devuce from two different hallways or large room.**
*** Relay motor ,AC and DC motor understanding the difference between AC (DC ,) motor their application and characteristics relay operational , learning how relay work ,their in controlling motor and other device , students may practice wiring relay to control ,AC and DC motors,**
- practical application : hands - on project that involves wiring circuit with one way and two way switch integrating.

6.Lab : workshop,
1. Industrial electronics.
*** Overview of industrial systems:**
Understanding the component and systeme used industrial electronics including sensor.actuor and controle systeme.
. installation practices : learning best practices for installation electronics system in industrial settings , including wiring ,mounting and configuration.
2.*Computer installation: hardware setup hands on experience with installing computer hardware compagny including matherboard ,power supplies and peripheral.
*** Software installation: understand the process of installing operating system and necessary software for computer system.**
*** Safety rules : electrical safety emphasising the importance of safety protocol when when working electrical system.**
- including proper use of personal protective , equipment ,(PPE) and safety handling of tools .
*** Compliance with standard , learning about industrial.**
*** ,4 . fault finding technique , troubleshooting teach systeme Pproach diagnosing the use of flowchart.fault in electronics system including the use of flowchart and checklist ,use of diagnostic tools familiarisation with tools such as multimeter oscilloscope and tester to identify and analyse fault.**
- 1* high voltage safety , safety protocol.emphasising the importance of safety when working with high voltage system including the used of personal protective equipment PPE and understanding hazard.
*** Emergency procedures , training response procedure incase of electrical accidents or equipment failure.**
*** Power generation : type of power generation exploration various methods of power generation**

includ thermal hydroelectric ,wind and solar power .
*** Generation , equipment hand on experience with generator, transformer and other equipment used in power generation.**
3. Transmission : line design understanding the design operation of high voltage transmission line . including factor affecting.
- efficiency and reliability.
Substation operation learning,about the role of substation in the transmission system , including switching,protection .
4. Engineering trademarks, standards and certification , familiarisation with industry standard and trademarks related to high voltage equipment and systems , IEEE,IEX,ANSI,,
-Quality assurance: understanding the importance to ensure safety reliability and performance in power systems.

Manufacturing process of electrical components.
1. Design and prototyping.
- concept development engineer design the electrical components consideration functionality material and specifications.
- prototyping: is created to test the design and functionality before mass production.
2.*material selection choosing material select material: selecting appropriate material based on electrical thermal ,and mechanical properties common material including metal , plastered and ceramic ..
3* fabrication , machining : cutting drilling and shsppings material to create the component parts ,
*** Molding : for plastic components , injection molding often used to create complex shapes.**
4* assembly : components assembly : parts are assembled together ,which may include soldering welding or using adhesive components like resistor capacitor and microcontroller into the assembly.

On Sat, 07 Dec 2024, 19:44 tshingombe fiston, wrote:
1.thesis research in electrical engineering.
-*1.1.overview : conducting thesis research in electrical engineering typically involves identifying a specific problem or area of interest within the field conducting.
Experiemental,or simulation and analysing risk.
***1.2. Key topic: possible research area could include power systems control system.**

Telecommunication,or embedded systems your thesis contribute new knowledge or soluyto existing challenge in the electrical and electronics.
***1.3.trade theory in electrical electronics.**
***1.4: overview this involves understanding principle and electronics relate to electrical and electronics system installation, maintenance and safety.**
***1.4. topics : you might study electrical code circuit design ,and troubleshooting technique this knowledge is essential for ensuring safety safe and efficient electrical installation in variouse..**
***1.5 . advantage and disadvantage trade theory in electrical engineering.**
*** Innovation and development trade theory encourage competition which can lead to innovation and development of new technologies In countries to specialise in the production of certain electrical good loading to more efficient use of resources.**
- economic growth: engagement in international trade can boaf economic growth by expanding market for electrical.
- knowledge transfer : trade can facilitate the the exchange of knowledge transfer trade can facilitate the exchange of knowledge and technology between countries, enhance the overall capabilities..
- disadvantages:
1.6. dependency: countries may become overly dependent on imported electrical good which can be risky if supply chain are disrupted .
-1.7.. trade theory in electrical engineering.

*** Overview trade theory in electrical engineering often refer to the principles and practices related to the electrical trade : including , installation maintenance and , safety standards.**
-irregularity in material design THR's could refer to issue related to the consistency and ,quality of material used in electrical .
- application , understanding how to identify and address irregularity in material is crucial for ensuring safety and performance in electrical
1.8. backlog issues:
- *overview: in the context of engineering and project management backlog issue refer to delay or outstanding task that need to beadress occured in variouse stage of a project from design to implementation ,
-1.9. key considerations: addressing backlog issues, often involves analizing task ,and efficiently this is crucial for maintenance project to timeline and ensuring successful.
-key topics : electrical : calculation understanding how to perform calculation related electrical . System, including loaf calculation voltage. Drop and circuit design.
- Power supply system : learning about different types of power supply system , including ,AC and DC system transformers and and distribution.

-2. Interested in Educational technology can impact the outcomes of manufacturing topics in electrical engineering Engineering.
2.1 simulation software: tools like MATLAB and Simulink allow students to model and simulation electrical ,system ,students to model and simulate electrical , system helping them understand complex concepts without the need for physical prototype.
*** Online learning platform these platforms provide access to a wealth of resources including video lecture interactive quizzes and forum for discussion making easier for student to learn at their .**
*** Collaborative tools : technologie like cloud based . collaboration platform allow students to work together projects and instructor enhythr system taillor Education content to the individual need of students helping them grasp difficult concept ,in manufacturing and electrical engineering more.**
- *industry parterneship: collaboration with industry can provide students with real world project and case studies bridging the gap between theoretical knowledge and practical application in manufacture.
- lab workshop electrical engineering .
1.circuit design and analysis ,student design and analyse various electrical circuits using bread board, simulation software.
2.microconyroller programming: workshop include.progtamming microct ,(like Arduino or raspberry control device and sensor.
3.Powe system: experiemental,may involve studying power generation transmission and distribution including renewable energy source .
4.conyrol system ,student learning about feedback system.ans controle theory through pratical application and simulation.
5. Electronics prototyping : workshop may focus on building prototype of electronic devices,allowing students to apply their knowledge in real - word scenario.
6. Testing and measurement student learn to used various testing equipment such as oscilloscope and multimeter,to measure electrical parameter.
- telecommunication, workshop may cover topics like signal processing and communication system in

Experimental

1. Overview v: school money make is budget academic voting wordsr asssessment order book copyrt order salary pay sleeping salary base shift teacher lecture learn auditing years pay bonus lesson from ,100 rand per day day shifting ,2500 rand salary wage bonus annual ,× 12 month over e extra class teacher in lecture assessor moderator granted primary,6 teacher high School ,12 teacher subject n2 to ,n 6 six lecture if double shift teacher and lecture rand house home air time water,× 100 rand ,× 30,3000 ×9000+900 water water = 18000 rand class per month grade ,10×800, rand 800×6 = 400000×12= 48000000 rand ,pay government returned tax , Amandment .
- bank account school have ,200000 rand account school ,2000000 estimate budget and money granted award now compliance ,5 5000 rand by school desk chaire desk panel wiring buyer ,poy Ccma labour court award ,bank school teacher e to Ccma t seta casebook , money school pay is not for boss is school ,pay money school pay is not for boss is school pay money school make arrested irregularity .
- school fee policy arrested report didn't pay search exhibition years proof ecourse subjt no record books ,till point policy ,
- pay granted settlement arrange demage interested court pay complain pay case order pay review transct payment irregularity payment judge made aware money assesment order book judge pay the pay granted skill development levy bargaining.
Uif

We already told you about the rise and evolution of toll-free servive numbers and the first mobile phone. In this article you'll learn more about the further evolution of digital telephony and mobile telephones; from large analogue machines to the intelligent smartphone as we all know it.
3 generations: 1G, 2G and 3G
When the first mobile phone appeared in the '80s there were different mobile networks throughout the world. Frontrunner was Japan, where the first commercial network in the world was already active in 1979. At first it was just Tokyo that was connected, but within 5 years the rest of the country had joined the capital. The USA followed soon after, and Europe responded by setting up Global System for Mobile Communications, or GSM.
All these different networks had one thing in common: they could only be used for calls through the analogue network, but soon the 1G network made a name for itself. In 1991 the first digital mobile network was implemented in Finland by Radiolinja, what later became part of Elisa Oyj. This network was labelled 2G; the second generation of wireless telephone technology. 2G had a few advantages. Information could be transferred encrypted over the digital network, it was more efficient and 2G made it possible to send data as well – like SMS and MMS. And everything was secured digitally.
Though the mobile network evolved rapidly with the introduction of 3G in 2001 and 4G in 2014, the 2g network is still being used today in some parts of the world, but more and more countries choose to gradually shut down their 2G networks.
Digital telephony and smartphones
Already in the '70s Theodor G. Paraskevakos developed the concept of a phone that could connect intelligence and data processing with a virtual display. This concept could be realized once 2G made it possible to send data over the network.
There are several answers to the question who introduced the first smartphone to the market. In the middle of the '90s there were 2 devices which could be used to call, email and fax, and a digital calendar was built in as well. These were the early versions of the smartphone:
• In 1994 BellSouth introduced the Simon Personal Communicator.
• In 1996 introduced Nokia their Nokia Communicator.
The Nokia Communicator (photo on the right) had all the functionalities of a personal computer, like email, web browsing, word processing, and combined these with fax and telephony. The device costed over 800 dollar and was everything but handy and easy to use, which scared away consumers and kept them from buying one. The owner of Nokia at that time, Jorma Ollia, stated: "We had exactly the right view of what it was all about… We were about 5 years ahead."
The first smartphones were primarily suitable for business use, because of the high prices and

unpractical design. Its main use was that of personal digital assistent.
The name 'smartphone' was coined by Ericsson in 2000, in a campaign to increase its popularity amongst consumers as well. With the introduction of the 3G network (in 2001) and the arrival of lighter, more simple devices at the same time the smartphone finally became a hit. In 2007, the year that Apple launched the first iPhone, already 8 million people worldwide had used a smartphone to visit the internet, and in 2012 there were over 1 billion smartphone users worldwide. It's estimated that by 2017 the number of smartphone users will approach 2,5 billion. This makes us wonder what technological developments the near future has in store for us – something we'll discuss in the final part of this series.

Contents

Pendulum example

•

See also

•

References

•

Further reading

•

External links

State-space representation

- **Article**
- **Talk**
- **Read**
- **Edit**
- **View history**

Tools

•
•
•
•
•
•
•
•
•
•
•
•
•

Appearance
Text
•
Small
Standard
Large
Width
•
Standard
Wide
Color (beta)
•
Automatic
Light
Dark
From Wikipedia, the free encyclopedia
See also: quantum state space and configuration space (physics)
In control engineering and system identification, a state-space representation is a mathematical model of a physical system specified as a set of input, output, and variables related by first-order

differential equations or difference equations. Such variables, called state variables, evolve over time in a way that depends on the values they have at any given instant and on the externally imposed values of input variables. Output variables' values depend on the state variable values and may also depend on the input variable values.

The state space or phase space is the geometric space in which the axes are the state variables. The system state can be represented as a vector, the state vector.

If the dynamical system is linear, time-invariant, and finite-dimensional, then the differential and algebraic equations may be written in matrix form.[1][2] The state-space method is characterized by the algebraization of general system theory, which makes it possible to use Kronecker vector-matrix structures. The capacity of these structures can be efficiently applied to research systems with or without modulation.[3] The state-space representation (also known as the "time-domain approach") provides a convenient and compact way to model and analyze systems with multiple inputs and outputs. With inputs and outputs, we would otherwise have to write down Laplace transforms to encode all the information about a system. Unlike the frequency domain approach, the use of the state-space representation is not limited to systems with linear components and zero initial conditions.

The state-space model can be applied in subjects such as economics,[4] statistics,[5] computer science and electrical engineering,[6] and neuroscience.[7] In econometrics, for example, state-space models can be used to decompose a time series into trend and cycle, compose individual indicators into a composite index,[8] identify turning points of the business cycle, and estimate GDP using latent and unobserved time series.[9][10] Many applications rely on the Kalman Filter or a state observer to produce estimates of the current unknown state variables using their previous observations.[11][12]

State variables

The internal state variables are the smallest possible subset of system variables that can represent the entire state of the system at any given time.[13] The minimum number of state variables required to represent a given system, , is usually equal to the order of the system's defining differential equation, but not necessarily. If the system is represented in transfer function form, the minimum number of state variables is equal to the order of the transfer function's denominator after it has been reduced to a proper fraction. It is important to understand that converting a state-space realization to a transfer function form may lose some internal information about the system, and may provide a description of a system which is stable, when the state-space realization is unstable at certain points. In electric circuits, the number of state variables is often, though not always, the same as the number of energy storage elements in the circuit such as capacitors and inductors. The state variables defined must be linearly independent, i.e., no state variable can be written as a linear combination of the other state variables, or the system cannot be solved.

Linear systems

Block diagram representation of the linear state-space equations

The most general state-space representation of a linear system with inputs, outputs and state variables is written in the following form:[14]

where:

- **is called the "state vector", ;**
- **is called the "output vector", ;**
- **is called the "input (or control) vector", ;**
- **is the "state (or system) matrix", ,**
- **is the "input matrix", ,**
- **is the "output matrix", ,**
- **is the "feedthrough (or feedforward) matrix" (in cases where the system model does not have a direct feedthrough, is the zero matrix), ,**
- **.**

In this general formulation, all matrices are allowed to be time-variant (i.e. their elements can depend on time); however, in the common LTI case, matrices will be time invariant. The time variable can be continuous (e.g.) or discrete (e.g.). In the latter case, the time variable is usually used instead of . Hybrid systems allow for time domains that have both continuous and discrete

parts. Depending on the assumptions made, the state-space model representation can assume the following forms:
System type State-space model
Continuous time-invariant

Continuous time-variant

Explicit discrete time-invariant

Explicit discrete time-variant

Laplace domain of
continuous time-invariant

Z-domain of
discrete time-invariant

Example: continuous-time LTI case
Stability and natural response characteristics of a continuous-time LTI system (i.e., linear with matrices that are constant with respect to time) can be studied from the eigenvalues of the matrix . The stability of a time-invariant state-space model can be determined by looking at the system's transfer function in factored form. It will then look something like this:

The denominator of the transfer function is equal to the characteristic polynomial found by taking the determinant of , The roots of this polynomial (the eigenvalues) are the system transfer function's poles (i.e., the singularities where the transfer function's magnitude is unbounded). These poles can be used to analyze whether the system is asymptotically stable or marginally stable. An alternative approach to determining stability, which does not involve calculating eigenvalues, is to analyze the system's Lyapunov stability.
The zeros found in the numerator of can similarly be used to determine whether the system is minimum phase.
The system may still be input–output stable (see BIBO stable) even though it is not internally stable. This may be the case if unstable poles are canceled out by zeros (i.e., if those singularities in the transfer function are removable).
Controllability
Main article: Controllability
The state controllability condition implies that it is possible – by admissible inputs – to steer the states from any initial value to any final value within some finite time window. A continuous time-invariant linear state-space model is controllable if and only if where rank is the number of linearly independent rows in a matrix, and where n is the number of state variables.
Observability
Main article: Observability
Observability is a measure for how well internal states of a system can be inferred by knowledge of its external outputs. The observability and controllability of a system are mathematical duals (i.e., as controllability provides that an input is available that brings any initial state to any desired final state, observability provides that knowing an output trajectory provides enough information to predict the initial state of the system).
A continuous time-invariant linear state-space model is observable if and only if

Transfer function
The "transfer function" of a continuous time-invariant linear state-space model can be derived in the following way:
First, taking the Laplace transform of
yields Next, we simplify for , giving and thus
Substituting for in the output equation
giving
Assuming zero initial conditions and a single-input single-output (SISO) system, the transfer function is defined as the ratio of output and input . For a multiple-input multiple-output (MIMO) system, however, this ratio is not defined. Therefore, assuming zero initial conditions, the transfer function matrix is derived from
using the method of equating the coefficients which yields

Consequently, is a matrix with the dimension which contains transfer functions for each input output combination. Due to the simplicity of this matrix notation, the state-space representation is commonly used for multiple-input, multiple-output systems. The Rosenbrock system matrix provides a bridge between the state-space representation and its transfer function.
Canonical realizations
Main article: Realization (systems)
Any given transfer function which is strictly proper can easily be transferred into state-space by the following approach (this example is for a 4-dimensional, single-input, single-output system):
Given a transfer function, expand it to reveal all coefficients in both the numerator and denominator. This should result in the following form:
The coefficients can now be inserted directly into the state-space model by the following approach:

This state-space realization is called controllable canonical form because the resulting model is guaranteed to be controllable (i.e., because the control enters a chain of integrators, it has the ability to move every state).
The transfer function coefficients can also be used to construct another type of canonical form
This state-space realization is called observable canonical form because the resulting model is guaranteed to be observable (i.e., because the output exits from a chain of integrators, every state has an effect on the output).
Proper transfer functions
Transfer functions which are only proper (and not strictly proper) can also be realised quite easily. The trick here is to separate the transfer function into two parts: a strictly proper part and a constant.
The strictly proper transfer function can then be transformed into a canonical state-space realization using techniques shown above. The state-space realization of the constant is trivially . Together we then get a state-space realization with matrices A, B and C determined by the strictly proper part, and matrix D determined by the constant.
Here is an example to clear things up a bit: which yields the following controllable realization Notice how the output also depends directly on the input. This is due to the constant in the transfer function.
Feedback
Typical state-space model with feedback
A common method for feedback is to multiply the output by a matrix K and setting this as the input to the system: . Since the values of K are unrestricted the values can easily be negated for negative feedback. The presence of a negative sign (the common notation) is merely a notational one and its absence has no impact on the end results.

becomes

solving the output equation for and substituting in the state equation results in

The advantage of this is that the eigenvalues of A can be controlled by setting K appropriately through eigendecomposition of . This assumes that the closed-loop system is controllable or that the unstable eigenvalues of A can be made stable through appropriate choice of K.
Example
For a strictly proper system D equals zero. Another fairly common situation is when all states are outputs, i.e. y = x, which yields C = I, the identity matrix. This would then result in the simpler equations

This reduces the necessary eigendecomposition to just .
Feedback with setpoint (reference) input
Output feedback with set point
In addition to feedback, an input, , can be added such that .

becomes

solving the output equation for and substituting in the state equation results in

One fairly common simplification to this system is removing D, which reduces the equations to

Moving object example
A classical linear system is that of one-dimensional movement of an object (e.g., a cart). Newton's laws of motion for an object moving horizontally on a plane and attached to a wall with a spring:

where
- **is position; is velocity; is acceleration**
- **is an applied force**
- **is the viscous friction coefficient**
- **is the spring constant**
- **is the mass of the object**

The state equation would then become

where
- **represents the position of the object**
- **is the velocity of the object**
- **is the acceleration of the object**
- **the output is the position of the object**

The controllability test is then

which has full rank for all and . This means, that if initial state of the system is known (, ,), and if the and are constants, then there is a force that could move the cart into any other position in the system.
The observability test is then

which also has full rank. Therefore, this system is both controllable and observable.
Nonlinear systems
The more general form of a state-space model can be written as two functions.

The first is the state equation and the latter is the output equation. If the function is a linear combination of states and inputs then the equations can be written in matrix notation like above. The argument to the functions can be dropped if the system is unforced (i.e., it has no inputs).
Pendulum example
A classic nonlinear system is a simple unforced pendulum

where

• is the angle of the pendulum with respect to the direction of gravity
• is the mass of the pendulum (pendulum rod's mass is assumed to be zero)
• is the gravitational acceleration
• is coefficient of friction at the pivot point
• is the radius of the pendulum (to the center of gravity of the mass)
The state equations are then

where
• is the angle of the pendulum
• is the rotational velocity of the pendulum
• is the rotational acceleration of the pendulum
Instead, the state equation can be written in the general form

The equilibrium/stationary points of a system are when and so the equilibrium points of a pendulum are those that satisfy

for integers n.
See also
• Control engineering
• Control theory
• State observer
• Observability
• Controllability
• Discretization of state-space models
• Phase space for information about phase state (like state space) in physics and mathematics.
• State space for information about state space with discrete states in computer science.
• Kalman filter for a statistical application.
References
1.
• Katalin M. Hangos; R. Lakner & M. Gerzson (2001). Intelligent Control Systems: An Introduction with Examples. Springer. p. 254. ISBN 978-1-4020-0134-5.
• • Katalin M. Hangos; József Bokor & Gábor Szederkényi (2004). Analysis and Control of Nonlinear Process Systems. Springer. p. 25. ISBN 978-1-85233-600-4.
• • Vasilyev A.S.; Ushakov A.V. (2015). "Modeling of dynamic systems with modulation by means of Kronecker vector-matrix representation". Scientific and Technical Journal of Information Technologies, Mechanics and Optics. 15 (5): 839–848. doi:10.17586/2226-1494-2015-15-5-839-848.
• • Stock, J.H.; Watson, M.W. (2016), "Dynamic Factor Models, Factor-Augmented Vector Autoregressions, and Structural Vector Autoregressions in Macroeconomics", Handbook of Macroeconomics, vol. 2, Elsevier, pp. 415–525, doi:10.1016/bs.hesmac.2016.04.002, ISBN 978-0-444-59487-7
• • Durbin, James; Koopman, Siem Jan (2012). Time series analysis by state space methods. Oxford University Press. ISBN 978-0-19-964117-8. OCLC 794591362.
• • Roesser, R. (1975). "A discrete state-space model for linear image processing". IEEE Transactions on Automatic Control. 20 (1): 1–10. doi:10.1109/tac.1975.1100844. ISSN 0018-9286.
• • Smith, Anne C.; Brown, Emery N. (2003). "Estimating a State-Space Model from Point Process Observations". Neural Computation. 15 (5): 965–991. doi:10.1162/089976603765202622. ISSN 0899-7667. PMID 12803953. S2CID 10020032.
• • James H. Stock & Mark W. Watson, 1989. "New Indexes of Coincident and Leading Economic Indicators," NBER Chapters, in: NBER Macroeconomics Annual 1989, Volume 4, pages 351-409, National Bureau of Economic Research, Inc.
• • Bańbura, Marta; Modugno, Michele (2012-11-12). "Maximum Likelihood Estimation of Factor Models on Datasets with Arbitrary Pattern of Missing Data". Journal of Applied Econometrics. 29 (1): 133–160. doi:10.1002/jae.2306. hdl:10419/153623. ISSN 0883-7252. S2CID 14231301.
• • "State-Space Models with Markov Switching and Gibbs-Sampling", State-Space Models with

Regime Switching, The MIT Press, pp. 237–274, 2017, doi:10.7551/mitpress/6444.003.0013, ISBN 978-0-262-27711-2
• • Kalman, R. E. (1960-03-01). "A New Approach to Linear Filtering and Prediction Problems". Journal of Basic Engineering. 82 (1): 35–45. doi:10.1115/1.3662552. ISSN 0021-9223. S2CID 259115248.
• • Harvey, Andrew C. (1990). Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge: Cambridge University Press. doi:10.1017/CBO9781107049994
• • Nise, Norman S. (2010). Control Systems Engineering (6th ed.). John Wiley & Sons, Inc. ISBN 978-0-470-54756-4.
14. • Brogan, William L. (1974). Modern Control Theory (1st ed.). Quantum Publishers, Inc. p. 172.

Further reading

• Antsaklis, P. J.; Michel, A. N. (2007). A Linear Systems Primer. Birkhauser. ISBN 978-0-8176-4460-4.
• Chen, Chi-Tsong (1999). Linear System Theory and Design (3rd ed.). Oxford University Press. ISBN 0-19-511777-8.
• Khalil, Hassan K. (2001). Nonlinear Systems (3rd ed.). Prentice Hall. ISBN 0-13-067389-7.
• Hinrichsen, Diederich; Pritchard, Anthony J. (2005). Mathematical Systems Theory I, Modelling, State Space Analysis, Stability and Robustness. Springer. ISBN 978-3-540-44125-0.
• Sontag, Eduardo D. (1999). Mathematical Control Theory: Deterministic Finite Dimensional Systems (PDF) (2nd ed.). Springer. ISBN 0-387-98489-5. Retrieved June 28, 2012.
• Friedland, Bernard (2005). Control System Design: An Introduction to State-Space Methods. Dover. ISBN 0-486-44278-0.
• Zadeh, Lotfi A.; Desoer, Charles A. (1979). Linear System Theory. Krieger Pub Co. ISBN 978-0-88275-809-1.

On the applications of state-space models in econometrics

• Durbin, J.; Koopman, S. (2001). Time series analysis by state space methods. Oxford, UK: Oxford University Press. ISBN 978-0-19-852354-3.

External links

• Wolfram language functions for linear state-space models, affine state-space models, and nonlinear state-space models.

• v
• t
• e

Differentiable computing

General

• Differentiable programming
• Information geometry
• Statistical manifold
• Automatic differentiation
• Neuromorphic computing
• Pattern recognition
• Ricci calculus
• Computational learning theory
• Inductive bias

Hardware • IPU
• TPU
• VPU
• Memristor
• SpiNNaker

Software libraries • TensorFlow
• PyTorch

- Keras
- scikit-learn
- Theano
- JAX
- Flux.jl
- MindSpore

- Portals
o Computer programming
o Technology

Categories:
- Classical control theory
- Mathematical modeling
- Time domain analysis
- Time series models
- This page was last edited on 11 Nove

Topics 2
Advanced Telecommunications Computing Architecture

- Article
- Talk
- Read
- Edit
- View history

Tools
•
•
•
•
•
•
•
•
•
•
•
•
•
Appearance
Text
•
Small
Standard
Large
Width
•
Standard
Wide
Color (beta)
•
Automatic
Light
Dark

From Wikipedia, the free encyclopedia
This article may be too technical for most readers to understand. Please help improve it to make it understandable to non-experts, without removing the technical details. (August 2013) (Learn how and when to remove this message)

Advanced Telecommunications Computing Architecture[1] (ATCA or AdvancedTCA) is the largest specification effort in the history of the PCI Industrial Computer Manufacturers Group (PICMG), with more than 100 companies participating. Known as AdvancedTCA, the official specification designation PICMG 3.x (see below) was ratified by the PICMG organization in December 2002.[2] AdvancedTCA is targeted primarily to requirements for "carrier grade" communications equipment, but has recently expanded its reach into more ruggedized applications geared toward the military/aerospace industries as well.[3] This series of specifications incorporates the latest trends in high speed interconnect technologies, next-generation processors, and improved Reliability, Availability and Serviceability (RAS).
Mechanical specifications
12U 14-slot AdvancedTCA shelf
An AdvancedTCA board (blade) is 280 mm deep and 322 mm high. The boards have a metal front panel and a metal cover on the bottom of the printed circuit board to limit electromagnetic interference and to limit the spread of fire. The locking injector-ejector handle (lever) actuates a microswitch to let the Intelligent Platform Management Controller (IPMC) know that an operator wants to remove a board, or that the board has just been installed, thus activating the hot-swap procedure. AdvancedTCA boards support the use of PCI Mezzanine Card (PMC) or Advanced Mezzanine Card (AMC) expansion mezzanines.
The shelf supports RTMs (Rear Transition Modules). RTMs plug into the back of the shelf in slot locations that match the front boards. The RTM and the front board are interconnected through a Zone-3 connector. The Zone-3 connector is not defined by the AdvancedTCA specification.
Each shelf slot is 30.48 mm wide. This allows for 14-board chassis to be installed in a 19-inch rack-mountable system and 16 boards in an ETSI rack-mountable system. A typical 14-slot system is 12 or 13 rack units high. The large AdvancedTCA shelves are targeted to the telecommunication market so the airflow goes in the front of the shelf, across the boards from bottom to top, and out the rear of the shelf. Smaller shelves that are used in enterprise applications typically have horizontal air flow.
The small-medium AdvancedTCA shelves are targeted to the telecommunication market; for the lab research operation, some shelves have an open cover in order to make testing easier.
Backplane architecture
The AdvancedTCA backplane provides point-to-point connections between the boards and does not use a data bus. The backplane definition is divided into three sections; Zone-1, Zone-2, and Zone-3. The connectors in Zone-1 provide redundant −48 VDC power and Shelf Management signals to the boards. The connectors in Zone-2 provide the connections to the Base Interface and Fabric Interface. All Fabric connections use point-to-point 100 Ω differential signals. Zone-2 is called "Fabric Agnostic" which means that any Fabric that can use 100 Ω differential signals can be used with an AdvancedTCA backplane.[4]
The connectors in Zone-3 are user defined and are usually used to connect a front board to a Rear Transition Module. The Zone-3 area can also hold a special backplane to interconnect boards with signals that are not defined in the AdvancedTCA specification.
The AdvancedTCA Fabric specification uses Logical Slots to describe the interconnections. The Fabric Switch Boards go in Logical Slots 1 and 2. The chassis manufacturer is free to decide the relationship between Logical and Physical Slots in a chassis. The chassis Field Replaceable Units (FRU) data includes an Address Table that describes the relationship between the Logical and Physical slots.
The Shelf Managers communicate with each board and FRU in the chassis with IPMI (Intelligent Platform Management Interface) protocols running on redundant I^2C buses on the Zone-1 connectors.
The Base Interface is the primary Fabric on the Zone-2 connectors and allocates 4 differential pairs

per Base Channel. It is wired as a Dual-Star with redundant fabric hub slots at the core. It is commonly used for out of band management, firmware uploading, OS boot, etc.

The Fabric Interface on the backplane supports many different Fabrics and can be wired as a Dual-Star, Dual-Dual-Star, Mesh, Replicated-Mesh or other architectures. It allocates 8 differential pairs per Fabric Channel and each Channel can be divided into four 2-pair Ports. The Fabric Interface is typically used to move data between the boards and the outside network.

The Synchronization Clock Interface routes MLVDS (Multipoint Low-voltage differential signaling) clock signals over multiple 130 Ω buses. The clocks are typically used to synchronize telecom interfaces.

Update Channel Interface is a set of 10 differential signal pairs that interconnect two slots. Which slots are interconnected depends on the particular backplane design. These are signals commonly used to interconnect two hub boards, or redundant processor boards.

Fabrics

The Base Interface can only be 10BASE-T, 100BASE-TX, or 1000BASE-T Ethernet. Since all boards and hubs are required to support one of these interfaces there is always a network connection to the boards.

The Fabric is commonly SerDes Gigabit Ethernet, but can also be Fibre Channel, XAUI 10-Gigabit Ethernet, InfiniBand, PCI Express, or Serial RapidIO. Any Fabric that can use the point-to-point 100 Ω differential signals can be used with an AdvancedTCA backplane.

The PICMG 3.1 Ethernet/Fibre Channel specification has been revised to include IEEE 100GBASE-KR4 signaling to the existing IEEE 40GBASE-KR4, 10GBASE-KX4, 10GBASE-KR, and XAUI signaling.

Blades (boards)

AdvancedTCA blades can be Processors, Switches, AMC carriers, etc. A typical shelf will contain one or more switch blades and several processor blades.

When they are first inserted into the shelf the onboard IPMC is powered from the redundant −48 V on the backplane. The IPMC sends an IPMI event message to the Shelf Manager to let it know that it has been installed. The Shelf Manager reads information from the blade and determines if there is enough power available. If there is, the Shelf Manager sends a command to the IPMC to power-up the payload part of the blade. The Shelf Manager also determines what fabric ports are supported by the blade. It then looks at the fabric interconnect information for the backplane to find out what fabric ports are on the other end of the fabric connections. If the fabric ports on both ends of the backplane wires match then it sends an IPMI command to both blades to enable the matching ports. Once the blade is powered-up and connected to the fabrics the Shelf Manager listens for event messages from the sensors on the blade. If a temperature sensor reports that it is too warm then the Shelf Manager will increase the speed of the fans.

The FRU data in the board contains descriptive information like the manufacturer, model number, serial number, manufacturing date, revision, etc. This information can be read remotely to perform an inventory of the blades in a shelf.

Shelf Management

AdvancedTCA Shelf manager

The Shelf Manager monitors and controls the boards (blades) and FRU in the shelf. If any sensor reports a problem the Shelf Manager can take action or report the problem to a System Manager. This action could be something simple like making the fans go faster, or more drastic such as powering off a board. Each board and FRU contains inventory information (FRU Data) that can be retrieved by the Shelf Manager. The FRU data is used by the Shelf Manager to determine if there is enough power available for a board or FRU and if the Fabric ports that interconnect boards are compatible. The FRU data can also reveal the manufacturer, manufacturing date, model number, serial number, and asset tag.

Each blade, intelligent FRU, and Shelf Manager contains an Intelligent Platform Management Controller (IPMC). The Shelf Manager communicates with the boards and intelligent FRUs with IPMI protocols running on redundant I²C buses. IPMI protocols include packet checksums to ensure that data transmission is reliable. It is also possible to have non-intelligent FRUs managed by an intelligent FRUs. These are called Managed FRUs and have the same capabilities as an intelligent

FRU.
The interconnection between the Shelf Manager and the boards is a redundant pair of Intelligent Platform Management Buses (IPMBs). The IPMB architecture can be a pair of buses (Bused IPMB) or a pair of radial connections (Radial IPMB). Radial IPMB implementations usually include the capability to isolate individual IPMB connections to improve reliability in the event of an IPMC failure.
The Shelf Manager communicates with outside entities with RMCP (IPMI over TCP/IP), HTTP, SNMP over an Ethernet network. Some Shelf Managers support the Hardware Platform Interface, a technical specification defined by the Service Availability Forum.
New specification activity
Two new working groups have been started to adapt ATCA to the specific requirements of physics research.
• WG1: Physics xTCA I/O, Timing and Synchronization Working Group
WG1 will define rear I/O for AMC modules and a new component called the µRTM. Additions will be made to the µTCA Shelf specification to accommodate the µRTM and to the ATCA specification to accommodate AMC Rear I/O for an ATCA carrier RTM. Signal lines be identified for use as clocks, gates, and triggers that are commonly used in Physics data acquisition systems.
• WG2: Physics xTCA Software Architectures and Protocols Working Group
WG2 will define a common set of software architectures and supporting infrastructure to facilitate inter-operability and portability of both hardware and software modules among the various applications developed for the Physics xTCA platform and that will minimize the development effort and time required to construct experiments and systems using that platform.
A working group was formed to extend ATCA to non-telecom markets.
• PICMG 3.7 ATCA Extensions for Applications Outside the Telecom Central Office
The goals of this new working group are to define enhanced features to support double-wide boards; add enhancements to support 600W single-slot boards and 800W double-slot boards; add support for double-sided shelves with full sized boards plugged into both the front and rear of the shelf; and add support for 10Gbs signaling on the Base Interface.
PICMG specifications
• 3.0 is the "base" or "core" specification. The AdvancedTCA definition alone defines a Fabric agnostic chassis backplane that can be used with any of the Fabrics defined in the following specifications:
• 3.1 Ethernet (and Fibre Channel)
• 3.2 InfiniBand
• 3.3 StarFabric
• 3.4 PCI Express (and PCI Express Advanced Switching)
• 3.5 RapidIO
See also
e Wireless (disambiguation).
"Over the air broadcasting" redirects here. For the technology over the air television, see Terrestrial television.
A handheld on-board communication station of the maritime mobile service
Part of a series on
Antennas

Common types

Components

Systems

- **Antenna farm**
- **Amateur radio**
- **Cellular network**
- **Hotspot**
- **Municipal wireless network**
- **Radio**
- **Radio masts and towers**
- **Wi-Fi**
- **Wireless**

Safety and regulation

Radiation sources / regions

Characteristics

Techniques
- **v**
- **t**
- **e**

Wireless communication (or just wireless, when the context allows) is the transfer of information (telecommunication) between two or more points without the use of an electrical conductor, optical fiber or other continuous guided medium for the transfer. The most common wireless technologies use radio waves. With radio waves, intended distances can be short, such as a few meters for Bluetooth, or as far as millions of kilometers for deep-space radio communications. It encompasses various types of fixed, mobile, and portable applications, including two-way radios, cellular telephones, personal digital assistants (PDAs), and wireless networking. Other examples of applications of radio wireless technology include GPS units, garage door openers, wireless computer mouse, keyboards and headsets, headphones, radio receivers, satellite television, broadcast television and cordless telephones. Somewhat less common methods of achieving wireless communications involve other electromagnetic phenomena, such as light and magnetic or electric fields, or the use of sound.
The term wireless has been used twice in communications history, with slightly different meanings. It was initially used from about 1890 for the first radio transmitting and receiving technology, as in wireless telegraphy, until the new word radio replaced it around 1920. Radio sets in the UK and the English-speaking world that were not portable continued to be referred to as wireless sets into the 1960s.[1][2] The term wireless was revived in the 1980s and 1990s mainly to distinguish digital devices that communicate without wires, such as the examples listed in the previous paragraph, from those that require wires or cables. This became its primary usage in the 2000s, due to the advent of technologies such as mobile broadband, Wi-Fi, and Bluetooth.
Wireless operations permit services, such as mobile and interplanetary communications, that are impossible or impractical to implement with the use of wires. The term is commonly used in the telecommunications industry to refer to telecommunications systems (e.g. radio transmitters and receivers, remote controls, etc.) that use some form of energy (e.g. radio waves and acoustic energy) to transfer information without the use of wires.[3][4][5] Information is transferred in this manner over both short and long distances.
History
See also: History of telecommunication
Photophone
Main article: Photophone
Bell and Tainter's photophone, of 1880.
The first wireless telephone conversation occurred in 1880 when Alexander Graham Bell and Charles Sumner Tainter invented the photophone, a telephone that sent audio over a beam of light.

The photophone required sunlight to operate, and a clear line of sight between the transmitter and receiver, which greatly decreased the viability of the photophone in any practical use.[6] It would be several decades before the photophone's principles found their first practical applications in military communications and later in fiber-optic communications.
Electric wireless technology
Early wireless
Main article: Wireless telegraphy
A number of wireless electrical signaling schemes including sending electric currents through water and the ground using electrostatic and electromagnetic induction were investigated for telegraphy in the late 19th century before practical radio systems became available. These included a patented induction system by Thomas Edison allowing a telegraph on a running train to connect with telegraph wires running parallel to the tracks, a William Preece induction telegraph system for sending messages across bodies of water, and several operational and proposed telegraphy and voice earth conduction systems.
The Edison system was used by stranded trains during the Great Blizzard of 1888 and earth conductive systems found limited use between trenches during World War I but these systems were never successful economically.
Radio waves
Main article: History of radio
Marconi transmitting the first radio signal across the Atlantic.
In 1894, Guglielmo Marconi began developing a wireless telegraph system using radio waves, which had been known about since proof of their existence in 1888 by Heinrich Hertz, but discounted as a communication format since they seemed, at the time, to be a short-range phenomenon.[7] Marconi soon developed a system that was transmitting signals way beyond distances anyone could have predicted (due in part to the signals bouncing off the then unknown ionosphere). Marconi and Karl Ferdinand Braun were awarded the 1909 Nobel Prize for Physics for their contribution to this form of wireless telegraphy.
Millimetre wave communication was first investigated by Jagadish Chandra Bose during 1894–1896, when he reached an extremely high frequency of up to 60 GHz in his experiments.[8] He also introduced the use of semiconductor junctions to detect radio waves,[9] when he patented the radio crystal detector in 1901.[10][11]
Wireless revolution
Power MOSFETs, which are used in RF power amplifiers to boost radio frequency (RF) signals in long-distance wireless networks.
The wireless revolution began in the 1990s,[12][13][14] with the advent of digital wireless networks leading to a social revolution, and a paradigm shift from wired to wireless technology,[15] including the proliferation of commercial wireless technologies such as cell phones, mobile telephony, pagers, wireless computer networks,[12] cellular networks, the wireless Internet, and laptop and handheld computers with wireless connections.[16] The wireless revolution has been driven by advances in radio frequency (RF), microelectronics, and microwave engineering,[12] and the transition from analog to digital RF technology,[15][16] which enabled a substantial increase in voice traffic along with the delivery of digital data such as text messaging, images and streaming media.[15]
Modes
Wireless communications can be via:
Radio
Main article: Radio
Further information: Microwave transmission
Radio and microwave communication carry information by modulating properties of electromagnetic waves transmitted through space. Specifically, the transmitter generates artificial electromagnetic waves by applying time-varying electric currents to its antenna. The waves travel away from the antenna until they eventually reach the antenna of a receiver, which induces an electric current in the receiving antenna. This current can be detected and demodulated to recreate the information sent by the transmitter.

Free-space optical

Main article: Free-space optical communication

An 8-beam free space optics laser link, rated for 1 Gbit/s at a distance of approximately 2 km. The receptor is the large disc in the middle, and the transmitters are the smaller ones. To the top and right corner is a monocular for assisting the alignment of the two heads.

Free-space optical communication (FSO) is an optical communication technology that uses light propagating in free space to transmit wireless data for telecommunications or computer networking. "Free space" means the light beams travel through the open air or outer space. This contrasts with other communication technologies that use light beams traveling through transmission lines such as optical fiber or dielectric "light pipes".

The technology is useful where physical connections are impractical due to high costs or other considerations. For example, free space optical links are used in cities between office buildings that are not wired for networking, where the cost of running cable through the building and under the street would be prohibitive. Another widely used example is consumer IR devices such as remote controls and IrDA (Infrared Data Association) networking, which is used as an alternative to WiFi networking to allow laptops, PDAs, printers, and digital cameras to exchange data.

Sonic

Sonic, especially ultrasonic short-range communication involves the transmission and reception of sound.

Electromagnetic induction

Electromagnetic induction only allows short-range communication and power transmission. It has been used in biomedical situations such as pacemakers, as well as for short-range RFID tags.

Services

Common examples of wireless equipment include:[17]

- Infrared and ultrasonic remote control devices
- Professional LMR (Land Mobile Radio) and SMR (Specialized Mobile Radio) are typically used by business, industrial, and Public Safety entities.
- Consumer Two-way radio including FRS Family Radio Service, GMRS (General Mobile Radio Service), and Citizens band ("CB") radios.
- The Amateur Radio Service (Ham radio).
- Consumer and professional Marine VHF radios.
- Airband and radio navigation equipment used by aviators and air traffic control
- Cellular telephones and pagers: provide connectivity for portable and mobile applications, both personal and business.
- Global Positioning System (GPS): allows drivers of cars and trucks, captains of boats and ships, and pilots of aircraft to ascertain their location anywhere on earth.[18]
- Cordless computer peripherals: the cordless mouse is a common example; wireless headphones, keyboards, and printers can also be linked to a computer via wireless using technology such as Wireless USB or Bluetooth.
- Cordless telephone sets: these are limited-range devices, not to be confused with cell phones.
- Satellite television: Is broadcast from satellites in geostationary orbit. Typical services use direct broadcast satellite to provide multiple television channels to viewers.

Electromagnetic spectrum

See also: Spectrum management

AM and FM radios and other electronic devices make use of the electromagnetic spectrum. The frequencies of the radio spectrum that are available for use for communication are treated as a public resource and are regulated by organizations such as the American Federal Communications Commission, Ofcom in the United Kingdom, the international ITU-R or the European ETSI. Their regulations determine which frequency ranges can be used for what purpose and by whom. In the absence of such control or alternative arrangements such as a privatized electromagnetic spectrum, chaos might result if, for example, airlines did not have specific frequencies to work under and an amateur radio operator was interfering with a pilot's ability to land an aircraft. Wireless communication spans the spectrum from 9 kHz to 300 GHz.[citation needed]

Applications

Mobile telephones

One of the best-known examples of wireless technology is the mobile phone, also known as a cellular phone, with more than 6.6 billion mobile cellular subscriptions worldwide as of the end of 2010.[19] These wireless phones use radio waves from signal-transmission towers to enable their users to make phone calls from many locations worldwide. They can be used within the range of the mobile telephone site used to house the equipment required to transmit and receive the radio signals from these instruments.[20]

Data communications

"Wireless Internet" redirects here. For all wireless Internet access, see Wireless broadband. For mobile wireless Internet, see Mobile broadband.

See also: Radio data communication

Wireless data communications allow wireless networking between desktop computers, laptops, tablet computers, cell phones, and other related devices. The various available technologies differ in local availability, coverage range, and performance,[21] and in some circumstances, users employ multiple connection types and switch between them using connection manager software[22][23] or a mobile VPN to handle the multiple connections as a secure, single virtual network.[24] Supporting technologies include:

Wi-Fi is a wireless local area network that enables portable computing devices to connect easily with other devices, peripherals, and the Internet.[citation needed] Standardized as IEEE 802.11 a, b, g, n, ac, ax, Wi-Fi has link speeds similar to older standards of wired Ethernet. Wi-Fi has become the de facto standard for access in private homes, within offices, and at public hotspots.[25] Some businesses charge customers a monthly fee for service, while others have begun offering it free in an effort to increase the sales of their goods.[26]

Cellular data service offers coverage within a range of 10-15 miles from the nearest cell site.[21] Speeds have increased as technologies have evolved, from earlier technologies such as GSM, CDMA and GPRS, through 3G, to 4G networks such as W-CDMA, EDGE or CDMA2000.[27][28] As of 2018, the proposed next generation is 5G.

Low-power wide-area networks (LPWAN) bridge the gap between Wi-Fi and Cellular for low-bitrate Internet of things (IoT) applications.

Mobile-satellite communications may be used where other wireless connections are unavailable, such as in largely rural areas[29] or remote locations.[21] Satellite communications are especially important for transportation, aviation, maritime and military use.[30]

Wireless sensor networks are responsible for sensing noise, interference, and activity in data collection networks. This allows us to detect relevant quantities, monitor and collect data, formulate clear user displays, and to perform decision-making functions[31]

Wireless data communications are used to span a distance beyond the capabilities of typical cabling in point-to-point communication and point-to-multipoint communication, to provide a backup communications link in case of normal network failure, to link portable or temporary workstations, to overcome situations where normal cabling is difficult or financially impractical, or to remotely connect mobile users or networks.

Peripherals

Peripheral devices in computing can also be connected wirelessly, as part of a Wi-Fi network or directly via an optical or radio-frequency (RF) peripheral interface. Originally these units used bulky, highly local transceivers to mediate between a computer and a keyboard and mouse; however, more recent generations have used smaller, higher-performance devices. Radio-frequency interfaces, such as Bluetooth or Wireless USB, provide greater ranges of efficient use, usually up to 10 feet, but distance, physical obstacles, competing signals, and even human bodies can all degrade the signal quality.[32] Concerns about the security of wireless keyboards arose at the end of 2007 when it was revealed that Microsoft's implementation of encryption in some of its 27 MHz models were highly insecure.[33]

Energy transfer

Main article: Wireless energy transfer

Wireless energy transfer is a process whereby electrical energy is transmitted from a power source to an electrical load that does not have a built-in power source, without the use of interconnecting

wires. There are two different fundamental methods for wireless energy transfer. Energy can be transferred using either far-field methods that involve beaming power/lasers, radio or microwave transmissions, or near-field using electromagnetic induction.[34] Wireless energy transfer may be combined with wireless information transmission in what is known as Wireless Powered Communication.[35] In 2015, researchers at the University of Washington demonstrated far-field energy transfer using Wi-Fi signals to power cameras.[36]

Medical technologies

New wireless technologies, such as mobile body area networks (MBAN), have the capability to monitor blood pressure, heart rate, oxygen level, and body temperature. The MBAN works by sending low-powered wireless signals to receivers that feed into nursing stations or monitoring sites. This technology helps with the intentional and unintentional risk of infection or disconnection that arise from wired connections.[37]

Categories of implementations, devices, and standards

• Cellular networks: 0G, 1G, 2G, 3G, 4G, 5G, 6G
• Cordless telephony: DECT (Digital Enhanced Cordless Telecommunications)
• Land Mobile Radio or Professional Mobile Radio: TETRA, P25, OpenSky, EDACS, DMR, dPMR
• List of emerging technologies
• Radio station in accordance with ITU RR (article 1.61)
• Radiocommunication service in accordance with ITU RR (article 1.19)
• Radio communication system
• Short-range point-to-point communication: Wireless microphones, Remote controls, IrDA, RFID (Radio Frequency Identification), TransferJet, Wireless USB, DSRC (Dedicated Short Range Communications), EnOcean, Near Field Communication
• Wireless sensor networks: Zigbee, EnOcean; Personal area networks, Bluetooth, TransferJet, Ultra-wideband (UWB from WiMedia Alliance).
• Wireless networks: Wireless LAN (WLAN), (IEEE 802.11 branded as Wi-Fi and HiperLAN), Wireless Metropolitan Area Networks (WMAN) and (LMDS, WiMAX, and HiperMAN)

See also

• Comparison of wireless data standards
• Digital radio
• Hotspot (Wi-Fi)
• ISO 15118 (Vehicle to Grid)
• Li-Fi
• MiFi
• Mobile (disambiguation)
• Radio antenna
• Radio resource management (RRM)
• Timeline of radio
• Tuner (radio)
• Wireless access point
• Wireless security
• Wireless Wide Area Network (True wireless)
• WSSUS model

References

1.
• U.S. Army (1944). Technical Manual. US War Department. Retrieved 13 August 2022. In definitions given in the index, p. 162, the term "radio set" is listed as synonymous with the term "wireless set"
• • Paulu, Burton (1956). British Broadcasting: Radio and Television in the United Kingdom. U of Minnesota Press. ISBN 9781452909547. Retrieved 13 August 2022. (p.396) In a public opinion poll in Sweden in 1942, 31.4 percent answered 'Yes' to the question "Do you usually listen to the foreign news on the wireless?'
• • "ATIS Telecom Glossary 2007". atis.org. Archived from the original on 2 March 2008. Retrieved 16 March 2008.
• • Franconi, Nicholas G.; Bunger, Andrew P.; Sejdić, Ervin; Mickle, Marlin H. (24 October 2014).

"Wireless Communication in Oil and Gas Wells". Energy Technology. 2 (12): 996–1005. doi:10.1002/ente.201402067. ISSN 2194-4288. S2CID 111149917.
• • Biswas, S.; Tatchikou, R.; Dion, F. (January 2006). "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety". IEEE Communications Magazine. 44 (1): 74–82. doi:10.1109/mcom.2006.1580935. ISSN 0163-6804. S2CID 6076106.
• • Amédée Guillemin (1891). Electricity and Magnetism. Macmillan and Company. p. 31. Retrieved 17 April 2021.
• • Icons of Invention: The Makers of the Modern World from Gutenberg to Gates. ABC-CLIO. 2009. p. 162. ISBN 978-0-313-34743-6.
• • "Milestones: First Millimeter-wave Communication Experiments by J.C. Bose, 1894-96". List of IEEE milestones. Institute of Electrical and Electronics Engineers. Retrieved 1 October 2019.
• • Emerson, D. T. (1997). "The work of Jagadis Chandra Bose: 100 years of mm-wave research". 1997 IEEE MTT-S International Microwave Symposium Digest, Vol. 3. IEEE Transactions on Microwave Theory and Research. Vol. 45, no. 12. pp. 2267–2273. Bibcode:1997imsd.conf..553E. doi:10.1109/MWSYM.1997.602853. ISBN 9780986488511. S2CID 9039614. reprinted in Igor Grigorov, Ed., Antentop, Vol. 2, No.3, pp. 87–96.
• • "Timeline". The Silicon Engine. Computer History Museum. Retrieved 22 August 2019.
• • "1901: Semiconductor Rectifiers Patented as "Cat's Whisker" Detectors". The Silicon Engine. Computer History Museum. Retrieved 23 August 2019.
• • Golio, Mike; Golio, Janet (2018). RF and Microwave Passive and Active Technologies. CRC Press. pp. ix, I-1, 18–2. ISBN 9781420006728.
• • Rappaport, T. S. (November 1991). "The wireless revolution". IEEE Communications Magazine. 29 (11): 52–71. doi:10.1109/35.109666. S2CID 46573735.
• • "The wireless revolution". The Economist. 21 January 1999. Retrieved 12 September 2019.
• • Baliga, B. Jayant (2005). Silicon RF Power MOSFETS. World Scientific. ISBN 9789812561213.
• • Harvey, Fiona (8 May 2003). "The Wireless Revolution". Encyclopædia Britannica. Retrieved 12 September 2019.
• • Tech Target – Definition of Wireless – Posted by Margaret Rouse (2 April control and traffic control systems
• • Tsai, Allen. "AT&T Releases Navigator GPS Service with Speech Recognition". Telecom Industry News. Archived from the original on 14 June 2012. Retrieved 2 April 2008.
• • "Robust demand for mobile phone service will continue; UN agency predicts". UN News Centre. 15 February 2010. Retrieved 6 September 2011.
• • Vilorio, Dennis. "You're a what? Tower Climber" (PDF). Occupational Outlook Quarterly. Archived (PDF) from the original on 3 February 2013. Retrieved 6 December 2013.
• • "High Speed Internet on the Road". Archived from the original on 3 September 2011. Retrieved 6 September 2011.
• • "What is Connection Manager?". Microsoft Technet. 28 March 2003. Retrieved 6 September 2011.
• • "Our Products". Unwired Revolution. Archived from the original on 9 January 2012. Retrieved 6 September 2011.
• • "General Dynamics- NetMotion Mobility XE". Archived from the original on 26 September 2011. Retrieved 30 August 2011.
• • "Wi-Fi". Retrieved 6 September 2011.
• • O'Brien, J; Marakas, G.M (2008). Management Information Systems. New York, NY: McGraw-Hill Irwin. p. 239.
• • Aravamudhan, Lachu; Faccin, Stefano; Mononen, Risto; Patil, Basavaraj; Saifullah, Yousuf; Sharma, Sarvesh; Sreemanthula, Srinivas (4 July 2003). "Getting to Know Wireless Networks and Technology". InformIT. Retrieved 12 July 2011.
• • "What really is a Third Generation (3G) Mobile Technology" (PDF). ITU. Archived from the original (PDF) on 7 June 2011. Retrieved 12 July 2011.
• • Geier, Jim (2008). "Wireless Network Industry Report 2007" (PDF). Wireless-Nets, Ltd. Archived from the original (PDF) on 12 October 2012. Retrieved 6 September 2011.
• • Ilcev, Stojce Dimov (2006). Global Mobile Satellite Communications for Maritime, Land and Aeronautical Applications. Springer. ISBN 9781402027840.

• • Lewis, F.L. (2004). "Wireless Sensor Networks" (PDF). Smart Environments: Technologies, Protocols, and Applications. New York: John Wiley: 11–46. doi:10.1002/047168659X.ch2. ISBN 9780471686590.
• • Paventi, Jared (26 October 2013). "How does a Wireless Keyboard Work?". Ehow.
• • Moser, Max; Schrödel, Philipp (5 December 2007). "27Mhz Wireless Keyboard Analysis Report aka 'We know what you typed last summer'" (PDF). Archived from the original (PDF) on 23 January 2009. Retrieved 6 February 2012.
• • Jones, George (14 September 2010). "Future Proof: How Wireless Energy Transfer Will Kill the Power Cable". MaximumPC.
• • Dusit Niyato; Lotfollah Shafai (2017). Wireless-Powered Communication Networks. Cambridge University Press. p. 329. ISBN 978-1-107-13569-7. Retrieved 17 April 2021.
• • "First Demonstration of a Surveillance Camera Powered by Ordinary Wi-Fi Broadcasts". MIT Technology Review. Retrieved 20 November 2020.
37. • Linebaugh, Kate (23 May 2012). "More Hospital Medical Devices to Go Wireless". The Wall Street Journal. Retrieved 13 May 2022.

Further reading

• Geier, Jim (2001). Wireless LANs. Sams. ISBN 0-672-32058-4.
• Goldsmith, Andrea (2005). Wireless Communications. Cambridge University Press. ISBN 0-521-83716-2.
• Larsson, Erik; Stoica, Petre (2003). Space-Time Block Coding For Wireless Communications. Cambridge University Press.
• Molisch, Andreas (2005). Wireless Communications. Wiley-IEEE Press. ISBN 0-470-84888-X.
• Pahlavan, Kaveh; Krishnamurthy, Prashant (2002). Principles of Wireless Networks – a Unified Approach. Prentice Hall. ISBN 0-13-093003-2.
• Pahlavan, Kaveh; Levesque, Allen H (1995). Wireless Information Networks. John Wiley & Sons. ISBN 0-471-10607-0.
• Rappaport, Theodore (2002). Wireless Communications: Principles and Practice. Prentice Hall. ISBN 0-13-042232-0.
• Rhoton, John (2001). The Wireless Internet Explained. Digital Press. ISBN 1-55558-257-5.
• Tse, David; Viswanath, Pramod (2005). Fundamentals of Wireless Communication. Cambridge University Press. ISBN 0-521-84527-0.

External links

Look up wireless in Wiktionary, the free dictionary.

• Bibliography - History of wireless and radio broadcasting
• Nets, Webs and the Information Infrastructure at Wikibooks
• Sir Jagadis Chandra Bose - The man who (almost) invented the radio

• v
• t
• e

Telecommunications

Authority control databases

Categories:

• Wireless
• History of radio
• Television terminology
• This page was last edited on 13 Novem

•
•

- Donate
- Create account
- Log in

Contents

•
References

•
Bibliography

•
External links

Neural network (machine learning)

• **Article**
• **Talk**
• **Read**
• **Edit**
• **View history**
Tools
•
•
•
•
•
•
•
•
•
•
•
•
•
•
•
•
Appearance
Text
•
Small
Standard
Large
Width
•
Standard
Wide
Color (beta)
•
Automatic
Light
Dark

From Wikipedia, the free encyclopedia
This article is about the computational models used for artificial intelligence. For other uses, see Neural network (disambiguation).
An artificial neural network is an interconnected group of nodes, inspired by a simplification of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow

represents a connection from the output of one artificial neuron to the input of another.

Part of a series on
Machine learning
and data mining

Paradigms

Problems

Supervised learning
(classification • regression)

Clustering

Dimensionality reduction

Structured prediction

Anomaly detection

Artificial neural network

- **Autoencoder**
- **Deep learning**
- **Feedforward neural network**
- **Recurrent neural network**
  - **LSTM**
  - **GRU**
  - **ESN**
  - **reservoir computing**
- **Boltzmann machine**
  - **Restricted**
- **GAN**
- **Diffusion model**
- **SOM**
- **Convolutional neural network**
  - **U-Net**
  - **LeNet**
  - **AlexNet**
  - **DeepDream**
- **Neural radiance field**
- **Transformer**
  - **Vision**
- **Mamba**
- **Spiking neural network**
- **Memtransistor**
- **Electrochemical RAM (ECRAM)**

Reinforcement learning

Learning with humans

Model diagnostics

Mathematical foundations

Journals and conferences

Related articles
- v
- t
- e

In machine learning, a neural network (also artificial neural network or neural net, abbreviated ANN or NN) is a model inspired by the structure and function of biological neural networks in animal brains.[1][2]
An ANN consists of connected units or nodes called artificial neurons, which loosely model the neurons in the brain. These are connected by edges, which model the synapses in the brain. Each artificial neuron receives signals from connected neurons, then processes them and sends a signal to other connected neurons. The "signal" is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs, called the activation function. The strength of the signal at each connection is determined by a weight, which adjusts during the learning process.
Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly passing through multiple intermediate layers (hidden layers). A network is typically called a deep neural network if it has at least two hidden layers.[3]
Artificial neural networks are used for various tasks, including predictive modeling, adaptive control, and solving problems in artificial intelligence. They can learn from experience, and can derive conclusions from a complex and seemingly unrelated set of information.
Training
Neural networks are typically trained through empirical risk minimization. This method is based on the idea of optimizing the network's parameters to minimize the difference, or empirical risk, between the predicted output and the actual target values in a given dataset.[4] Gradient-based methods such as backpropagation are usually used to estimate the parameters of the network.[4] During the training phase, ANNs learn from labeled training data by iteratively updating their parameters to minimize a defined loss function.[5] This method allows the network to generalize to unseen data.

Simplified example of training a neural network in object detection: The network is trained by multiple images that are known to depict starfish and sea urchins, which are correlated with "nodes" that represent visual features. The starfish match with a ringed texture and a star outline, whereas most sea urchins match with a striped texture and oval shape. However, the instance of a ring textured sea urchin creates a weakly weighted association between them.

Subsequent run of the network on an input image (left):[6] The network correctly detects the starfish. However, the weakly weighted association between ringed texture and sea urchin also confers a weak signal to the latter from one of two intermediate nodes. In addition, a shell that was not included in the training gives a weak signal for the oval shape, also resulting in a weak signal for the sea urchin output. These weak signals may result in a false positive result for sea urchin.
In reality, textures and outlines would not be represented by single nodes, but rather by associated

weight patterns of multiple nodes.
History
Main article: History of artificial neural networks
Early work
Today's deep neural networks are based on early work in statistics over 200 years ago. The simplest kind of feedforward neural network (FNN) is a linear network, which consists of a single layer of output nodes with linear activation functions; the inputs are fed directly to the outputs via a series of weights. The sum of the products of the weights and the inputs is calculated at each node. The mean squared errors between these calculated outputs and the given target values are minimized by creating an adjustment to the weights. This technique has been known for over two centuries as the method of least squares or linear regression. It was used as a means of finding a good rough linear fit to a set of points by Legendre (1805) and Gauss (1795) for the prediction of planetary movement.[7][8][9][10][11]
Historically, digital computers such as the von Neumann model operate via the execution of explicit instructions with access to memory by a number of processors. Some neural networks, on the other hand, originated from efforts to model information processing in biological systems through the framework of connectionism. Unlike the von Neumann model, connectionist computing does not separate memory and processing.
Warren McCulloch and Walter Pitts[12] (1943) considered a non-learning computational model for neural networks.[13] This model paved the way for research to split into two approaches. One approach focused on biological processes while the other focused on the application of neural networks to artificial intelligence.
In the late 1940s, D. O. Hebb[14] proposed a learning hypothesis based on the mechanism of neural plasticity that became known as Hebbian learning. It was used in many early neural networks, such as Rosenblatt's perceptron and the Hopfield network. Farley and Clark[15] (1954) used computational machines to simulate a Hebbian network. Other neural network computational machines were created by Rochester, Holland, Habit and Duda (1956).[16]
In 1958, psychologist Frank Rosenblatt described the perceptron, one of the first implemented artificial neural networks,[17][18][19][20] funded by the United States Office of Naval Research.[21] R. D. Joseph (1960)[22] mentions an even earlier perceptron-like device by Farley and Clark:[10] "Farley and Clark of MIT Lincoln Laboratory actually preceded Rosenblatt in the development of a perceptron-like device." However, "they dropped the subject." The perceptron raised public excitement for research in Artificial Neural Networks, causing the US government to drastically increase funding. This contributed to "the Golden Age of AI" fueled by the optimistic claims made by computer scientists regarding the ability of perceptrons to emulate human intelligence.[23]
The first perceptrons did not have adaptive hidden units. However, Joseph (1960)[22] also discussed multilayer perceptrons with an adaptive hidden layer. Rosenblatt (1962)[24]: section 16 cited and adopted these ideas, also crediting work by H. D. Block and B. W. Knight. Unfortunately, these early efforts did not lead to a working learning algorithm for hidden units, i.e., deep learning.
Deep learning breakthroughs in the 1960s and 1970s
Fundamental research was conducted on ANNs in the 1960s and 1970s. The first working deep learning algorithm was the Group method of data handling, a method to train arbitrarily deep neural networks, published by Alexey Ivakhnenko and Lapa in Ukraine (1965). They regarded it as a form of polynomial regression,[25] or a generalization of Rosenblatt's perceptron.[26] A 1971 paper described a deep network with eight layers trained by this method,[27] which is based on layer by layer training through regression analysis. Superfluous hidden units are pruned using a separate validation set. Since the activation functions of the nodes are Kolmogorov-Gabor polynomials, these were also the first deep networks with multiplicative units or "gates."[10]
The first deep learning multilayer perceptron trained by stochastic gradient descent[28] was published in 1967 by Shun'ichi Amari.[29] In computer experiments conducted by Amari's student Saito, a five layer MLP with two modifiable layers learned internal representations to classify non-linearly separable pattern classes.[10] Subsequent developments in hardware and hyperparameter tunings have made end-to-end stochastic gradient descent the currently dominant training technique.

In 1969, Kunihiko Fukushima introduced the ReLU (rectified linear unit) activation function.[10][30][31] The rectifier has become the most popular activation function for deep learning.[32]
Nevertheless, research stagnated in the United States following the work of Minsky and Papert (1969),[33] who emphasized that basic perceptrons were incapable of processing the exclusive-or circuit. This insight was irrelevant for the deep networks of Ivakhnenko (1965) and Amari (1967).
In 1976 transfer learning was introduced in neural networks learning. [34] [35]
Deep learning architectures for convolutional neural networks (CNNs) with convolutional layers and downsampling layers and weight replication began with the Neocognitron introduced by Kunihiko Fukushima in 1979, though not trained by backpropagation.[36][37][38]
Backpropagation
Backpropagation is an efficient application of the chain rule derived by Gottfried Wilhelm Leibniz in 1673[39] to networks of differentiable nodes. The terminology "back-propagating errors" was actually introduced in 1962 by Rosenblatt,[24] but he did not know how to implement this, although Henry J. Kelley had a continuous precursor of backpropagation in 1960 in the context of control theory.[40] In 1970, Seppo Linnainmaa published the modern form of backpropagation in his master thesis (1970).[41][42][10] G.M. Ostrovski et al. republished it in 1971.[43][44] Paul Werbos applied backpropagation to neural networks in 1982[45][46] (his 1974 PhD thesis, reprinted in a 1994 book,[47] did not yet describe the algorithm[44]). In 1986, David E. Rumelhart et al. popularised backpropagation but did not cite the original work.[48]
Convolutional neural networks
Kunihiko Fukushima's convolutional neural network (CNN) architecture of 1979[36] also introduced max pooling,[49] a popular downsampling procedure for CNNs. CNNs have become an essential tool for computer vision.
The time delay neural network (TDNN) was introduced in 1987 by Alex Waibel to apply CNN to phoneme recognition. It used convolutions, weight sharing, and backpropagation.[50][51] In 1988, Wei Zhang applied a backpropagation-trained CNN to alphabet recognition.[52] In 1989, Yann LeCun et al. created a CNN called LeNet for recognizing handwritten ZIP codes on mail. Training required 3 days.[53] In 1990, Wei Zhang implemented a CNN on optical computing hardware.[54] In 1991, a CNN was applied to medical image object segmentation[55] and breast cancer detection in mammograms.[56] LeNet-5 (1998), a 7-level CNN by Yann LeCun et al., that classifies digits, was applied by several banks to recognize hand-written numbers on checks digitized in 32×32 pixel images.[57]
From 1988 onward,[58][59] the use of neural networks transformed the field of protein structure prediction, in particular when the first cascading networks were trained on profiles (matrices) produced by multiple sequence alignments.[60]
Recurrent neural networks
One origin of RNN was statistical mechanics. In 1972, Shun'ichi Amari proposed to modify the weights of an Ising model by Hebbian learning rule as a model of associative memory, adding in the component of learning.[61] This was popularized as the Hopfield network by John Hopfield(1982).[62] Another origin of RNN was neuroscience. The word "recurrent" is used to describe loop-like structures in anatomy. In 1901, Cajal observed "recurrent semicircles" in the cerebellar cortex.[63] Hebb considered "reverberating circuit" as an explanation for short-term memory.[64] The McCulloch and Pitts paper (1943) considered neural networks that contains cycles, and noted that the current activity of such networks can be affected by activity indefinitely far in the past.[12]
In 1982 a recurrent neural network, with an array architecture (rather than a multilayer perceptron architecture), named Crossbar Adaptive Array [65][66] used direct recurrent connections from the output to the supervisor (teaching) inputs. In addition of computing actions (decisions), it computed internal state evaluations (emotions) of the consequence situations. Eliminating the external supervisor, it introduced the self-learning method in neural networks.
In cognitive psychology, the journal American Psychologist in early 1980's carried out a debate on relation between cognition and emotion. Zajonc in 1980 stated that emotion is computed first and is independent from cognition, while Lazarus in 1982 stated that cognition is computed first and is inseparable from emotion. [67][68] In 1982 the Crossbar Adaptive Array gave a neural network model of cognition-emotion relation. [65][69] It was an example of a debate where an AI system, a

recurrent neural network, contributed to an issue in the same time addressed by cognitive psychology.
Two early influential works were the Jordan network (1986) and the Elman network (1990), which applied RNN to study cognitive psychology.
In the 1980s, backpropagation did not work well for deep RNNs. To overcome this problem, in 1991, Jürgen Schmidhuber proposed the "neural sequence chunker" or "neural history compressor"[70][71] which introduced the important concepts of self-supervised pre-training (the "P" in ChatGPT) and neural knowledge distillation.[10] In 1993, a neural history compressor system solved a "Very Deep Learning" task that required more than 1000 subsequent layers in an RNN unfolded in time.[72]
In 1991, Sepp Hochreiter's diploma thesis [73] identified and analyzed the vanishing gradient problem[73][74] and proposed recurrent residual connections to solve it. He and Schmidhuber introduced long short-term memory (LSTM), which set accuracy records in multiple applications domains.[75][76] This was not yet the modern version of LSTM, which required the forget gate, which was introduced in 1999.[77] It became the default choice for RNN architecture.
During 1985–1995, inspired by statistical mechanics, several architectures and methods were developed by Terry Sejnowski, Peter Dayan, Geoffrey Hinton, etc., including the Boltzmann machine,[78] restricted Boltzmann machine,[79] Helmholtz machine,[80] and the wake-sleep algorithm.[81] These were designed for unsupervised learning of deep generative models.
Deep learning
Between 2009 and 2012, ANNs began winning prizes in image recognition contests, approaching human level performance on various tasks, initially in pattern recognition and handwriting recognition.[82][83] In 2011, a CNN named DanNet[84][85] by Dan Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber achieved for the first time superhuman performance in a visual pattern recognition contest, outperforming traditional methods by a factor of 3.[38] It then won more contests.[86][87] They also showed how max-pooling CNNs on GPU improved performance significantly.[88]
In October 2012, AlexNet by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton[89] won the large-scale ImageNet competition by a significant margin over shallow machine learning methods. Further incremental improvements included the VGG-16 network by Karen Simonyan and Andrew Zisserman[90] and Google's Inceptionv3.[91]
In 2012, Ng and Dean created a network that learned to recognize higher-level concepts, such as cats, only from watching unlabeled images.[92] Unsupervised pre-training and increased computing power from GPUs and distributed computing allowed the use of larger networks, particularly in image and visual recognition problems, which became known as "deep learning".[5]
Radial basis function and wavelet networks were introduced in 2013. These can be shown to offer best approximation properties and have been applied in nonlinear system identification and classification applications.[93]
Generative adversarial network (GAN) (Ian Goodfellow et al., 2014)[94] became state of the art in generative modeling during 2014–2018 period. The GAN principle was originally published in 1991 by Jürgen Schmidhuber who called it "artificial curiosity": two neural networks contest with each other in the form of a zero-sum game, where one network's gain is the other network's loss.[95][96] The first network is a generative model that models a probability distribution over output patterns. The second network learns by gradient descent to predict the reactions of the environment to these patterns. Excellent image quality is achieved by Nvidia's StyleGAN (2018)[97] based on the Progressive GAN by Tero Karras et al.[98] Here, the GAN generator is grown from small to large scale in a pyramidal fashion. Image generation by GAN reached popular success, and provoked discussions concerning deepfakes.[99] Diffusion models (2015)[100] eclipsed GANs in generative modeling since then, with systems such as DALL•E 2 (2022) and Stable Diffusion (2022).
In 2014, the state of the art was training "very deep neural network" with 20 to 30 layers.[101] Stacking too many layers led to a steep reduction in training accuracy,[102] known as the "degradation" problem.[103] In 2015, two techniques were developed to train very deep networks: the highway network was published in May 2015,[104] and the residual neural network (ResNet) in December 2015.[105][106] ResNet behaves like an open-gated Highway Net.

Main article: Transformer (deep learning architecture) § History
During the 2010s, the seq2seq model was developed, and attention mechanisms were added. It led to the modern Transformer architecture in 2017 in Attention Is All You Need.[107] It requires computation time that is quadratic in the size of the context window. Jürgen Schmidhuber's fast weight controller (1992)[108] scales linearly and was later shown to be equivalent to the unnormalized linear Transformer.[109][110][10] Transformers have increasingly become the model of choice for natural language processing.[111] Many modern large language models such as ChatGPT, GPT-4, and BERT use this architecture.

Models

This section may be confusing or unclear to readers. Please help clarify the section. There might be a discussion about this on the talk page. (April 2017) (Learn how and when to remove this message)

Further information: Mathematics of artificial neural networks
Neuron and myelinated axon, with signal flow from inputs at dendrites to outputs at axon terminals
ANNs began as an attempt to exploit the architecture of the human brain to perform tasks that conventional algorithms had little success with. They soon reoriented towards improving empirical results, abandoning attempts to remain true to their biological precursors. ANNs have the ability to learn and model non-linearities and complex relationships. This is achieved by neurons being connected in various patterns, allowing the output of some neurons to become the input of others. The network forms a directed, weighted graph.[112]

An artificial neural network consists of simulated neurons. Each neuron is connected to other nodes via links like a biological axon-synapse-dendrite connection. All the nodes connected by links take in some data and use it to perform specific operations and tasks on the data. Each link has a weight, determining the strength of one node's influence on another,[113] allowing weights to choose the signal between neurons.

Artificial neurons

ANNs are composed of artificial neurons which are conceptually derived from biological neurons. Each artificial neuron has inputs and produces a single output which can be sent to multiple other neurons.[114] The inputs can be the feature values of a sample of external data, such as images or documents, or they can be the outputs of other neurons. The outputs of the final output neurons of the neural net accomplish the task, such as recognizing an object in an image.[citation needed]

To find the output of the neuron we take the weighted sum of all the inputs, weighted by the weights of the connections from the inputs to the neuron. We add a bias term to this sum.[115] This weighted sum is sometimes called the activation. This weighted sum is then passed through a (usually nonlinear) activation function to produce the output. The initial inputs are external data, such as images and documents. The ultimate outputs accomplish the task, such as recognizing an object in an image.[116]

Organization

The neurons are typically organized into multiple layers, especially in deep learning. Neurons of one layer connect only to neurons of the immediately preceding and immediately following layers. The layer that receives external data is the input layer. The layer that produces the ultimate result is the output layer. In between them are zero or more hidden layers. Single layer and unlayered networks are also used. Between two layers, multiple connection patterns are possible. They can be 'fully connected', with every neuron in one layer connecting to every neuron in the next layer. They can be pooling, where a group of neurons in one layer connects to a single neuron in the next layer, thereby reducing the number of neurons in that layer.[117] Neurons with only such connections form a directed acyclic graph and are known as feedforward networks.[118] Alternatively, networks that allow connections between neurons in the same or previous layers are known as recurrent networks.[119]

Hyperparameter

Main article: Hyperparameter (machine learning)
A hyperparameter is a constant parameter whose value is set before the learning process begins. The values of parameters are derived via learning. Examples of hyperparameters include learning rate, the number of hidden layers and batch size.[citation needed] The values of some

hyperparameters can be dependent on those of other hyperparameters. For example, the size of some layers can depend on the overall number of layers.[citation needed]

Learning

This section includes a list of references, related reading, or external links, but its sources remain unclear because it lacks inline citations. Please help improve this section by introducing more precise citations. (August 2019) (Learn how and when to remove this message)

See also: Mathematical optimization, Estimation theory, and Machine learning

Learning is the adaptation of the network to better handle a task by considering sample observations. Learning involves adjusting the weights (and optional thresholds) of the network to improve the accuracy of the result. This is done by minimizing the observed errors. Learning is complete when examining additional observations does not usefully reduce the error rate. Even after learning, the error rate typically does not reach 0. If after learning, the error rate is too high, the network typically must be redesigned. Practically this is done by defining a cost function that is evaluated periodically during learning. As long as its output continues to decline, learning continues. The cost is frequently defined as a statistic whose value can only be approximated. The outputs are actually numbers, so when the error is low, the difference between the output (almost certainly a cat) and the correct answer (cat) is small. Learning attempts to reduce the total of the differences across the observations. Most learning models can be viewed as a straightforward application of optimization theory and statistical estimation.[112][120]

Learning rate

Main article: Learning rate

The learning rate defines the size of the corrective steps that the model takes to adjust for errors in each observation.[121] A high learning rate shortens the training time, but with lower ultimate accuracy, while a lower learning rate takes longer, but with the potential for greater accuracy. Optimizations such as Quickprop are primarily aimed at speeding up error minimization, while other improvements mainly try to increase reliability. In order to avoid oscillation inside the network such as alternating connection weights, and to improve the rate of convergence, refinements use an adaptive learning rate that increases or decreases as appropriate.[122] The concept of momentum allows the balance between the gradient and the previous change to be weighted such that the weight adjustment depends to some degree on the previous change. A momentum close to 0 emphasizes the gradient, while a value close to 1 emphasizes the last change.[citation needed]

Cost function

While it is possible to define a cost function ad hoc, frequently the choice is determined by the function's desirable properties (such as convexity) or because it arises from the model (e.g. in a probabilistic model the model's posterior probability can be used as an inverse cost).[citation needed]

Backpropagation

Main article: Backpropagation

Backpropagation is a method used to adjust the connection weights to compensate for each error found during learning. The error amount is effectively divided among the connections. Technically, backprop calculates the gradient (the derivative) of the cost function associated with a given state with respect to the weights. The weight updates can be done via stochastic gradient descent or other methods, such as extreme learning machines,[123] "no-prop" networks,[124] training without backtracking,[125] "weightless" networks,[126][127] and non-connectionist neural networks. [citation needed]

Learning paradigms

This section includes a list of references, related reading, or external links, but its sources remain unclear because it lacks inline citations. Please help improve this section by introducing more precise citations. (August 2019) (Learn how and when to remove this message)

Machine learning is commonly separated into three main learning paradigms, supervised learning, [128] unsupervised learning[129] and reinforcement learning.[130] Each corresponds to a particular

learning task.
Supervised learning
Supervised learning uses a set of paired inputs and desired outputs. The learning task is to produce the desired output for each input. In this case, the cost function is related to eliminating incorrect deductions.[131] A commonly used cost is the mean-squared error, which tries to minimize the average squared error between the network's output and the desired output. Tasks suited for supervised learning are pattern recognition (also known as classification) and regression (also known as function approximation). Supervised learning is also applicable to sequential data (e.g., for handwriting, speech and gesture recognition). This can be thought of as learning with a "teacher", in the form of a function that provides continuous feedback on the quality of solutions obtained thus far.
Unsupervised learning
In unsupervised learning, input data is given along with the cost function, some function of the data and the network's output. The cost function is dependent on the task (the model domain) and any a priori assumptions (the implicit properties of the model, its parameters and the observed variables). As a trivial example, consider the model where is a constant and the cost . Minimizing this cost produces a value of that is equal to the mean of the data. The cost function can be much more complicated. Its form depends on the application: for example, in compression it could be related to the mutual information between and , whereas in statistical modeling, it could be related to the posterior probability of the model given the data (note that in both of those examples, those quantities would be maximized rather than minimized). Tasks that fall within the paradigm of unsupervised learning are in general estimation problems; the applications include clustering, the estimation of statistical distributions, compression and filtering.
Reinforcement learning
Main article: Reinforcement learning
See also: Stochastic control
In applications such as playing video games, an actor takes a string of actions, receiving a generally unpredictable response from the environment after each one. The goal is to win the game, i.e., generate the most positive (lowest cost) responses. In reinforcement learning, the aim is to weight the network (devise a policy) to perform actions that minimize long-term (expected cumulative) cost. At each point in time the agent performs an action and the environment generates an observation and an instantaneous cost, according to some (usually unknown) rules. The rules and the long-term cost usually only can be estimated. At any juncture, the agent decides whether to explore new actions to uncover their costs or to exploit prior learning to proceed more quickly.
Formally the environment is modeled as a Markov decision process (MDP) with states and actions . Because the state transitions are not known, probability distributions are used instead: the instantaneous cost distribution , the observation distribution and the transition distribution , while a policy is defined as the conditional distribution over actions given the observations. Taken together, the two define a Markov chain (MC). The aim is to discover the lowest-cost MC.
ANNs serve as the learning component in such applications.[132][133] Dynamic programming coupled with ANNs (giving neurodynamic programming)[134] has been applied to problems such as those involved in vehicle routing,[135] video games, natural resource management[136][137] and medicine[138] because of ANNs ability to mitigate losses of accuracy even when reducing the discretization grid density for numerically approximating the solution of control problems. Tasks that fall within the paradigm of reinforcement learning are control problems, games and other sequential decision making tasks.
Self-learning
Self-learning in neural networks was introduced in 1982 along with a neural network capable of self-learning named crossbar adaptive array (CAA).[139] It is a system with only one input, situation s, and only one output, action (or behavior) a. It has neither external advice input nor external reinforcement input from the environment. The CAA computes, in a crossbar fashion, both decisions about actions and emotions (feelings) about encountered situations. The system is driven by the interaction between cognition and emotion.[140] Given the memory matrix, W =||w(a,s)||, the crossbar self-learning algorithm in each iteration performs the following computation:

In situation s perform action a;
Receive consequence situation s';
Compute emotion of being in consequence situation v(s');
Update crossbar memory w'(a,s) = w(a,s) + v(s').

The backpropagated value (secondary reinforcement) is the emotion toward the consequence situation. The CAA exists in two environments, one is behavioral environment where it behaves, and the other is genetic environment, where from it initially and only once receives initial emotions about to be encountered situations in the behavioral environment. Having received the genome vector (species vector) from the genetic environment, the CAA will learn a goal-seeking behavior, in the behavioral environment that contains both desirable and undesirable situations.[141]

Neuroevolution

Main article: Neuroevolution

Neuroevolution can create neural network topologies and weights using evolutionary computation. It is competitive with sophisticated gradient descent approaches.[142][143] One advantage of neuroevolution is that it may be less prone to get caught in "dead ends".[144]

Stochastic neural network

Stochastic neural networks originating from Sherrington–Kirkpatrick models are a type of artificial neural network built by introducing random variations into the network, either by giving the network's artificial neurons stochastic transfer functions [citation needed], or by giving them stochastic weights. This makes them useful tools for optimization problems, since the random fluctuations help the network escape from local minima.[145] Stochastic neural networks trained using a Bayesian approach are known as Bayesian neural networks.[146]

Other

In a Bayesian framework, a distribution over the set of allowed models is chosen to minimize the cost. Evolutionary methods,[147] gene expression programming,[148] simulated annealing,[149] expectation–maximization, non-parametric methods and particle swarm optimization[150] are other learning algorithms. Convergent recursion is a learning algorithm for cerebellar model articulation controller (CMAC) neural networks.[151][152]

Modes

This section includes a list of references, related reading, or external links, but its sources remain unclear because it lacks inline citations. Please help improve this section by introducing more precise citations. (August 2019) (Learn how and when to remove this message)

Two modes of learning are available: stochastic and batch. In stochastic learning, each input creates a weight adjustment. In batch learning weights are adjusted based on a batch of inputs, accumulating errors over the batch. Stochastic learning introduces "noise" into the process, using the local gradient calculated from one data point; this reduces the chance of the network getting stuck in local minima. However, batch learning typically yields a faster, more stable descent to a local minimum, since each update is performed in the direction of the batch's average error. A common compromise is to use "mini-batches", small batches with samples in each batch selected stochastically from the entire data set.

Types

Main article: Types of artificial neural networks

ANNs have evolved into a broad family of techniques that have advanced the state of the art across multiple domains. The simplest types have one or more static components, including number of units, number of layers, unit weights and topology. Dynamic types allow one or more of these to evolve via learning. The latter is much more complicated but can shorten learning periods and produce better results. Some types allow/require learning to be "supervised" by the operator, while others operate independently. Some types operate purely in hardware, while others are purely software and run on general purpose computers.

Some of the main breakthroughs include:

• Convolutional neural networks that have proven particularly successful in processing visual and other two-dimensional data;[153][154] where long short-term memory avoids the vanishing gradient problem[155] and can handle signals that have a mix of low and high frequency components aiding

large-vocabulary speech recognition,[156][157] text-to-speech synthesis,[158][159][160] and photo-real talking heads;[161]

• Competitive networks such as generative adversarial networks in which multiple networks (of varying structure) compete with each other, on tasks such as winning a game[162] or on deceiving the opponent about the authenticity of an input.[94]

Network design

Using artificial neural networks requires an understanding of their characteristics.

• Choice of model: This depends on the data representation and the application. Model parameters include the number, type, and connectedness of network layers, as well as the size of each and the connection type (full, pooling, etc.). Overly complex models learn slowly.

• Learning algorithm: Numerous trade-offs exist between learning algorithms. Almost any algorithm will work well with the correct hyperparameters[163] for training on a particular data set. However, selecting and tuning an algorithm for training on unseen data requires significant experimentation.

• Robustness: If the model, cost function and learning algorithm are selected appropriately, the resulting ANN can become robust.

Neural architecture search (NAS) uses machine learning to automate ANN design. Various approaches to NAS have designed networks that compare well with hand-designed systems. The basic search algorithm is to propose a candidate model, evaluate it against a dataset, and use the results as feedback to teach the NAS network.[164] Available systems include AutoML and AutoKeras.[165] scikit-learn library provides functions to help with building a deep network from scratch. We can then implement a deep network with TensorFlow or Keras.

Hyperparameters must also be defined as part of the design (they are not learned), governing matters such as how many neurons are in each layer, learning rate, step, stride, depth, receptive field and padding (for CNNs), etc.[166]

The Python code snippet provides an overview of the training function, which uses the training dataset, number of hidden layer units, learning rate, and number of iterations as parameters:

```
def train(X, y, n_hidden, learning_rate, n_iter):
m, n_input = X.shape

# 1. random initialize weights and biases
w1 = np.random.randn(n_input, n_hidden)
b1 = np.zeros((1, n_hidden))
w2 = np.random.randn(n_hidden, 1)
b2 = np.zeros((1, 1))

# 2. in each iteration, feed all layers with the latest weights and biases
for i in range(n_iter + 1):
z2 = np.dot(X, w1) + b1
a2 = sigmoid(z2)
z3 = np.dot(a2, w2) + b2
a3 = z3
dz3 = a3 - y
dw2 = np.dot(a2.T, dz3)
db2 = np.sum(dz3, axis=0, keepdims=True)
dz2 = np.dot(dz3, w2.T) * sigmoid_derivative(z2)
dw1 = np.dot(X.T, dz2)
db1 = np.sum(dz2, axis=0)

# 3. update weights and biases with gradients
w1 -= learning_rate * dw1 / m
w2 -= learning_rate * dw2 / m
b1 -= learning_rate * db1 / m
b2 -= learning_rate * db2 / m
```

```
if i % 1000 == 0:
print("Epoch", i, "loss: ", np.mean(np.square(dz3)))

model = {"w1": w1, "b1": b1, "w2": w2, "b2": b2}
return model
```

[citation needed]

Applications

Because of their ability to reproduce and model nonlinear processes, artificial neural networks have found applications in many disciplines. These include:

- Function approximation,[167] or regression analysis,[168] (including time series prediction, fitness approximation,[169] and modeling)
- Data processing[170] (including filtering, clustering, blind source separation,[171] and compression)
- Nonlinear system identification[93] and control (including vehicle control, trajectory prediction,[172] adaptive control, process control, and natural resource management)
- Pattern recognition (including radar systems, face identification, signal classification,[173] novelty detection, 3D reconstruction,[174] object recognition, and sequential decision making[175])
- Sequence recognition (including gesture, speech, and handwritten and printed text recognition[176])
- Sensor data analysis[177] (including image analysis)
- Robotics (including directing manipulators and prostheses)
- Data mining (including knowledge discovery in databases)
- Finance[178] (such as ex-ante models for specific financial long-run forecasts and artificial financial markets)
- Quantum chemistry[179]
- General game playing[180]
- Generative AI[181]
- Data visualization
- Machine translation
- Social network filtering[182]
- E-mail spam filtering
- Medical diagnosis[183]

ANNs have been used to diagnose several types of cancers[184][185] and to distinguish highly invasive cancer cell lines from less invasive lines using only cell shape information.[186][187]
ANNs have been used to accelerate reliability analysis of infrastructures subject to natural disasters[188][189] and to predict foundation settlements.[190] It can also be useful to mitigate flood by the use of ANNs for modelling rainfall-runoff.[191] ANNs have also been used for building black-box models in geoscience: hydrology,[192][193] ocean modelling and coastal engineering,[194][195] and geomorphology.[196] ANNs have been employed in cybersecurity, with the objective to discriminate between legitimate activities and malicious ones. For example, machine learning has been used for classifying Android malware,[197] for identifying domains belonging to threat actors and for detecting URLs posing a security risk.[198] Research is underway on ANN systems designed for penetration testing, for detecting botnets,[199] credit cards frauds[200] and network intrusions.

ANNs have been proposed as a tool to solve partial differential equations in physics[201][202][203] and simulate the properties of many-body open quantum systems.[204][205][206][207] In brain research ANNs have studied short-term behavior of individual neurons,[208] the dynamics of neural circuitry arise from interactions between individual neurons and how behavior can arise from abstract neural modules that represent complete subsystems. Studies considered long-and short-term plasticity of neural systems and their relation to learning and memory from the individual neuron to the system level.

It is possible to create a profile of a user's interests from pictures, using artificial neural networks trained for object recognition.[209]

Beyond their traditional applications, artificial neural networks are increasingly being utilized in

interdisciplinary research, such as materials science. For instance, graph neural networks (GNNs) have demonstrated their capability in scaling deep learning for the discovery of new stable materials by efficiently predicting the total energy of crystals. This application underscores the adaptability and potential of ANNs in tackling complex problems beyond the realms of predictive modeling and artificial intelligence, opening new pathways for scientific discovery and innovation.[210]
Theoretical properties
Computational power
The multilayer perceptron is a universal function approximator, as proven by the universal approximation theorem. However, the proof is not constructive regarding the number of neurons required, the network topology, the weights and the learning parameters.
A specific recurrent architecture with rational-valued weights (as opposed to full precision real number-valued weights) has the power of a universal Turing machine,[211] using a finite number of neurons and standard linear connections. Further, the use of irrational values for weights results in a machine with super-Turing power.[212][213][failed verification]
Capacity
A model's "capacity" property corresponds to its ability to model any given function. It is related to the amount of information that can be stored in the network and to the notion of complexity. Two notions of capacity are known by the community. The information capacity and the VC Dimension. The information capacity of a perceptron is intensively discussed in Sir David MacKay's book[214] which summarizes work by Thomas Cover.[215] The capacity of a network of standard neurons (not convolutional) can be derived by four rules[216] that derive from understanding a neuron as an electrical element. The information capacity captures the functions modelable by the network given any data as input. The second notion, is the VC dimension. VC Dimension uses the principles of measure theory and finds the maximum capacity under the best possible circumstances. This is, given input data in a specific form. As noted in,[214] the VC Dimension for arbitrary inputs is half the information capacity of a Perceptron. The VC Dimension for arbitrary points is sometimes referred to as Memory Capacity.[217]
Convergence
Models may not consistently converge on a single solution, firstly because local minima may exist, depending on the cost function and the model. Secondly, the optimization method used might not guarantee to converge when it begins far from any local minimum. Thirdly, for sufficiently large data or parameters, some methods become impractical.
Another issue worthy to mention is that training may cross some Saddle point which may lead the convergence to the wrong direction.
The convergence behavior of certain types of ANN architectures are more understood than others. When the width of network approaches to infinity, the ANN is well described by its first order Taylor expansion throughout training, and so inherits the convergence behavior of affine models.[218][219] Another example is when parameters are small, it is observed that ANNs often fits target functions from low to high frequencies. This behavior is referred to as the spectral bias, or frequency principle, of neural networks.[220][221][222][223] This phenomenon is the opposite to the behavior of some well studied iterative numerical schemes such as Jacobi method. Deeper neural networks have been observed to be more biased towards low frequency functions.[224]
Generalization and statistics
This section includes a list of references, related reading, or external links, but its sources remain unclear because it lacks inline citations. Please help improve this section by introducing more precise citations. (August 2019) (Learn how and when to remove this message)

Applications whose goal is to create a system that generalizes well to unseen examples, face the possibility of over-training. This arises in convoluted or over-specified systems when the network capacity significantly exceeds the needed free parameters. Two approaches address over-training. The first is to use cross-validation and similar techniques to check for the presence of over-training and to select hyperparameters to minimize the generalization error.
The second is to use some form of regularization. This concept emerges in a probabilistic

(Bayesian) framework, where regularization can be performed by selecting a larger prior probability over simpler models; but also in statistical learning theory, where the goal is to minimize over two quantities: the 'empirical risk' and the 'structural risk', which roughly corresponds to the error over the training set and the predicted error in unseen data due to overfitting.
Confidence analysis of a neural network
Supervised neural networks that use a mean squared error (MSE) cost function can use formal statistical methods to determine the confidence of the trained model. The MSE on a validation set can be used as an estimate for variance. This value can then be used to calculate the confidence interval of network output, assuming a normal distribution. A confidence analysis made this way is statistically valid as long as the output probability distribution stays the same and the network is not modified.
By assigning a softmax activation function, a generalization of the logistic function, on the output layer of the neural network (or a softmax component in a component-based network) for categorical target variables, the outputs can be interpreted as posterior probabilities. This is useful in classification as it gives a certainty measure on classifications.
The softmax activation function is:

Criticism
Training
A common criticism of neural networks, particularly in robotics, is that they require too many training samples for real-world operation.[225] Any learning machine needs sufficient representative examples in order to capture the underlying structure that allows it to generalize to new cases. Potential solutions include randomly shuffling training examples, by using a numerical optimization algorithm that does not take too large steps when changing the network connections following an example, grouping examples in so-called mini-batches and/or introducing a recursive least squares algorithm for CMAC.[151] Dean Pomerleau uses a neural network to train a robotic vehicle to drive on multiple types of roads (single lane, multi-lane, dirt, etc.), and a large amount of his research is devoted to extrapolating multiple training scenarios from a single training experience, and preserving past training diversity so that the system does not become overtrained (if, for example, it is presented with a series of right turns—it should not learn to always turn right).[226]
Theory
A central claim[citation needed] of ANNs is that they embody new and powerful general principles for processing information. These principles are ill-defined. It is often claimed[by whom?] that they are emergent from the network itself. This allows simple statistical association (the basic function of artificial neural networks) to be described as learning or recognition. In 1997, Alexander Dewdney, a former Scientific American columnist, commented that as a result, artificial neural networks have a "something-for-nothing quality, one that imparts a peculiar aura of laziness and a distinct lack of curiosity about just how good these computing systems are. No human hand (or mind) intervenes; solutions are found as if by magic; and no one, it seems, has learned anything".[227] One response to Dewdney is that neural networks have been successfully used to handle many complex and diverse tasks, ranging from autonomously flying aircraft[228] to detecting credit card fraud to mastering the game of Go.
Technology writer Roger Bridgman commented:
Neural networks, for instance, are in the dock not only because they have been hyped to high heaven, (what hasn't?) but also because you could create a successful net without understanding how it worked: the bunch of numbers that captures its behaviour would in all probability be "an opaque, unreadable table...valueless as a scientific resource".
In spite of his emphatic declaration that science is not technology, Dewdney seems here to pillory neural nets as bad science when most of those devising them are just trying to be good engineers. An unreadable table that a useful machine could read would still be well worth having.[229]
Although it is true that analyzing what has been learned by an artificial neural network is difficult, it is much easier to do so than to analyze what has been learned by a biological neural network.

Moreover, recent emphasis on the explainability of AI has contributed towards the development of methods, notably those based on attention mechanisms, for visualizing and explaining learned neural networks. Furthermore, researchers involved in exploring learning algorithms for neural networks are gradually uncovering generic principles that allow a learning machine to be successful. For example, Bengio and LeCun (2007) wrote an article regarding local vs non-local learning, as well as shallow vs deep architecture.[230]

Biological brains use both shallow and deep circuits as reported by brain anatomy,[231] displaying a wide variety of invariance. Weng[232] argued that the brain self-wires largely according to signal statistics and therefore, a serial cascade cannot catch all major statistical dependencies.

Hardware

Large and effective neural networks require considerable computing resources.[233] While the brain has hardware tailored to the task of processing signals through a graph of neurons, simulating even a simplified neuron on von Neumann architecture may consume vast amounts of memory and storage. Furthermore, the designer often needs to transmit signals through many of these connections and their associated neurons – which require enormous CPU power and time.[citation needed]

Some argue that the resurgence of neural networks in the twenty-first century is largely attributable to advances in hardware: from 1991 to 2015, computing power, especially as delivered by GPGPUs (on GPUs), has increased around a million-fold, making the standard backpropagation algorithm feasible for training networks that are several layers deeper than before.[38] The use of accelerators such as FPGAs and GPUs can reduce training times from months to days.[233][234]

Neuromorphic engineering or a physical neural network addresses the hardware difficulty directly, by constructing non-von-Neumann chips to directly implement neural networks in circuitry. Another type of chip optimized for neural network processing is called a Tensor Processing Unit, or TPU.[235]

Practical counterexamples

Analyzing what has been learned by an ANN is much easier than analyzing what has been learned by a biological neural network. Furthermore, researchers involved in exploring learning algorithms for neural networks are gradually uncovering general principles that allow a learning machine to be successful. For example, local vs. non-local learning and shallow vs. deep architecture.[236]

Hybrid approaches

Advocates of hybrid models (combining neural networks and symbolic approaches) say that such a mixture can better capture the mechanisms of the human mind.[237][238]

Dataset bias

Neural networks are dependent on the quality of the data they are trained on, thus low quality data with imbalanced representativeness can lead to the model learning and perpetuating societal biases.[239][240] These inherited biases become especially critical when the ANNs are integrated into real-world scenarios where the training data may be imbalanced due to the scarcity of data for a specific race, gender or other attribute.[239] This imbalance can result in the model having inadequate representation and understanding of underrepresented groups, leading to discriminatory outcomes that exacerbate societal inequalities, especially in applications like facial recognition, hiring processes, and law enforcement.[240][241] For example, in 2018, Amazon had to scrap a recruiting tool because the model favored men over women for jobs in software engineering due to the higher number of male workers in the field.[241] The program would penalize any resume with the word "woman" or the name of any women's college. However, the use of synthetic data can help reduce dataset bias and increase representation in datasets.[242]

Gallery

-

A single-layer feedforward artificial neural network. Arrows originating from are omitted for clarity. There are p inputs to this network and q outputs. In this system, the value of the qth output, , is calculated as

-

A two-layer feedforward artificial neural network

-

An artificial neural network

•

An ANN dependency graph

•

A single-layer feedforward artificial neural network with 4 inputs, 6 hidden nodes and 2 outputs. Given position state and direction, it outputs wheel based control values.

•

A two-layer feedforward artificial neural network with 8 inputs, 2x8 hidden nodes and 2 outputs. Given position state, direction and other environment values, it outputs thruster based control values.

•

Parallel pipeline structure of CMAC neural network. This learning algorithm can converge in one step.

Recent advancements and future directions

Artificial neural networks (ANNs) have undergone significant advancements, particularly in their ability to model complex systems, handle large data sets, and adapt to various types of applications. Their evolution over the past few decades has been marked by a broad range of applications in fields such as image processing, speech recognition, natural language processing, finance, and medicine.[citation needed]

Image processing

In the realm of image processing, ANNs are employed in tasks such as image classification, object recognition, and image segmentation. For instance, deep convolutional neural networks (CNNs) have been important in handwritten digit recognition, achieving state-of-the-art performance.[243] This demonstrates the ability of ANNs to effectively process and interpret complex visual information, leading to advancements in fields ranging from automated surveillance to medical imaging.[243]

Speech recognition

By modeling speech signals, ANNs are used for tasks like speaker identification and speech-to-text conversion. Deep neural network architectures have introduced significant improvements in large vocabulary continuous speech recognition, outperforming traditional techniques.[243][244] These advancements have enabled the development of more accurate and efficient voice-activated systems, enhancing user interfaces in technology products.[citation needed]

Natural language processing

In natural language processing, ANNs are used for tasks such as text classification, sentiment analysis, and machine translation. They have enabled the development of models that can accurately translate between languages, understand the context and sentiment in textual data, and categorize text based on content.[243][244] This has implications for automated customer service, content moderation, and language understanding technologies.[citation needed]

Control systems

In the domain of control systems, ANNs are used to model dynamic systems for tasks such as system identification, control design, and optimization. For instance, deep feedforward neural networks are important in system identification and control applications.[citation needed]

Finance

Further information: Applications of artificial intelligence § Trading and investment

ANNs are used for stock market prediction and credit scoring:

- In investing, ANNs can process vast amounts of financial data, recognize complex patterns, and forecast stock market trends, aiding investors and risk managers in making informed decisions.[243]
- In credit scoring, ANNs offer data-driven, personalized assessments of creditworthiness, improving the accuracy of default predictions and automating the lending process.[244]

ANNs require high-quality data and careful tuning, and their "black-box" nature can pose challenges in interpretation. Nevertheless, ongoing advancements suggest that ANNs continue to play a role in finance, offering valuable insights and enhancing risk management strategies.[citation needed]

Medicine

ANNs are able to process and analyze vast medical datasets. They enhance diagnostic accuracy, especially by interpreting complex medical imaging for early disease detection, and by predicting patient outcomes for personalized treatment planning.[244] In drug discovery, ANNs speed up the identification of potential drug candidates and predict their efficacy and safety, significantly reducing development time and costs.[243] Additionally, their application in personalized medicine and healthcare data analysis allows tailored therapies and efficient patient care management.[244] Ongoing research is aimed at addressing remaining challenges such as data privacy and model interpretability, as well as expanding the scope of ANN applications in medicine.[citation needed]

Content creation

ANNs such as generative adversarial networks (GAN) and transformers are used for content creation across numerous industries.[245] This is because deep learning models are able to learn the style of an artist or musician from huge datasets and generate completely new artworks and music compositions. For instance, DALL-E is a deep neural network trained on 650 million pairs of images and texts across the internet that can create artworks based on text entered by the user.[246] In the field of music, transformers are used to create original music for commercials and documentaries through companies such as AIVA and Jukedeck.[247] In the marketing industry generative models are used to create personalized advertisements for consumers.[245] Additionally, major film companies are partnering with technology companies to analyze the financial success of a film, such as the partnership between Warner Bros and technology company Cinelytic established in 2020.[248] Furthermore, neural networks have found uses in video game creation, where Non Player Characters (NPCs) can make decisions based on all the characters currently in the game.[249]

See also

- **ADALINE**
- **Autoencoder**
- **Bio-inspired computing**
- **Blue Brain Project**
- **Catastrophic interference**
- **Cognitive architecture**
- **Connectionist expert system**
- **Connectomics**
- **Deep image prior**
- **Digital morphogenesis**
- **Efficiently updatable neural network**
- **Evolutionary algorithm**
- **Genetic algorithm**
- **Hyperdimensional computing**
- **In situ adaptive tabulation**
- **Large width limits of neural networks**
- **List of machine learning concepts**
- **Memristor**
- **Neural gas**
- **Neural network software**
- **Optical neural network**
- **Parallel distributed processing**
- **Philosophy of artificial intelligence**
- **Predictive analytics**
- **Quantum neural network**
- **Support vector machine**
- **Spiking neural network**
- **Stochastic parrot**
- **Tensor product network**

References

1.
• Hardesty L (14 April 2017). "Explained: Neural networks". MIT News Office. Archived from the original on 18 March 2024. Retrieved 2 June 2022.
• • Yang Z, Yang Z (2014). Comprehensive Biomedical Physics. Karolinska Institute, Stockholm, Sweden: Elsevier. p. 1. ISBN 978-0-444-53633-4. Archived from the original on 28 July 2022. Retrieved 28 July 2022.
• • Bishop CM (17 August 2006). Pattern Recognition and Machine Learning. New York: Springer. ISBN 978-0-387-31073-2.
• • Vapnik VN, Vapnik VN (1998). The nature of statistical learning theory (Corrected 2nd print. ed.). New York Berlin Heidelberg: Springer. ISBN 978-0-387-94559-0.
• • Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). Deep Learning. MIT Press. Archived from the original on 16 April 2016. Retrieved 1 June 2016.
• • Ferrie, C., Kaiser, S. (2019). Neural Networks for Babies. Sourcebooks. ISBN 978-1-4926-7120-6.
• • Mansfield Merriman, "A List of Writings Relating to the Method of Least Squares"
• • Stigler SM (1981). "Gauss and the Invention of Least Squares". Ann. Stat. 9 (3): 465–474. doi:10.1214/aos/1176345451.
• • Bretscher O (1995). Linear Algebra With Applications (3rd ed.). Upper Saddle River, NJ: Prentice Hall.
• • Schmidhuber J (2022). "Annotated History of Modern AI and Deep Learning". arXiv:2212.11279 [cs.NE].
• • Stigler SM (1986). The History of Statistics: The Measurement of Uncertainty before 1900. Cambridge: Harvard. ISBN 0-674-40340-1.
• • McCulloch WS, Pitts W (December 1943). "A logical calculus of the ideas immanent in nervous activity". The Bulletin of Mathematical Biophysics. 5 (4): 115–133. doi:10.1007/BF02478259. ISSN 0007-4985.
• • Kleene S (1956). "Representation of Events in Nerve Nets and Finite Automata". Annals of Mathematics Studies. No. 34. Princeton University Press. pp. 3–41. Retrieved 17 June 2017.
• • Hebb D (1949). The Organization of Behavior. New York: Wiley. ISBN 978-1-135-63190-1.
• • Farley B, W.A. Clark (1954). "Simulation of Self-Organizing Systems by Digital Computer". IRE Transactions on Information Theory. 4 (4): 76–84. doi:10.1109/TIT.1954.1057468.
• • Rochester N, J.H. Holland, L.H. Habit, W.L. Duda (1956). "Tests on a cell assembly theory of the action of the brain, using a large digital computer". IRE Transactions on Information Theory. 2 (3): 80–93. doi:10.1109/TIT.1956.1056810.
• • Haykin (2008) Neural Networks and Learning Machines, 3rd edition
• • Rosenblatt F (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization in the Brain". Psychological Review. 65 (6): 386–408. CiteSeerX 10.1.1.588.3775. doi:10.1037/h0042519. PMID 13602029. S2CID 12781225.
• • Werbos P (1975). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.
• • Rosenblatt F (1957). "The Perceptron—a perceiving and recognizing automaton". Report 85-460-1. Cornell Aeronautical Laboratory.
• • Olazaran M (1996). "A Sociological Study of the Official History of the Perceptrons Controversy". Social Studies of Science. 26 (3): 611–659. doi:10.1177/030631296026003005. JSTOR 285702. S2CID 16786738.
• • Joseph RD (1960). Contributions to Perceptron Theory, Cornell Aeronautical Laboratory Report No. VG-11 96--G-7, Buffalo.
• • Russel, Stuart, Norvig, Peter (2010). Artificial Intelligence A Modern Approach (PDF) (3rd ed.). United States of America: Pearson Education. pp. 16–28. ISBN 978-0-13-604259-4.
• • Rosenblatt F (1962). Principles of Neurodynamics. Spartan, New York.
• • Ivakhnenko AG, Lapa VG (1967). Cybernetics and Forecasting Techniques. American Elsevier Publishing Co. ISBN 978-0-444-00020-0.
• • Ivakhnenko A (March 1970). "Heuristic self-organization in problems of engineering cybernetics". Automatica. 6 (2): 207–219. doi:10.1016/0005-1098(70)90092-0.
• • Ivakhnenko A (1971). "Polynomial theory of complex systems" (PDF). IEEE Transactions on

Systems, Man, and Cybernetics. SMC-1 (4): 364–378. doi:10.1109/TSMC.1971.4308320. Archived (PDF) from the original on 29 August 2017. Retrieved 5 November 2019.
• • Robbins H, Monro S (1951). "A Stochastic Approximation Method". The Annals of Mathematical Statistics. 22 (3): 400. doi:10.1214/aoms/1177729586.
• • Amari S (1967). "A theory of adaptive pattern classifier". IEEE Transactions. EC (16): 279–307.
• • Fukushima K (1969). "Visual feature extraction by a multilayered network of analog threshold elements". IEEE Transactions on Systems Science and Cybernetics. 5 (4): 322–333. doi:10.1109/TSSC.1969.300225.
• • Sonoda S, Murata N (2017). "Neural network with unbounded activation functions is universal approximator". Applied and Computational Harmonic Analysis. 43 (2): 233–268. arXiv:1505.03654. doi:10.1016/j.acha.2015.12.005. S2CID 12149203.
• • Ramachandran P, Barret Z, Quoc VL (16 October 2017). "Searching for Activation Functions". arXiv:1710.05941 [cs.NE].
• • Minsky M, Papert S (1969). Perceptrons: An Introduction to Computational Geometry. MIT Press. ISBN 978-0-262-63022-1.
• • Bozinovski S. and Fulgosi A. (1976). "The influence of pattern similarity and transfer learning on the base perceptron training" (original in Croatian) Proceedings of Symposium Informatica 3-121-5, Bled.
• • Bozinovski S.(2020) "Reminder of the first paper on transfer learning in neural networks, 1976". Informatica 44: 291–302.
• • Fukushima K (1979). "Neural network model for a mechanism of pattern recognition unaffected by shift in position—Neocognitron". Trans. IECE (In Japanese). J62-A (10): 658–665. doi:10.1007/bf00344251. PMID 7370364. S2CID 206775608.
• • Fukushima K (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". Biol. Cybern. 36 (4): 193–202. doi:10.1007/bf00344251. PMID 7370364. S2CID 206775608.
• • Schmidhuber J (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks. 61: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.
• • Leibniz GW (1920). The Early Mathematical Manuscripts of Leibniz: Translated from the Latin Texts Published by Carl Immanuel Gerhardt with Critical and Historical Notes (Leibniz published the chain rule in a 1676 memoir). Open court publishing Company. ISBN 9780598818461.
• • Kelley HJ (1960). "Gradient theory of optimal flight paths". ARS Journal. 30 (10): 947–954. doi:10.2514/8.5282.
• • Linnainmaa S (1970). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors (Masters) (in Finnish). University of Helsinki. p. 6–7.
• • Linnainmaa S (1976). "Taylor expansion of the accumulated rounding error". BIT Numerical Mathematics. 16 (2): 146–160. doi:10.1007/bf01931367. S2CID 122357351.
• • Ostrovski, G.M., Volin,Y.M., and Boris, W.W. (1971). On the computation of derivatives. Wiss. Z. Tech. Hochschule for Chemistry, 13:382–384.
• • Schmidhuber J (25 October 2014). "Who Invented Backpropagation?". IDSIA, Switzerland. Archived from the original on 30 July 2024. Retrieved 14 September 2024.
• • Werbos P (1982). "Applications of advances in nonlinear sensitivity analysis" (PDF). System modeling and optimization. Springer. pp. 762–770. Archived (PDF) from the original on 14 April 2016. Retrieved 2 July 2017.
• • Anderson JA, Rosenfeld E, eds. (2000). Talking Nets: An Oral History of Neural Networks. The MIT Press. doi:10.7551/mitpress/6626.003.0016. ISBN 978-0-262-26715-1.
• • Werbos PJ (1994). The Roots of Backpropagation : From Ordered Derivatives to Neural Networks and Political Forecasting. New York: John Wiley & Sons. ISBN 0-471-59897-6.
• • Rumelhart DE, Hinton GE, Williams RJ (October 1986). "Learning representations by back-propagating errors". Nature. 323 (6088): 533–536. Bibcode:1986Natur.323..533R. doi:10.1038/323533a0. ISSN 1476-4687.
• • Fukushima K, Miyake S (1 January 1982). "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position". Pattern Recognition. 15 (6): 455–469. Bibcode:1982PatRe..15..455F. doi:10.1016/0031-3203(82)90024-3. ISSN 0031-3203.

• • Waibel A (December 1987). Phoneme Recognition Using Time-Delay Neural Networks (PDF). Meeting of the Institute of Electrical, Information and Communication Engineers (IEICE). Tokyo, Japan.
• • Alexander Waibel et al., Phoneme Recognition Using Time-Delay Neural Networks IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume 37, No. 3, pp. 328. – 339 March 1989.
• • Zhang W (1988). "Shift-invariant pattern recognition neural network and its optical architecture". Proceedings of Annual Conference of the Japan Society of Applied Physics.
• • LeCun et al., "Backpropagation Applied to Handwritten Zip Code Recognition", Neural Computation, 1, pp. 541–551, 1989.
• • Zhang W (1990). "Parallel distributed processing model with local space-invariant interconnections and its optical architecture". Applied Optics. 29 (32): 4790–7. Bibcode:1990ApOpt..29.4790Z. doi:10.1364/AO.29.004790. PMID 20577468.
• • Zhang W (1991). "Image processing of human corneal endothelium based on a learning network". Applied Optics. 30 (29): 4211–7. Bibcode:1991ApOpt..30.4211Z. doi:10.1364/AO.30.004211. PMID 20706526.
• • Zhang W (1994). "Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant artificial neural network". Medical Physics. 21 (4): 517–24. Bibcode:1994MedPh..21..517Z. doi:10.1118/1.597177. PMID 8058017.
• • LeCun Y, Léon Bottou, Yoshua Bengio, Patrick Haffner (1998). "Gradient-based learning applied to document recognition" (PDF). Proceedings of the IEEE. 86 (11): 2278–2324. CiteSeerX 10.1.1.32.9552. doi:10.1109/5.726791. S2CID 14542261. Retrieved 7 October 2016.
• • Qian, Ning, and Terrence J. Sejnowski. "Predicting the secondary structure of globular proteins using neural network models." Journal of molecular biology 202, no. 4 (1988): 865–884.
• • Bohr, Henrik, Jakob Bohr, Søren Brunak, Rodney MJ Cotterill, Benny Lautrup, Leif Nørskov, Ole H. Olsen, and Steffen B. Petersen. "Protein secondary structure and homology by neural networks The α-helices in rhodopsin." FEBS letters 241, (1988): 223–228
• • Rost, Burkhard, and Chris Sander. "Prediction of protein secondary structure at better than 70% accuracy." Journal of molecular biology 232, no. 2 (1993): 584–599.
• • Amari SI (November 1972). "Learning Patterns and Pattern Sequences by Self-Organizing Nets of Threshold Elements". IEEE Transactions on Computers. C-21 (11): 1197–1206. doi:10.1109/T-C.1972.223477. ISSN 0018-9340.
• • Hopfield JJ (1982). "Neural networks and physical systems with emergent collective computational abilities". Proceedings of the National Academy of Sciences. 79 (8): 2554–2558. Bibcode:1982PNAS...79.2554H. doi:10.1073/pnas.79.8.2554. PMC 346238. PMID 6953413.
• • Espinosa-Sanchez JM, Gomez-Marin A, de Castro F (5 July 2023). "The Importance of Cajal's and Lorente de Nó's Neuroscience to the Birth of Cybernetics". The Neuroscientist. doi:10.1177/10738584231179932. hdl:10261/348372. ISSN 1073-8584. PMID 37403768.
• • "reverberating circuit". Oxford Reference. Retrieved 27 July 2024.
• • Bozinovski, S. (1982). "A self-learning system using secondary reinforcement". In Trappl, Robert (ed.). Cybernetics and Systems Research: Proceedings of the Sixth European Meeting on Cybernetics and Systems Research. North-Holland. pp. 397–402. ISBN 978-0-444-86488-8
• • Bozinovski S. (1995) "Neuro genetic agents and structural theory of self-reinforcement learning systems". CMPSCI Technical Report 95-107, University of Massachusetts at Amherst [1]
• • R. Zajonc (1980) "Feeling and thinking: Preferences need no inferences". American Psychologist 35 (2): 151-175
• • Lazarus R. (1982) "Thoughts on the relations between emotion and cognition" American Psychologist 37 (9): 1019-1024
• • Bozinovski, S. (2014) "Modeling mechanisms of cognition-emotion interaction in artificial neural networks, since 1981" Procedia Computer Science p. 255-263 (https://core.ac.uk/download/pdf/81973924.pdf)
• • Schmidhuber J (April 1991). "Neural Sequence Chunkers" (PDF). TR FKI-148, TU Munich.
• • Schmidhuber J (1992). "Learning complex, extended sequences using the principle of history compression (based on TR FKI-148, 1991)" (PDF). Neural Computation. 4 (2): 234–242.

doi:10.1162/neco.1992.4.2.234. S2CID 18271205.
• • Schmidhuber J (1993). Habilitation thesis: System modeling and optimization (PDF). Page 150 ff demonstrates credit assignment across the equivalent of 1,200 layers in an unfolded RNN.
• • S. Hochreiter., "Untersuchungen zu dynamischen neuronalen Netzen", Archived 2015-03-06 at the Wayback Machine, Diploma thesis. Institut f. Informatik, Technische Univ. Munich. Advisor: J. Schmidhuber, 1991.
• • Hochreiter S, et al. (15 January 2001). "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies". In Kolen JF, Kremer SC (eds.). A Field Guide to Dynamical Recurrent Networks. John Wiley & Sons. ISBN 978-0-7803-5369-5. Archived from the original on 19 May 2024. Retrieved 26 June 2017.
• • Sepp Hochreiter, Jürgen Schmidhuber (21 August 1995), Long Short Term Memory, Wikidata Q98967430
• • Hochreiter S, Schmidhuber J (1 November 1997). "Long Short-Term Memory". Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276. S2CID 1915014.
• • Gers F, Schmidhuber J, Cummins F (1999). "Learning to forget: Continual prediction with LSTM". 9th International Conference on Artificial Neural Networks: ICANN '99. Vol. 1999. pp. 850–855. doi:10.1049/cp:19991218. ISBN 0-85296-721-7.
• • Ackley DH, Hinton GE, Sejnowski TJ (1 January 1985). "A learning algorithm for boltzmann machines". Cognitive Science. 9 (1): 147–169. doi:10.1016/S0364-0213(85)80012-4. ISSN 0364-0213.
• • Smolensky P (1986). "Chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory" (PDF). In Rumelhart DE, McLelland JL (eds.). Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations. MIT Press. pp. 194–281. ISBN 0-262-68053-X.
• • Peter D, Hinton GE, Neal RM, Zemel RS (1995). "The Helmholtz machine". Neural Computation. 7 (5): 889–904. doi:10.1162/neco.1995.7.5.889. hdl:21.11116/0000-0002-D6D3-E. PMID 7584891. S2CID 1890561.
• • Hinton GE, Dayan P, Frey BJ, Neal R (26 May 1995). "The wake-sleep algorithm for unsupervised neural networks". Science. 268 (5214): 1158–1161. Bibcode:1995Sci...268.1158H. doi:10.1126/science.7761831. PMID 7761831. S2CID 871473.
• • 2012 Kurzweil AI Interview Archived 31 August 2018 at the Wayback Machine with Juergen Schmidhuber on the eight competitions won by his Deep Learning team 2009–2012
• • "How bio-inspired deep learning keeps winning competitions | KurzweilAI". kurzweilai.net. Archived from the original on 31 August 2018. Retrieved 16 June 2017.
• • Cireşan DC, Meier U, Gambardella LM, Schmidhuber J (21 September 2010). "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition". Neural Computation. 22 (12): 3207–3220. arXiv:1003.0358. doi:10.1162/neco_a_00052. ISSN 0899-7667. PMID 20858131. S2CID 1918673.
• • Ciresan DC, Meier U, Masci J, Gambardella L, Schmidhuber J (2011). "Flexible, High Performance Convolutional Neural Networks for Image Classification" (PDF). International Joint Conference on Artificial Intelligence. doi:10.5591/978-1-57735-516-8/ijcai11-210. Archived (PDF) from the original on 29 September 2014. Retrieved 13 June 2017.
• • Ciresan D, Giusti A, Gambardella LM, Schmidhuber J (2012). Pereira F, Burges CJ, Bottou L, Weinberger KQ (eds.). Advances in Neural Information Processing Systems 25 (PDF). Curran Associates, Inc. pp. 2843–2851. Archived (PDF) from the original on 9 August 2017. Retrieved 13 June 2017.
• • Ciresan D, Giusti A, Gambardella L, Schmidhuber J (2013). "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks". Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013. Lecture Notes in Computer Science. Vol. 7908. pp. 411–418. doi:10.1007/978-3-642-40763-5_51. ISBN 978-3-642-38708-1. PMID 24579167.
• • Ciresan D, Meier U, Schmidhuber J (2012). "Multi-column deep neural networks for image classification". 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3642–3649. arXiv:1202.2745. doi:10.1109/cvpr.2012.6248110. ISBN 978-1-4673-1228-8. S2CID 2161592.
• • Krizhevsky A, Sutskever I, Hinton G (2012). "ImageNet Classification with Deep Convolutional Neural Networks" (PDF). NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada. Archived (PDF) from the original on 10 January 2017. Retrieved 24 May 2017.

- • Simonyan K, Andrew Z (2014). "Very Deep Convolution Networks for Large Scale Image Recognition". arXiv:1409.1556 [cs.CV].
- • Szegedy C (2015). "Going deeper with convolutions" (PDF). Cvpr2015. arXiv:1409.4842.
- • Ng A, Dean J (2012). "Building High-level Features Using Large Scale Unsupervised Learning". arXiv:1112.6209 [cs.LG].
- • Billings SA (2013). Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains. Wiley. ISBN 978-1-119-94359-4.
- • Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. (2014). Generative Adversarial Networks (PDF). Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680. Archived (PDF) from the original on 22 November 2019. Retrieved 20 August 2019.
- • Schmidhuber J (1991). "A possibility for implementing curiosity and boredom in model-building neural controllers". Proc. SAB'1991. MIT Press/Bradford Books. pp. 222–227.
- • Schmidhuber J (2020). "Generative Adversarial Networks are Special Cases of Artificial Curiosity (1990) and also Closely Related to Predictability Minimization (1991)". Neural Networks. 127: 58–66. arXiv:1906.04493. doi:10.1016/j.neunet.2020.04.008. PMID 32334341. S2CID 216056336.
- • "GAN 2.0: NVIDIA's Hyperrealistic Face Generator". SyncedReview.com. 14 December 2018. Retrieved 3 October 2019.
- • Karras T, Aila T, Laine S, Lehtinen J (26 February 2018). "Progressive Growing of GANs for Improved Quality, Stability, and Variation". arXiv:1710.10196 [cs.NE].
- • "Prepare, Don't Panic: Synthetic Media and Deepfakes". witness.org. Archived from the original on 2 December 2020. Retrieved 25 November 2020.
- • Sohl-Dickstein J, Weiss E, Maheswaranathan N, Ganguli S (1 June 2015). "Deep Unsupervised Learning using Nonequilibrium Thermodynamics" (PDF). Proceedings of the 32nd International Conference on Machine Learning. 37. PMLR: 2256–2265. arXiv:1503.03585.
- • Simonyan K, Zisserman A (10 April 2015), Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv:1409.1556
- • He K, Zhang X, Ren S, Sun J (2016). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". arXiv:1502.01852 [cs.CV].
- • He K, Zhang X, Ren S, Sun J (10 December 2015). Deep Residual Learning for Image Recognition. arXiv:1512.03385.
- • Srivastava RK, Greff K, Schmidhuber J (2 May 2015). "Highway Networks". arXiv:1505.00387 [cs.LG].
- • He K, Zhang X, Ren S, Sun J (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE. pp. 770–778. arXiv:1512.03385. doi:10.1109/CVPR.2016.90. ISBN 978-1-4673-8851-1.
- • Linn A (10 December 2015). "Microsoft researchers win ImageNet computer vision challenge". The AI Blog. Retrieved 29 June 2024.
- • Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. (12 June 2017). "Attention Is All You Need". arXiv:1706.03762 [cs.CL].
- • Schmidhuber J (1992). "Learning to control fast-weight memories: an alternative to recurrent nets" (PDF). Neural Computation. 4 (1): 131–139. doi:10.1162/neco.1992.4.1.131. S2CID 16683347.
- • Katharopoulos A, Vyas A, Pappas N, Fleuret F (2020). "Transformers are RNNs: Fast autoregressive Transformers with linear attention". ICML 2020. PMLR. pp. 5156–5165.
- • Schlag I, Irie K, Schmidhuber J (2021). "Linear Transformers Are Secretly Fast Weight Programmers". ICML 2021. Springer. pp. 9355–9366.
- • Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. (2020). "Transformers: State-of-the-Art Natural Language Processing". Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. doi:10.18653/v1/2020.emnlp-demos.6. S2CID 208117506.
- • Zell A (2003). "chapter 5.2". Simulation neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.). Addison-Wesley. ISBN 978-3-89319-554-1. OCLC 249017987.
- • Artificial intelligence (3rd ed.). Addison-Wesley Pub. Co. 1992. ISBN 0-201-53377-4.
- • Abbod MF (2007). "Application of Artificial Intelligence to the Management of Urological Cancer".

The Journal of Urology. 178 (4): 1150–1156. doi:10.1016/j.juro.2007.05.122. PMID 17698099.
• • Dawson CW (1998). "An artificial neural network approach to rainfall-runoff modelling". Hydrological Sciences Journal. 43 (1): 47–66. Bibcode:1998HydSJ..43...47D. doi:10.1080/02626669809492102.
• • "The Machine Learning Dictionary". cse.unsw.edu.au. Archived from the original on 26 August 2018. Retrieved 4 November 2009.
• • Ciresan D, Ueli Meier, Jonathan Masci, Luca M. Gambardella, Jurgen Schmidhuber (2011). "Flexible, High Performance Convolutional Neural Networks for Image Classification" (PDF). Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Two. 2: 1237–1242. Archived (PDF) from the original on 5 April 2022. Retrieved 7 July 2022.
• • Zell A (1994). Simulation Neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.). Addison-Wesley. p. 73. ISBN 3-89319-554-8.
• • Miljanovic M (February–March 2012). "Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction" (PDF). Indian Journal of Computer and Engineering. 3 (1). Archived (PDF) from the original on 19 May 2024. Retrieved 21 August 2019.
• • Kelleher JD, Mac Namee B, D'Arcy A (2020). "7-8". Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies (2nd ed.). Cambridge, MA: The MIT Press. ISBN 978-0-262-36110-1. OCLC 1162184998.
• • Wei J (26 April 2019). "Forget the Learning Rate, Decay Loss". arXiv:1905.00094 [cs.LG].
• • Li Y, Fu Y, Li H, Zhang SW (1 June 2009). "The Improved Training Algorithm of Back Propagation Neural Network with Self-adaptive Learning Rate". 2009 International Conference on Computational Intelligence and Natural Computing. Vol. 1. pp. 73–76. doi:10.1109/CINC.2009.111. ISBN 978-0-7695-3645-3. S2CID 10557754.
• • Huang GB, Zhu QY, Siew CK (2006). "Extreme learning machine: theory and applications". Neurocomputing. 70 (1): 489–501. CiteSeerX 10.1.1.217.3692. doi:10.1016/j.neucom.2005.12.126. S2CID 116858.
• • Widrow B, et al. (2013). "The no-prop algorithm: A new learning algorithm for multilayer neural networks". Neural Networks. 37: 182–188. doi:10.1016/j.neunet.2012.09.020. PMID 23140797.
• • Ollivier Y, Charpiat G (2015). "Training recurrent networks without backtracking". arXiv:1507.07680 [cs.NE].
• • Hinton GE (2010). "A Practical Guide to Training Restricted Boltzmann Machines". Tech. Rep. UTML TR 2010-003. Archived from the original on 9 May 2021. Retrieved 27 June 2017.
• • ESANN. 2009.[full citation needed]
• • Bernard E (2021). Introduction to machine learning. Champaign: Wolfram Media. p. 9. ISBN 978-1-57955-048-6. Archived from the original on 19 May 2024. Retrieved 22 March 2023.
• • Bernard E (2021). Introduction to machine learning. Champaign: Wolfram Media. p. 12. ISBN 978-1-57955-048-6. Archived from the original on 19 May 2024. Retrieved 22 March 2023.
• • Bernard E (2021). Introduction to Machine Learning. Wolfram Media Inc. p. 9. ISBN 978-1-57955-048-6. Archived from the original on 19 May 2024. Retrieved 28 July 2022.
• • Ojha VK, Abraham A, Snášel V (1 April 2017). "Metaheuristic design of feedforward neural networks: A review of two decades of research". Engineering Applications of Artificial Intelligence. 60: 97–116. arXiv:1705.05584. Bibcode:2017arXiv170505584O. doi:10.1016/j.engappai.2017.01.013. S2CID 27910748.
• • Dominic, S., Das, R., Whitley, D., Anderson, C. (July 1991). "Genetic reinforcement learning for neural networks". IJCNN-91-Seattle International Joint Conference on Neural Networks. IJCNN-91-Seattle International Joint Conference on Neural Networks. Seattle, Washington, US: IEEE. pp. 71–76. doi:10.1109/IJCNN.1991.155315. ISBN 0-7803-0164-1.
• • Hoskins J, Himmelblau, D.M. (1992). "Process control via artificial neural networks and reinforcement learning". Computers & Chemical Engineering. 16 (4): 241–251. doi:10.1016/0098-1354(92)80045-B.
• • Bertsekas D, Tsitsiklis J (1996). Neuro-dynamic programming. Athena Scientific. p. 512. ISBN 978-1-886529-10-6. Archived from the original on 29 June 2017. Retrieved 17 June 2017.
• • Secomandi N (2000). "Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands". Computers & Operations Research. 27 (11–12): 1201–

1225. CiteSeerX 10.1.1.392.4034. doi:10.1016/S0305-0548(99)00146-X.
• • de Rigo, D., Rizzoli, A. E., Soncini-Sessa, R., Weber, E., Zenesi, P. (2001). "Neuro-dynamic programming for the efficient management of reservoir networks". Proceedings of MODSIM 2001, International Congress on Modelling and Simulation. MODSIM 2001, International Congress on Modelling and Simulation. Canberra, Australia: Modelling and Simulation Society of Australia and New Zealand. doi:10.5281/zenodo.7481. ISBN 0-86740-525-2. Archived from the original on 7 August 2013. Retrieved 29 July 2013.
• • Damas, M., Salmeron, M., Diaz, A., Ortega, J., Prieto, A., Olivares, G. (2000). "Genetic algorithms and neuro-dynamic programming: application to water supply networks". Proceedings of 2000 Congress on Evolutionary Computation. 2000 Congress on Evolutionary Computation. Vol. 1. La Jolla, California, US: IEEE. pp. 7–14. doi:10.1109/CEC.2000.870269. ISBN 0-7803-6375-2.
• • Deng G, Ferris, M.C. (2008). "Neuro-dynamic programming for fractionated radiotherapy planning". Optimization in Medicine. Springer Optimization and Its Applications. Vol. 12. pp. 47–70. CiteSeerX 10.1.1.137.8288. doi:10.1007/978-0-387-73299-2_3. ISBN 978-0-387-73298-5.
• • Bozinovski, S. (1982). "A self-learning system using secondary reinforcement". In R. Trappl (ed.) Cybernetics and Systems Research: Proceedings of the Sixth European Meeting on Cybernetics and Systems Research. North Holland. pp. 397–402. ISBN 978-0-444-86488-8.
• • Bozinovski, S. (2014) "Modeling mechanisms of cognition-emotion interaction in artificial neural networks, since 1981 Archived 23 March 2019 at the Wayback Machine." Procedia Computer Science p. 255-263
• • Bozinovski S, Bozinovska L (2001). "Self-learning agents: A connectionist theory of emotion based on crossbar value judgment". Cybernetics and Systems. 32 (6): 637–667. doi:10.1080/01969720118145. S2CID 8944741.
• • Salimans T, Ho J, Chen X, Sidor S, Sutskever I (7 September 2017). "Evolution Strategies as a Scalable Alternative to Reinforcement Learning". arXiv:1703.03864 [stat.ML].
• • Such FP, Madhavan V, Conti E, Lehman J, Stanley KO, Clune J (20 April 2018). "Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning". arXiv:1712.06567 [cs.NE].
• • "Artificial intelligence can 'evolve' to solve problems". Science | AAAS. 10 January 2018. Archived from the original on 9 December 2021. Retrieved 7 February 2018.
• • Turchetti C (2004), Stochastic Models of Neural Networks, Frontiers in artificial intelligence and applications: Knowledge-based intelligent engineering systems, vol. 102, IOS Press, ISBN 978-1-58603-388-0
• • Jospin LV, Laga H, Boussaid F, Buntine W, Bennamoun M (2022). "Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users". IEEE Computational Intelligence Magazine. Vol. 17, no. 2. pp. 29–48. arXiv:2007.06823. doi:10.1109/mci.2022.3155327. ISSN 1556-603X. S2CID 220514248.
• • de Rigo, D., Castelletti, A., Rizzoli, A. E., Soncini-Sessa, R., Weber, E. (January 2005). "A selective improvement technique for fastening Neuro-Dynamic Programming in Water Resources Network Management". In Pavel Zítek (ed.). Proceedings of the 16th IFAC World Congress – IFAC-PapersOnLine. 16th IFAC World Congress. Vol. 16. Prague, Czech Republic: IFAC. pp. 7–12. doi:10.3182/20050703-6-CZ-1902.02172. hdl:11311/255236. ISBN 978-3-902661-75-3. Archived from the original on 26 April 2012. Retrieved 30 December 2011.
• • Ferreira C (2006). "Designing Neural Networks Using Gene Expression Programming". In A. Abraham, B. de Baets, M. Köppen, B. Nickolay (eds.). Applied Soft Computing Technologies: The Challenge of Complexity (PDF). Springer-Verlag. pp. 517–536. Archived (PDF) from the original on 19 December 2013. Retrieved 8 October 2012.
• • Da, Y., Xiurun, G. (July 2005). "An improved PSO-based ANN with simulated annealing technique". In T. Villmann (ed.). New Aspects in Neurocomputing: 11th European Symposium on Artificial Neural Networks. Vol. 63. Elsevier. pp. 527–533. doi:10.1016/j.neucom.2004.07.002. Archived from the original on 25 April 2012. Retrieved 30 December 2011.
• • Wu, J., Chen, E. (May 2009). "A Novel Nonparametric Regression Ensemble for Rainfall Forecasting Using Particle Swarm Optimization Technique Coupled with Artificial Neural Network". In Wang, H., Shen, Y., Huang, T., Zeng, Z. (eds.). 6th International Symposium on Neural Networks,

ISNN 2009. Lecture Notes in Computer Science. Vol. 5553. Springer. pp. 49–58. doi:10.1007/978-3-642-01513-7_6. ISBN 978-3-642-01215-0. Archived from the original on 31 December 2014. Retrieved 1 January 2012.
• • Ting Qin, Zonghai Chen, Haitao Zhang, Sifu Li, Wei Xiang, Ming Li (2004). "A learning algorithm of CMAC based on RLS" (PDF). Neural Processing Letters. 19 (1): 49–61. doi:10.1023/B:NEPL.0000016847.18175.60. S2CID 6233899. Archived (PDF) from the original on 14 April 2021. Retrieved 30 January 2019.
• • Ting Qin, Haitao Zhang, Zonghai Chen, Wei Xiang (2005). "Continuous CMAC-QRLS and its systolic array" (PDF). Neural Processing Letters. 22 (1): 1–16. doi:10.1007/s11063-004-2694-0. S2CID 16095286. Archived (PDF) from the original on 18 November 2018. Retrieved 30 January 2019.
• • LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". Neural Computation. 1 (4): 541–551. doi:10.1162/neco.1989.1.4.541. S2CID 41312633.
• • Yann LeCun (2016). Slides on Deep Learning Online Archived 23 April 2016 at the Wayback Machine
• • Hochreiter S, Schmidhuber J (1 November 1997). "Long Short-Term Memory". Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. ISSN 0899-7667. PMID 9377276. S2CID 1915014.
• • Sak H, Senior A, Beaufays F (2014). "Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling" (PDF). Archived from the original (PDF) on 24 April 2018.
• • Li X, Wu X (15 October 2014). "Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition". arXiv:1410.4281 [cs.CL].
• • Fan Y, Qian Y, Xie F, Soong FK (2014). "TTS synthesis with bidirectional LSTM based Recurrent Neural Networks". Proceedings of the Annual Conference of the International Speech Communication Association, Interspeech: 1964–1968. Retrieved 13 June 2017.
• • Schmidhuber J (2015). "Deep Learning". Scholarpedia. 10 (11): 85–117. Bibcode:2015SchpJ..1032832S. doi:10.4249/scholarpedia.32832.
• • Zen H, Sak H (2015). "Unidirectional Long Short-Term Memory Recurrent Neural Network with Recurrent Output Layer for Low-Latency Speech Synthesis" (PDF). Google.com. ICASSP. pp. 4470–4474. Archived (PDF) from the original on 9 May 2021. Retrieved 27 June 2017.
• • Fan B, Wang L, Soong FK, Xie L (2015). "Photo-Real Talking Head with Deep Bidirectional LSTM" (PDF). Proceedings of ICASSP. Archived (PDF) from the original on 1 November 2017. Retrieved 27 June 2017.
• • Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, et al. (5 December 2017). "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". arXiv:1712.01815 [cs.AI].
• • Probst P, Boulesteix AL, Bischl B (26 February 2018). "Tunability: Importance of Hyperparameters of Machine Learning Algorithms". J. Mach. Learn. Res. 20: 53:1–53:32. S2CID 88515435.
• • Zoph B, Le QV (4 November 2016). "Neural Architecture Search with Reinforcement Learning". arXiv:1611.01578 [cs.LG].
• • Haifeng Jin, Qingquan Song, Xia Hu (2019). "Auto-keras: An efficient neural architecture search system". Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM. arXiv:1806.10282. Archived from the original on 21 August 2019. Retrieved 21 August 2019 – via autokeras.com.
• • Claesen M, De Moor B (2015). "Hyperparameter Search in Machine Learning". arXiv:1502.02127 [cs.LG]. Bibcode:2015arXiv150202127C
• • Esch R (1990). "Functional Approximation". Handbook of Applied Mathematics (Springer US ed.). Boston, MA: Springer US. pp. 928–987. doi:10.1007/978-1-4684-1423-3_17. ISBN 978-1-4684-1423-3.
• • Sarstedt M, Moo E (2019). "Regression Analysis". A Concise Guide to Market Research. Springer Texts in Business and Economics. Springer Berlin Heidelberg. pp. 209–256. doi:10.1007/978-3-662-56707-4_7. ISBN 978-3-662-56706-7. S2CID 240396965. Archived from the original on 20 March 2023. Retrieved 20 March 2023.

• • Tian J, Tan Y, Sun C, Zeng J, Jin Y (December 2016). "A self-adaptive similarity-based fitness approximation for evolutionary optimization". 2016 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–8. doi:10.1109/SSCI.2016.7850209. ISBN 978-1-5090-4240-1. S2CID 14948018. Archived from the original on 19 May 2024. Retrieved 22 March 2023.
• • Alaloul WS, Qureshi AH (2019). "Data Processing Using Artificial Neural Networks". Dynamic Data Assimilation – Beating the Uncertainties. doi:10.5772/intechopen.91935. ISBN 978-1-83968-083-0. S2CID 219735060. Archived from the original on 20 March 2023. Retrieved 20 March 2023.
• • Pal M, Roy R, Basu J, Bepari MS (2013). "Blind source separation: A review and analysis". 2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE). IEEE. pp. 1–5. doi:10.1109/ICSDA.2013.6709849. ISBN 978-1-4799-2378-6. S2CID 37566823. Archived from the original on 20 March 2023. Retrieved 20 March 2023.
• • Zissis D (October 2015). "A cloud based architecture capable of perceiving and predicting multiple vessel behaviour". Applied Soft Computing. 35: 652–661. doi:10.1016/j.asoc.2015.07.002. Archived from the original on 26 July 2020. Retrieved 18 July 2019.
• • Sengupta N, Sahidullah, Md, Saha, Goutam (August 2016). "Lung sound classification using cepstral-based statistical features". Computers in Biology and Medicine. 75 (1): 118–129. doi:10.1016/j.compbiomed.2016.05.013. PMID 27286184.
• • Choy, Christopher B., et al. "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction Archived 26 July 2020 at the Wayback Machine." European conference on computer vision. Springer, Cham, 2016.
• • Turek, Fred D. (March 2007). "Introduction to Neural Net Machine Vision". Vision Systems Design. 12 (3). Archived from the original on 16 May 2013. Retrieved 5 March 2013.
• • Maitra DS, Bhattacharya U, Parui SK (August 2015). "CNN based common approach to handwritten character recognition of multiple scripts". 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 1021–1025. doi:10.1109/ICDAR.2015.7333916. ISBN 978-1-4799-1805-8. S2CID 25739012. Archived from the original on 16 October 2023. Retrieved 18 March 2021.
• • Gessler J (August 2021). "Sensor for food analysis applying impedance spectroscopy and artificial neural networks". RiuNet UPV (1): 8–12. Archived from the original on 21 October 2021. Retrieved 21 October 2021.
• • French J (2016). "The time traveller's CAPM". Investment Analysts Journal. 46 (2): 81–96. doi:10.1080/10293523.2016.1255469. S2CID 157962452.
• • Roman M. Balabin, Ekaterina I. Lomakina (2009). "Neural network approach to quantum-chemistry data: Accurate prediction of density functional theory energies". J. Chem. Phys. 131 (7): 074104. Bibcode:2009JChPh.131g4104B. doi:10.1063/1.3206326. PMID 19708729.
• • Silver D, et al. (2016). "Mastering the game of Go with deep neural networks and tree search" (PDF). Nature. 529 (7587): 484–489. Bibcode:2016Natur.529..484S. doi:10.1038/nature16961. PMID 26819042. S2CID 515925. Archived (PDF) from the original on 23 November 2018. Retrieved 31 January 2019.
• • Pasick A (27 March 2023). "Artificial Intelligence Glossary: Neural Networks and Other Terms Explained". The New York Times. ISSN 0362-4331. Archived from the original on 1 September 2023. Retrieved 22 April 2023.
• • Schechner S (15 June 2017). "Facebook Boosts A.I. to Block Terrorist Propaganda". The Wall Street Journal. ISSN 0099-9660. Archived from the original on 19 May 2024. Retrieved 16 June 2017.
• • Ciaramella A, Ciaramella M (2024). Introduction to Artificial Intelligence: from data analysis to generative AI. Intellisemantic Editions. ISBN 978-8-8947-8760-3.
• • Ganesan N (2010). "Application of Neural Networks in Diagnosing Cancer Disease Using Demographic Data". International Journal of Computer Applications. 1 (26): 81–97. Bibcode:2010IJCA....1z..81G. doi:10.5120/476-783.
• • Bottaci L (1997). "Artificial Neural Networks Applied to Outcome Prediction for Colorectal Cancer Patients in Separate Institutions" (PDF). Lancet. 350 (9076). The Lancet: 469–72. doi:10.1016/S0140-6736(96)11196-X. PMID 9274582. S2CID 18182063. Archived from the original (PDF) on 23 November 2018. Retrieved 2 May 2012.

• • Alizadeh E, Lyons SM, Castle JM, Prasad A (2016). "Measuring systematic changes in invasive cancer cell shape using Zernike moments". Integrative Biology. 8 (11): 1183–1193. doi:10.1039/C6IB00100A. PMID 27735002. Archived from the original on 19 May 2024. Retrieved 28 March 2017.
• • Lyons S (2016). "Changes in cell shape are correlated with metastatic potential in murine". Biology Open. 5 (3): 289–299. doi:10.1242/bio.013409. PMC 4810736. PMID 26873952.
• • Nabian MA, Meidani H (28 August 2017). "Deep Learning for Accelerated Reliability Analysis of Infrastructure Networks". Computer-Aided Civil and Infrastructure Engineering. 33 (6): 443–458. arXiv:1708.08551. Bibcode:2017arXiv170808551N. doi:10.1111/mice.12359. S2CID 36661983.
• • Nabian MA, Meidani H (2018). "Accelerating Stochastic Assessment of Post-Earthquake Transportation Network Connectivity via Machine-Learning-Based Surrogates". Transportation Research Board 97th Annual Meeting. Archived from the original on 9 March 2018. Retrieved 14 March 2018.
• • Díaz E, Brotons V, Tomás R (September 2018). "Use of artificial neural networks to predict 3-D elastic settlement of foundations on soils with inclined bedrock". Soils and Foundations. 58 (6): 1414–1422. Bibcode:2018SoFou..58.1414D. doi:10.1016/j.sandf.2018.08.001. hdl:10045/81208. ISSN 0038-0806.
• • Tayebiyan A, Mohammad TA, Ghazali AH, Mashohor S. "Artificial Neural Network for Modelling Rainfall-Runoff". Pertanika Journal of Science & Technology. 24 (2): 319–330. Archived from the original on 17 May 2023. Retrieved 17 May 2023.
• • Govindaraju RS (1 April 2000). "Artificial Neural Networks in Hydrology. I: Preliminary Concepts". Journal of Hydrologic Engineering. 5 (2): 115–123. doi:10.1061/(ASCE)1084-0699(2000)5:2(115).
• • Govindaraju RS (1 April 2000). "Artificial Neural Networks in Hydrology. II: Hydrologic Applications". Journal of Hydrologic Engineering. 5 (2): 124–137. doi:10.1061/(ASCE)1084-0699(2000)5:2(124).
• • Peres DJ, Iuppa C, Cavallaro L, Cancelliere A, Foti E (1 October 2015). "Significant wave height record extension by neural networks and reanalysis wind data". Ocean Modelling. 94: 128–140. Bibcode:2015OcMod..94..128P. doi:10.1016/j.ocemod.2015.08.002.
• • Dwarakish GS, Rakshith S, Natesan U (2013). "Review on Applications of Neural Network in Coastal Engineering". Artificial Intelligent Systems and Machine Learning. 5 (7): 324–331. Archived from the original on 15 August 2017. Retrieved 5 July 2017.
• • Ermini L, Catani F, Casagli N (1 March 2005). "Artificial Neural Networks applied to landslide susceptibility assessment". Geomorphology. Geomorphological hazard and human impact in mountain environments. 66 (1): 327–343. Bibcode:2005Geomo..66..327E. doi:10.1016/j.geomorph.2004.09.025.
• • Nix R, Zhang J (May 2017). "Classification of Android apps and malware using deep neural networks". 2017 International Joint Conference on Neural Networks (IJCNN). pp. 1871–1878. doi:10.1109/IJCNN.2017.7966078. ISBN 978-1-5090-6182-2. S2CID 8838479.
• • "Detecting Malicious URLs". The systems and networking group at UCSD. Archived from the original on 14 July 2019. Retrieved 15 February 2019.
• • Homayoun S, Ahmadzadeh M, Hashemi S, Dehghantanha A, Khayami R (2018), Dehghantanha A, Conti M, Dargahi T (eds.), "BoTShark: A Deep Learning Approach for Botnet Traffic Detection", Cyber Threat Intelligence, Advances in Information Security, vol. 70, Springer International Publishing, pp. 137–153, doi:10.1007/978-3-319-73951-9_7, ISBN 978-3-319-73951-9
• • Ghosh, Reilly (January 1994). "Credit card fraud detection with a neural-network". Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences HICSS-94. Vol. 3. pp. 621–630. doi:10.1109/HICSS.1994.323314. ISBN 978-0-8186-5090-1. S2CID 13260377.
• • Ananthaswamy A (19 April 2021). "Latest Neural Nets Solve World's Hardest Equations Faster Than Ever Before". Quanta Magazine. Archived from the original on 19 May 2024. Retrieved 12 May 2021.
• • "AI has cracked a key mathematical puzzle for understanding our world". MIT Technology Review. Archived from the original on 19 May 2024. Retrieved 19 November 2020.
• • "Caltech Open-Sources AI for Solving Partial Differential Equations". InfoQ. Archived from the

original on 25 January 2021. Retrieved 20 January 2021.
• • Nagy A (28 June 2019). "Variational Quantum Monte Carlo Method with a Neural-Network Ansatz for Open Quantum Systems". Physical Review Letters. 122 (25): 250501. arXiv:1902.09483. Bibcode:2019PhRvL.122y0501N. doi:10.1103/PhysRevLett.122.250501. PMID 31347886. S2CID 119074378.
• • Yoshioka N, Hamazaki R (28 June 2019). "Constructing neural stationary states for open quantum many-body systems". Physical Review B. 99 (21): 214306. arXiv:1902.07006. Bibcode:2019PhRvB..99u4306Y. doi:10.1103/PhysRevB.99.214306. S2CID 119470636.
• • Hartmann MJ, Carleo G (28 June 2019). "Neural-Network Approach to Dissipative Quantum Many-Body Dynamics". Physical Review Letters. 122 (25): 250502. arXiv:1902.05131. Bibcode:2019PhRvL.122y0502H. doi:10.1103/PhysRevLett.122.250502. PMID 31347862. S2CID 119357494.
• • Vicentini F, Biella A, Regnault N, Ciuti C (28 June 2019). "Variational Neural-Network Ansatz for Steady States in Open Quantum Systems". Physical Review Letters. 122 (25): 250503. arXiv:1902.10104. Bibcode:2019PhRvL.122y0503V. doi:10.1103/PhysRevLett.122.250503. PMID 31347877. S2CID 119504484.
• • Forrest MD (April 2015). "Simulation of alcohol action upon a detailed Purkinje neuron model and a simpler surrogate model that runs >400 times faster". BMC Neuroscience. 16 (27): 27. doi:10.1186/s12868-015-0162-6. PMC 4417229. PMID 25928094.
• • Wieczorek S, Filipiak D, Filipowska A (2018). "Semantic Image-Based Profiling of Users' Interests with Neural Networks". Studies on the Semantic Web. 36 (Emerging Topics in Semantic Technologies). doi:10.3233/978-1-61499-894-5-179. Archived from the original on 19 May 2024. Retrieved 20 January 2024.
• • Merchant A, Batzner S, Schoenholz SS, Aykol M, Cheon G, Cubuk ED (December 2023). "Scaling deep learning for materials discovery". Nature. 624 (7990): 80–85. Bibcode:2023Natur.624...80M. doi:10.1038/s41586-023-06735-9. ISSN 1476-4687. PMC 10700131. PMID 38030720.
• • Siegelmann H, Sontag E (1991). "Turing computability with neural nets" (PDF). Appl. Math. Lett. 4 (6): 77–80. doi:10.1016/0893-9659(91)90080-F. Archived (PDF) from the original on 19 May 2024. Retrieved 10 January 2017.
• • Bains S (3 November 1998). "Analog computer trumps Turing model". EE Times. Archived from the original on 11 May 2023. Retrieved 11 May 2023.
• • Balcázar J (July 1997). "Computational Power of Neural Networks: A Kolmogorov Complexity Characterization". IEEE Transactions on Information Theory. 43 (4): 1175–1183. CiteSeerX 10.1.1.411.7782. doi:10.1109/18.605580.
• • MacKay DJ (2003). Information Theory, Inference, and Learning Algorithms (PDF). Cambridge University Press. ISBN 978-0-521-64298-9. Archived (PDF) from the original on 19 October 2016. Retrieved 11 June 2016.
• • Cover T (1965). "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition" (PDF). IEEE Transactions on Electronic Computers. EC-14 (3). IEEE: 326–334. doi:10.1109/PGEC.1965.264137. Archived (PDF) from the original on 5 March 2016. Retrieved 10 March 2020.
• • Gerald F (2019). "Reproducibility and Experimental Design for Machine Learning on Audio and Multimedia Data". Proceedings of the 27th ACM International Conference on Multimedia. ACM. pp. 2709–2710. doi:10.1145/3343031.3350545. ISBN 978-1-4503-6889-6. S2CID 204837170.
• • "Stop tinkering, start measuring! Predictable experimental design of Neural Network experiments". The Tensorflow Meter. Archived from the original on 18 April 2022. Retrieved 10 March 2020.
• • Lee J, Xiao L, Schoenholz SS, Bahri Y, Novak R, Sohl-Dickstein J, et al. (2020). "Wide neural networks of any depth evolve as linear models under gradient descent". Journal of Statistical Mechanics: Theory and Experiment. 2020 (12): 124002. arXiv:1902.06720. Bibcode:2020JSMTE2020l4002L. doi:10.1088/1742-5468/abc62b. S2CID 62841516.
• • Arthur Jacot, Franck Gabriel, Clement Hongler (2018). Neural Tangent Kernel: Convergence and Generalization in Neural Networks (PDF). 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, Canada. Archived (PDF) from the original on 22 June 2022.

Retrieved 4 June 2022.
• • Xu ZJ, Zhang Y, Xiao Y (2019). "Training Behavior of Deep Neural Network in Frequency Domain". In Gedeon T, Wong K, Lee M (eds.). Neural Information Processing. Lecture Notes in Computer Science. Vol. 11953. Springer, Cham. pp. 264–274. arXiv:1807.01251. doi:10.1007/978-3-030-36708-4_22. ISBN 978-3-030-36707-7. S2CID 49562099.
• • Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, et al. (2019). "On the Spectral Bias of Neural Networks" (PDF). Proceedings of the 36th International Conference on Machine Learning. 97: 5301–5310. arXiv:1806.08734. Archived (PDF) from the original on 22 October 2022. Retrieved 4 June 2022.
• • Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, Zheng Ma (2020). "Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks". Communications in Computational Physics. 28 (5): 1746–1767. arXiv:1901.06523. Bibcode:2020CCoPh..28.1746X. doi:10.4208/cicp.OA-2020-0085. S2CID 58981616.
• • Tao Luo, Zheng Ma, Zhi-Qin John Xu, Yaoyu Zhang (2019). "Theory of the Frequency Principle for General Deep Neural Networks". arXiv:1906.09235 [cs.LG].
• • Xu ZJ, Zhou H (18 May 2021). "Deep Frequency Principle Towards Understanding Why Deeper Learning is Faster". Proceedings of the AAAI Conference on Artificial Intelligence. 35 (12): 10541–10550. arXiv:2007.14313. doi:10.1609/aaai.v35i12.17261. ISSN 2374-3468. S2CID 220831156. Archived from the original on 5 October 2021. Retrieved 5 October 2021.
• • Parisi GI, Kemker R, Part JL, Kanan C, Wermter S (1 May 2019). "Continual lifelong learning with neural networks: A review". Neural Networks. 113: 54–71. arXiv:1802.07569. doi:10.1016/j.neunet.2019.01.012. ISSN 0893-6080. PMID 30780045.
• • Dean Pomerleau, "Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving"
• • Dewdney AK (1 April 1997). Yes, we have no neutrons: an eye-opening tour through the twists and turns of bad science. Wiley. p. 82. ISBN 978-0-471-10806-1.
• • NASA – Dryden Flight Research Center – News Room: News Releases: NASA NEURAL NETWORK PROJECT PASSES MILESTONE Archived 2 April 2010 at the Wayback Machine. Nasa.gov. Retrieved on 20 November 2013.
• • "Roger Bridgman's defence of neural networks". Archived from the original on 19 March 2012. Retrieved 12 July 2010.
• • "Scaling Learning Algorithms towards {AI} – LISA – Publications – Aigaion 2.0". iro.umontreal.ca.
• • D. J. Felleman and D. C. Van Essen, "Distributed hierarchical processing in the primate cerebral cortex," Cerebral Cortex, 1, pp. 1–47, 1991.
• • J. Weng, "Natural and Artificial Intelligence: Introduction to Computational Brain-Mind Archived 19 May 2024 at the Wayback Machine," BMI Press, ISBN 978-0-9858757-2-5, 2012.
• • Edwards C (25 June 2015). "Growing pains for deep learning". Communications of the ACM. 58 (7): 14–16. doi:10.1145/2771283. S2CID 11026540.
• • "The Bitter Lesson". incompleteideas.net. Retrieved 7 August 2024.
• • Cade Metz (18 May 2016). "Google Built Its Very Own Chips to Power Its AI Bots". Wired. Archived from the original on 13 January 2018. Retrieved 5 March 2017.
• • "Scaling Learning Algorithms towards AI" (PDF). Archived (PDF) from the original on 12 August 2022. Retrieved 6 July 2022.
• • Tahmasebi, Hezarkhani (2012). "A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation". Computers & Geosciences. 42: 18–27. Bibcode:2012CG.....42...18T. doi:10.1016/j.cageo.2012.02.004. PMC 4268588. PMID 25540468.
• • Sun and Bookman, 1990
• • Norori N, Hu Q, Aellen FM, Faraci FD, Tzovara A (October 2021). "Addressing bias in big data and AI for health care: A call for open science". Patterns. 2 (10): 100347. doi:10.1016/j.patter.2021.100347. PMC 8515002. PMID 34693373.
• • Carina W (27 October 2022). "Failing at Face Value: The Effect of Biased Facial Recognition Technology on Racial Discrimination in Criminal Justice". Scientific and Social Research. 4 (10): 29–40. doi:10.26689/ssr.v4i10.4402. ISSN 2661-4332.

• • Chang X (13 September 2023). "Gender Bias in Hiring: An Analysis of the Impact of Amazon's Recruiting Algorithm". Advances in Economics, Management and Political Sciences. 23 (1): 134–140. doi:10.54254/2754-1169/23/20230367. ISSN 2754-1169. Archived from the original on 9 December 2023. Retrieved 9 December 2023.
• • Kortylewski A, Egger B, Schneider A, Gerig T, Morel-Forster A, Vetter T (June 2019). "Analyzing and Reducing the Damage of Dataset Bias to Face Recognition with Synthetic Data". 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (PDF). IEEE. pp. 2261–2268. doi:10.1109/cvprw.2019.00279. ISBN 978-1-7281-2506-0. S2CID 198183828. Archived (PDF) from the original on 19 May 2024. Retrieved 30 December 2023.
• • Huang Y (2009). "Advances in Artificial Neural Networks – Methodological Development and Application". Algorithms. 2 (3): 973–1007. doi:10.3390/algor2030973. ISSN 1999-4893.
• • Kariri E, Louati H, Louati A, Masmoudi F (2023). "Exploring the Advancements and Future Research Directions of Artificial Neural Networks: A Text Mining Approach". Applied Sciences. 13 (5): 3186. doi:10.3390/app13053186. ISSN 2076-3417.
• • Fui-Hoon Nah F, Zheng R, Cai J, Siau K, Chen L (3 July 2023). "Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration". Journal of Information Technology Case and Application Research. 25 (3): 277–304. doi:10.1080/15228053.2023.2233814. ISSN 1522-8053.
• • "DALL-E 2's Failures Are the Most Interesting Thing About It – IEEE Spectrum". IEEE. Archived from the original on 15 July 2022. Retrieved 9 December 2023.
• • Briot JP (January 2021). "From artificial neural networks to deep learning for music generation: history, concepts and trends". Neural Computing and Applications. 33 (1): 39–65. doi:10.1007/s00521-020-05399-0. ISSN 0941-0643.
• • Chow PS (6 July 2020). "Ghost in the (Hollywood) machine: Emergent applications of artificial intelligence in the film industry". NECSUS_European Journal of Media Studies. doi:10.25969/MEDIAREP/14307. ISSN 2213-0217.
249. • Yu X, He S, Gao Y, Yang J, Sha L, Zhang Y, et al. (June 2010). "Dynamic difficulty adjustment of game AI for video game Dead-End". The 3rd International Conference on Information Sciences and Interaction Sciences. IEEE. pp. 583–587. doi:10.1109/icicis.2010.5534761. ISBN 978-1-4244-7384-7. S2CID 17555595.
Bibliography
• Bhadeshia H. K. D. H. (1999). "Neural Networks in Materials Science" (PDF). ISIJ International. 39 (10): 966–979. doi:10.2355/isijinternational.39.966.
• Bishop CM (1995). Neural networks for pattern recognition. Clarendon Press. ISBN 978-0-19-853849-3. OCLC 33101074.
• Borgelt C (2003). Neuro-Fuzzy-Systeme: von den Grundlagen künstlicher Neuronaler Netze zur Kopplung mit Fuzzy-Systemen. Vieweg. ISBN 978-3-528-25265-6. OCLC 76538146.
• Cybenko G (2006). "Approximation by Superpositions of a Sigmoidal function". In van Schuppen JH (ed.). Mathematics of Control, Signals, and Systems. Springer International. pp. 303–314. PDF
• Dewdney AK (1997). Yes, we have no neutrons: an eye-opening tour through the twists and turns of bad science. New York: Wiley. ISBN 978-0-471-10806-1. OCLC 35558945.
• Duda RO, Hart PE, Stork DG (2001). Pattern classification (2 ed.). Wiley. ISBN 978-0-471-05669-0. OCLC 41347061.
• Egmont-Petersen M, de Ridder D, Handels H (2002). "Image processing with neural networks – a review". Pattern Recognition. 35 (10): 2279–2301. CiteSeerX 10.1.1.21.5444. doi:10.1016/S0031-3203(01)00178-9.
• Fahlman S, Lebiere C (1991). "The Cascade-Correlation Learning Architecture" (PDF). Archived from the original (PDF) on 3 May 2013. Retrieved 28 August 2006.
o created for National Science Foundation, Contract Number EET-8716324, and Defense Advanced Research Projects Agency (DOD), ARPA Order No. 4976 under Contract F33615-87-C-1499.
• Gurney K (1997). An introduction to neural networks. UCL Press. ISBN 978-1-85728-673-1. OCLC 37875698.
• Haykin SS (1999). Neural networks: a comprehensive foundation. Prentice Hall. ISBN 978-0-13-273350-2. OCLC 38908586.
• Hertz J, Palmer RG, Krogh AS (1991). Introduction to the theory of neural computation. Addison-

Wesley. ISBN 978-0-201-51560-2. OCLC 21522159.
• Information theory, inference, and learning algorithms. Cambridge University Press. 25 September 2003. Bibcode:2003itil.book.....M. ISBN 978-0-521-64298-9. OCLC 52377690.
• Kruse R, Borgelt C, Klawonn F, Moewes C, Steinbrecher M, Held P (2013). Computational intelligence: a methodological introduction. Springer. ISBN 978-1-4471-5012-1. OCLC 837524179.
• Lawrence J (1994). Introduction to neural networks: design, theory and applications. California Scientific Software. ISBN 978-1-883157-00-5. OCLC 32179420.
• Masters T (1994). Signal and image processing with neural networks: a C++ sourcebook. J. Wiley. ISBN 978-0-471-04963-0. OCLC 29877717.
• Maurer H (2021). Cognitive science: integrative synchronization mechanisms in cognitive neuroarchitectures of the modern connectionism. CRC Press. doi:10.1201/9781351043526. ISBN 978-1-351-04352-6. S2CID 242963768.
• Ripley BD (2007). Pattern Recognition and Neural Networks. Cambridge University Press. ISBN 978-0-521-71770-0.
• Siegelmann H, Sontag ED (1994). "Analog computation via neural networks". Theoretical Computer Science. 131 (2): 331–360. doi:10.1016/0304-3975(94)90178-3. S2CID 2456483.
• Smith M (1993). Neural networks for statistical modeling. Van Nostrand Reinhold. ISBN 978-0-442-01310-3. OCLC 27145760.
• Wasserman PD (1993). Advanced methods in neural computing. Van Nostrand Reinhold. ISBN 978-0-442-00461-3. OCLC 27429729.
• Wilson H (2018). Artificial intelligence. Grey House Publishing. ISBN 978-1-68217-867-6.
External links
Listen to this article (31 minutes)
Duration: 31 minutes and 23 seconds.31:23

This audio file was created from a revision of this article dated 27 November 2011, and does not reflect subsequent edits.
(Audio help • More spoken articles)
• A Brief Introduction to Neural Networks (D. Kriesel) – Illustrated, bilingual manuscript about artificial neural networks; Topics so far: Perceptrons, Backpropagation, Radial Basis Functions, Recurrent Neural Networks, Self Organizing Maps, Hopfield Networks.
• Review of Neural Networks in Materials Science Archived 7 June 2015 at the Wayback Machine
• Artificial Neural Networks Tutorial in three languages (Univ. Politécnica de Madrid)
• Another introduction to ANN
• Next Generation of Neural Networks Archived 24 January 2011 at the Wayback Machine – Google Tech Talks
• Performance of Neural Networks
• Neural Networks and Information Archived 9 July 2009 at the Wayback Machine
• Sanderson G (5 October 2017). "But what is a Neural Network?". 3Blue1Brown. Archived from the original on 7 November 2021 – via YouTube.

• v
• t
• e
Artificial intelligence

Concepts • Parameter
o Hyperparameter
• Loss functions
• Regression
o Bias–variance tradeoff
o Double descent
o Overfitting
• Clustering

• Gradient descent
o SGD
o Quasi-Newton method
o Conjugate gradient method
• Backpropagation
• Attention
• Convolution
• Normalization
o Batchnorm
• Activation
o Softmax
o Sigmoid
o Rectifier
• Gating
• Weight initialization
• Regularization
• Datasets
o Augmentation
• Reinforcement learning
o Q-learning
o SARSA
o Imitation
• Diffusion
• Latent diffusion model
• Autoregression
• Adversary
• RAG
• RLHF
• Self-supervised learning
• Prompt engineering
• Word embedding
• Hallucination

Applications • Machine learning
o In-context learning
• Artificial neural network
o Deep learning
• Language model
o Large language model
o NMT
• Artificial general intelligence

Implementations Audio–visual • AlexNet
• WaveNet
• Human image synthesis
• HWR
• OCR
• Speech synthesis
o ElevenLabs
• Speech recognition
o Whisper
• Facial recognition
• AlphaFold
• Text-to-image models

o DALL-E
o Flux
o Ideogram
o Midjourney
o Stable Diffusion
• Text-to-video models
o Sora
o Dream Machine
o VideoPoet
• Music generation
o Suno AI
o Udio

Text • Word2vec
• Seq2seq
• GloVe
• BERT
• T5
• Llama
• Chinchilla AI
• PaLM
• GPT
o 1
o 2
o 3
o J
o ChatGPT
o 4
o 4o
o o1
• Claude
• Gemini
• Grok
• LaMDA
• BLOOM
• Project Debater
• IBM Watson
• IBM Watsonx
• Granite
• PanGu-Σ

Decisional • AlphaGo
• AlphaZero
• OpenAI Five
• Self-driving car
• MuZero
• Action selection
o AutoGPT
• Robot control

People • Alan Turing
• Warren Sturgis McCulloch
• Walter Pitts

• John von Neumann
• Claude Shannon
• Marvin Minsky
• John McCarthy
• Nathaniel Rochester
• Allen Newell
• Cliff Shaw
• Herbert A. Simon
• Oliver Selfridge
• Frank Rosenblatt
• Bernard Widrow
• Joseph Weizenbaum
• Seymour Papert
• Seppo Linnainmaa
• Paul Werbos
• Jürgen Schmidhuber
• Yann LeCun
• Geoffrey Hinton
• John Hopfield
• Yoshua Bengio
• Lotfi A. Zadeh
• Stephen Grossberg
• Alex Graves
• Andrew Ng
• Fei-Fei Li
• Alex Krizhevsky
• Ilya Sutskever
• Demis Hassabis
• David Silver
• Ian Goodfellow
• Andrej Karpathy

Organizations • Anthropic
• EleutherAI
• Google DeepMind
• Hugging Face
• Kuaishou
• Meta AI
• Mila
• MiniMax
• Mistral AI
• MIT CSAIL
• OpenAI
• Runway
• Stability AI
• xAI

Architectures • Neural Turing machine
• Differentiable neural computer
• Transformer
o Vision transformer (ViT)
• Recurrent neural network (RNN)
• Long short-term memory (LSTM)
• Gated recurrent unit (GRU)

• Echo state network
• Multilayer perceptron (MLP)
• Convolutional neural network (CNN)
• Residual neural network (RNN)
• Highway network
• Mamba
• Autoencoder
• Variational autoencoder (VAE)
• Generative adversarial network (GAN)
• Graph neural network (GNN)

• Portals
o Technology
• Categories
o Artificial neural networks
o Machine learning

• v
• t
• e
Complex systems

• v
• t
• e
Control theory

• v
• t
• e
Neuroscience

• v
• t
• e
Self-driving cars, self-driving vehicles and enabling technologies
Authority control databases: National
• Germany
• United States
• Japan
• Czech Republic
• Latvia
• Israel

Categories:
• Computational statistics
• Artificial neural networks
• Classification algorithms
• Computational neuroscience
• Market research

- Mathematical psychology
- Mathematical and quantitative methods (economics)
- Bioinspiration
- This page was last edited on 25 November 2024, at 20:43 (UTC).

Neural computation

- Article
- Talk
- Read
- Edit
- View history

Tools

-
-
-
-
-
-
-
-
-
-
-
-

Appearance
Text

-

Small
Standard
Large
Width

-

Standard
Wide
Color (beta)

-

Automatic
Light

Dark
From Wikipedia, the free encyclopedia

This article needs additional citations for verification. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.
Find sources: "Neural computation" – news • newspapers • books • scholar • JSTOR (June 2019) (Learn how and when to remove this message)

Neural computation is the information processing performed by networks of neurons. Neural computation is affiliated with the philosophical tradition known as Computational theory of mind, also referred to as computationalism, which advances the thesis that neural computation explains cognition. The first persons to propose an account of neural activity as being computational was Warren McCullock and Walter Pitts in their seminal 1943 paper, A Logical Calculus of the Ideas Immanent in Nervous Activity.
There are three general branches of computationalism, including classicism, connectionism, and computational neuroscience. All three branches agree that cognition is computation, however, they disagree on what sorts of computations constitute cognition. The classicism tradition believes that computation in the brain is digital, analogous to digital computing. Both connectionism and computational neuroscience do not require that the computations that realize cognition are necessarily digital computations. However, the two branches greatly disagree upon which sorts of experimental data should be used to construct explanatory models of cognitive phenomena. Connectionists rely upon behavioral evidence to construct models to explain cognitive phenomena, whereas computational neuroscience leverages neuroanatomical and neurophysiological information to construct mathematical models that explain cognition.[1]
When comparing the three main traditions of the computational theory of mind, as well as the different possible forms of computation in the brain, it is helpful to define what we mean by computation in a general sense. Computation is the processing of information, otherwise known as variables or entities, according to a set of rules. A rule in this sense is simply an instruction for executing a manipulation on the current state of the variable, in order to produce a specified output. In other words, a rule dictates which output to produce given a certain input to the computing system. A computing system is a mechanism whose components must be functionally organized to process the information in accordance with the established set of rules. The types of information processed by a computing system determine which type of computations it performs. Traditionally, in cognitive science there have been two proposed types of computation related to neural activity - digital and analog, with the vast majority of theoretical work incorporating a digital understanding of cognition. Computing systems that perform digital computation are functionally organized to execute operations on strings of digits with respect to the type and location of the digit on the string. It has been argued that neural spike train signaling implements some form of digital computation, since neural spikes may be considered as discrete units or digits, like 0 or 1 - the neuron either fires an action potential or it does not. Accordingly, neural spike trains could be seen as strings of digits. Alternatively, analog computing systems perform manipulations on non-discrete, irreducibly continuous variables, that is, entities that vary continuously as a function of time. These sorts of operations are characterized by systems of differential equations.[1]
Neural computation can be studied for example by building models of neural computation.
There is a scientific journal dedicated to this subject, Neural Computation.
Artificial neural networks (ANN) is a subfield of the research area machine learning. Work on ANNs has been somewhat inspired by knowledge of neural computation.[1]
References
1.
Piccinini, Gualtiero; Bahar, Sonya (2013). "Neural Computation and the Computational Theory of Cognition". Cognitive Science. 37 (3): 453–488. doi:10.1111/cogs.12012. PMID 23126542.

MIT Press books may be purchased at special quantity discounts for business or sales promotional use.
For information, please email special_sales@mitpress.mit.edu or write to Special Sales Department, The
MIT Press, 55 Hayward Street, Cambridge, MA 02142.
This book was set in Adobe InDesign CS3 by John P. Frisby. Printed and bound in the United States of
America.
Library of Congress Cataloging-in-Publication Data
Seeing: the computational approach to biological vision / John P. Frisby and James V. Stone—2nd ed.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-262-51427-9 (pbk.: alk. paper)
1. Visual perception. 2. Optical illusions. 3. Psychophysiology. I. Stone, James V. II. Title.
BF241.F74 2010
152.14—dc22 2009031989
1.1 An "inner screen" theory of seeing
One theory of this kind proposes that there is a set of brain cells whose level of activity represents the brightness of points
in the scene. This theory therefore suggests that seeing is akin to photography. Note that the image of Lennon is inverted in
the eye, due to the optics of the eye, but it is shown upright in the brain to match our perceptions of the world—see page 8.
Lennon photograph courtesy Associated Newspapers Archive.
1
Seeing: What Is It?
Observed scene: a photograph of John Lennon
Retinal image of the scene
focused upside-down and
left-right reversed on to the
light-sensitive retina of the eye

1.5 Diagrammatic section through the head
This shows principal features of the major visual pathway
that links the eyes to the cortex.
world, and how is it obtained? It is this problem
which provides the subject of this book.
Perception, Consciousness, and Brain Cells
One reason why it might feel strange to regard
visual experience as being encoded in brain cells is
that they may seem quite insufficient for the task.
The "inner screen" theory posits a direct relation-
ship between conscious visual experiences and
activity in certain brain cells. That is, activity in

certain cells is somehow accompanied by conscious experience. Proposing this kind of parallelism between brain-cell activity and visual experience is characteristic of many theories of perceptual brain mechanisms. But is there more to it than this? Can the richness of visual experience really be identified with activity in a few million, or even a few trillion, brain cells? Are brain cells the right kind of entities to provide conscious perceptual experience? We return to these questions in Ch 22. For the moment, we simply note that most vision scientists get on with the job of studying seeing without concerning themselves much with the issue of consciousness.

Pictures in the Brain

You might reasonably ask at this point: has neuroscience has anything to say directly about the "inner screen" theory? Is there any evidence from studies of the brain as to whether such a screen or anything like it exists?

The major visual pathway carrying the messages from the eyes to the brain is shown in broad outline in 1.5. Fuller details are shown in 1.6 in which the eyes are shown inspecting a person, and the locations of the various parts of this scene "in" the visual system are shown with the help of numbers.

The first thing to notice is that the eyes do not receive identical images. The left eye sees rather more of the scene to the left of the central line of sight (regions 1 and 2), and vice versa for the right eye (regions 8 and 9). There are other differences between the left and right eyes' images in the case of 3D scenes: these are described fully in Ch 18.

Next, notice the optic nerves leaving the eyes. The fibers within each optic nerve are the axons of certain retinal cells, and they carry messages from the retina to the brain. The left and right optic nerves meet at the optic chiasm, 1.6 and 9.9, where the optic nerve bundle from each eye splits in two. Half of the fibers from each eye cross to the opposite side of the brain, whereas the other half stay on the same side of the brain throughout.

The net result of this partial crossing-over of fibers (technically called partial decussation) is that messages dealing with any given region of the field of view arrive at a common destination in the cortex, regardless of which eye they come from. In other words, left- and right-eye views of any given feature of a scene are analyzed in the same physical location in the striate cortex. This is the major receiving area in the cortex for messages sent along nerve fibers in the optic nerves.

Fibers from the optic chiasm enter the left and right lateral geniculate nuclei. These are the first

“relay stations” of the fibers from the eyes on their way to the striate cortex. That is, axons from the retina terminate here on the dendrites of new neurons, and it is the axons of the latter cells that then proceed to the cortex. A good deal of mystery still

Striate
cortex
Optic radiations
Eye
Cortex
Optic chiasm
Optic nerve
Retina
Lateral geniculate nucleus

Chapter 1
6

brain structure which lies underneath the cortex, 1.5, so it is said to be sub-cortical. Its function is different from that performed by regions of the cortex devoted to vision. The weight of evidence at present suggests that the superior colliculus is concerned with guiding visual attention. For example, if an object suddenly appears in the field of view, mechanisms within the superior colliculus detect its presence, work out its location, and then guide eye movements so that the novel object can be observed directly.

It is important to realize that other visual pathways exist apart from the two main ones shown in 1.6. In fact, in monkeys and most probably also in man, optic nerve fibers directly feed at least six different brain sites. This is testimony to the enormously important role of vision for ourselves and similar species. Indeed, it has been estimated that roughly 60% of the brain is involved in vision in one way or another.

Returning now to the issue of pictures-in-the-brain, the striking thing in 1.6 is the orderly, albeit curious, layout of fiber terminations in the striate cortex.

First, note that a face is shown mapped out on the cortical surface (cortical means “of the cortex”). This is the face that the eyes are inspecting.

Second, the representation is upside-down. The retinal images (not shown in 1.6) are also upside-down due to the way the optics of the eyes work, 1.1. Notice that the sketch of the “inner screen” in 1.1 showed a right-way-up image, so it is different in that respect from the mapping found in the striate cortex.

Third, the mapping is such that the representation of the scene is cut right down the middle, and each half of the cortex (technically, each cerebral hemisphere) deals initially with just one half of the

scene.

Fourth, and perhaps most oddly, the cut in the representation places adjacent regions of the central part of the scene farthest apart in the brain!

Fifth, the mapping is spatially distorted in that a greater area of cortex is devoted to central vision than to peripheral: hence the relatively swollen face and the diminutive arm and hand, 1.7. This doesn't mean of course that we actually see people in this distorted way—obviously we don't. But it

1.7 Mapping of the retinal image in the striate cortex (schematic)

Turn the book upside-down for a better appreciation of the distortion of the scene in cortex. The hyperfields are regions of
the retinal image that project to hypothetical structures called hypercolumns (denoted as graph-paper squares in the part
of the striate cortex map shown here, which derives from the left hand sides of the left and right retinal images; more details
in Chs 9 and10). Hyperfields are much smaller in central than in peripheral vision, so relatively more cells are devoted to
central vision. Hyperfields have receptive fields in both images but here two are shown for the right image only.

Striate cortex of left
cerebral hemisphere
Scene Retinal image in right eye
Hyperfields in
right image
4
4
1
6

reveals that a much larger area in our brain is assigned to foveal vision (i.e., analyzing what we are directly looking at) than is devoted to peripheral vision. This dedication of most cortical tissue to foveal vision is why we are much better at seeing details in the region of the scene we are looking at than we are at seeing details which fall out toward the edges of our field of view.

All in all, the cortical mapping of incoming visual fibers is curious but orderly. That is, adjacent regions of cortex deal with adjacent regions of the scene (with the exception of the mid-line cut). The technical term for this sort of mapping is topographic. In this instance it is called retinotopic as the mapping preserves the neighborhood relationships that exist between cells in the retina (except for the split down the middle). The general orderliness of the mapping is reminiscent of the "inner screen" proposed in 1.1. But the oddities of the mapping should give any "inner screen" theorist pause for thought. The first "screen," if such it is, we meet in the brain is a very strange one indeed.

The striate cortex is not the only region of cortex to be concerned with vision—far from it. Fibers travel from the striate cortex to adjacent regions, called the pre-striate cortex because they lie just in front of the striate region. These fibers preserve the orderliness of the mapping found in the striate region. There are in fact topographically organized visual regions in the pre-striate zone and we describe these maps in Ch 10. For the present, we just note that each one seems to be specialized for a particular kind of visual analysis, such as color, motion, etc. One big mystery is how the visual world can appear to us as such a well-integrated whole if its analysis is actually conducted at very many different sites, each one serving a different analytic function.

To summarize this section, brain maps exist which bear some resemblance to the kind of "inner screen" idea hesitantly advanced by our fictional "ordinary person" who was pressed to hazard a guess at what goes on the brain when we see. However, the map shown in 1.6-1.7 is not much like the one envisaged in 1.1, being both distorted, upside-down and cut into two.

These oddities seriously undermine the photographic metaphor for seeing. But it is timely to change tack now from looking inside the brain for an "inner screen" and to examine in detail serious logical problems with the "inner screen" idea as a theory of seeing. We begin this task by considering man-made systems for seeing.

Machines for Seeing

A great deal of research has been done on building computer vision systems that can do visual tasks. These take in images of a scene as input, analyze the visual information in these images, and then use that information for some purpose or other, such as guiding a robot hand or stating what objects the scene contains and where they are. In our terminology, a machine of this type is deriving a scene description from input images.

Whether one should call such a device a "perceiver," a "seeing machine," or more humbly an "image processor" or "pattern recognizer," is a moot point which may hinge on whether consciousness can ever be associated with non-biological brains. In any event, scientists who work on the problem of devising automatic image-processing machines would call the activity appearing on the "inner screen" of 1.1 a kind of gray level description of the painting. This is because the "inner screen" is a representation signaling the various shades of gray all over the picture, 1.8. (We ignore color in the present discussion, and also many

intricacies in the perception of gray: see Ch 16). Each individual brain cell in the screen is describing the gray level at one particular point of the picture in terms of an activity code. The code is simple: the lighter or more brightly illuminated the point in the painting, the more active the cell.

The familiar desktop image scanner is an example of a human-made device that delivers gray level descriptions. Its optical sensor sweeps over the image laid face down on its glass surface, thereby measuring gray levels directly rather than from a lens-focused image. Their scanning is technically described as a serial operation as it deals with different regions of the image in sequence.

Digital cameras measure the point by point intensities of images focused on their light sensitive surfaces, so in this regard they are similar to biological eyes. They are said to operate in parallel because they take their intensity measurements everywhere over the image at the same instant. Hence they can deliver their gray levels quickly.

The intensity measurements taken by both scanners and digital cameras are recorded as numbers stored in a digital memory. To call this collection of numbers a “gray level description” is apt because this is exactly what the numbers are providing, as in 1.8.

The term “gray level” arises from the black-and-white nature of the system, with black being regarded as a very dark gray (and recorded with a small number) and white as a very light gray (and recorded with a large number).

The numbers are a description in the sense defined earlier: they make explicit the gray levels in the input image. That is, they make these gray levels immediately usable (which means there is no need for further processing to recover them) by subsequent stages of image processing.

Retinal images are upside-down, due to the optics of the eyes (Ch 2) and many people are worried by this. “Why doesn’t the world therefore appear upside down?”, they ask.

The answer is simple: as long as there is a systematic correspondence between the outside scene and the retinal image, the processes of image interpretation can rely on this correspondence, and build up the required scene description accordingly. Upside-down in the image is simply interpreted as right-way-up in the world, and that’s all there is to it.

If an observer is equipped with special spectacles which optically invert the retinal images so that

they become the "right-way-up," then the world appears upside-down until the observer learns to cope with the new correspondence between image and scene. This adjustment process can take weeks, but it is possible. The exact nature of the adjustment process is not yet clear: does the upside-down world really begin to "look" right-way-up again, or is it simply that the observer learns new patterns of adjusted movement to cope with the strange new perceptual world he finds himself in?

1.8 Gray level description for a small region of an image of Lennon

Input image

Spectacle lens region enlarged to reveal individual pixels as squares with different gray levels

A sample of pixels from the upper left section of the spectacle region picked out above. This shows the pixel intensities both as different shades of gray and as the numbers stored in the gray level description in the computer's memory.

An ordinary domestic black-and-white TV set also produces an image that is an array of dots. The individual dots are so tiny that they cannot be readily distinguished (unless a TV screen is observed from quite close).

Representations and Descriptions

It is easy to see why the computer's gray level description illustrated in 1.8 is a similar sort of representation to the hypothetical "inner screen" shown in 1.1. In the latter, brain cells adopt different levels of activity to represent (or code) different pixel intensities. In the former, the computer holds different numbers in its memory registers to do exactly the same job. So both systems provide a representation of the gray level description of their input image, even though the physical stuff carrying this description (computer hardware vs. brainware) is different in the two cases. This distinction between the functional or design status of a representation (the job it performs) and the physical embodiment of the representation (different in man or machine of course) is an extremely important one which deserves further elaboration.

Consider, for example, the physical layout of the hypothetical "inner screen" of brain cells. This is an anatomically neat one, with the various pixel cells arranged in a format which physically matches the arrangement of the corresponding image points.

Gray Level Resolution

The number of pixels (shorthand for picture elements) in a computer's gray level description varies according to the capabilities of the computer (e.g., the size of its memory) and the needs of the user. For example, a dense array of pixels requires a large memory and produces a gray level description that picks up very fine details. This is now familiar to many people due to the availability of digital cameras that capture high resolution images using millions of pixels. When these are output as full-tone printouts, the images are difficult to discriminate from film-based photographs.

If fewer pixels are used, so that each pixel represents the average intensity over quite a large area of the input image, then a full-tone printout of the same size takes on a block-like appearance. That is, these images are said to show quantization effects. These possibilities are illustrated in 1.9, where the same input image is represented by four different gray level images, with pixel arrays ranging from high to low resolution.

of their
even
ph
ca
d
(
of repr
scree
ado
se
th
n
e
h

Try squinting to blur your vision while looking at the "block portrait" versions. You will find that Lennon magically appears more visible. See pages 128–131.

1.9 Gray level images
The images differ in pixel size from small to large.

io-inspired computer vision: Towards a synergistic approach of artificial and biological vision
Author links open overlay panelN. V. Kartheek Medathati a

, Heiko Neumann b, Guillaume S. Masson c, Pierre Kornprobst aShow moreAdd to Mendeley
Share
Cite
https://doi.org/10.1016/j.cviu.2016.04.009

Highlights
• •
We present an overview of computational approaches to biological vision.
• •
We develop a task centered presentation of biological vision studies.
• •
We revisit three tasks: image sensing, scene segmentation and optical flow.
• •
We show how new computer vision methods could be developed from biological insights.
• •
We identify key task specific biological vision models which could be scaled up.
Abstract
Studies in biological vision have always been a great source of inspiration for design of computer vision algorithms. In the past, several successful methods were designed with varying degrees of correspondence with biological vision studies, ranging from purely functional inspiration to methods that utilise models that were primarily developed for explaining biological observations. Even though it seems well recognised that computational models of biological vision can help in design of computer vision algorithms, it is a non-trivial exercise for a computer vision researcher to mine relevant information from biological vision literature as very few studies in biology are organised at a task level. In this paper we aim to bridge this gap by providing a computer vision task centric presentation of models primarily originating in biological vision studies. Not only do we revisit some of the main features of biological vision and discuss the foundations of existing computational studies modelling biological vision, but also we consider three classical computer vision tasks from a biological perspective: image sensing, segmentation and optical flow. Using this task-centric approach, we discuss well-known biological functional principles and compare them with approaches taken by computer vision. Based on this comparative analysis of computer and biological vision, we present some recent models in biological vision and highlight a few models that we think are promising for future investigations in computer vision. To this extent, this paper provides new insights and a starting point for investigators interested in the design of biology-based computer vision algorithms and pave a way for much needed interaction between the two communities leading to the development of synergistic models of artificial and biological vision.
Graphical abstract

1. Download: Download high-res image (244KB)
2. Download: Download full-size image
•
• Previous article in issue
• Next article in issue
Keywords
Canonical computations
Event based processing
Dynamic sensors
Multiplexed representation
Population coding
Soft selectivity
Feedback

Lateral interactions
Form-motion interactions
1. Introduction

Biological vision systems are remarkable at extracting and analysing the essential information for vital functional needs such as navigating through complex environments, finding food or escaping from a danger. It is remarkable that biological visual systems perform all these tasks with both high sensitivity and strong reliability given the fact that natural images are highly noisy, cluttered, highly variable and ambiguous. Still, even simple biological systems can efficiently and quickly solve most of the difficult computational problems that are still challenging for artificial systems such as scene segmentation, local and global optical flow computation, 3D perception or extracting the meaning of complex objects or movements. All these aspects have been intensively investigated in human psychophysics and the neuronal underpinnings of visual performance have been scrutinised over a wide range of temporal and spatial scales, from single cell to large cortical networks so that visual systems are certainly the best-known of all neural systems (see Chalupa and Werner, 2004 for an encyclopaedic review). As a consequence, biological visual computations are certainly the most understood of all cognitive neural systems.

It would seem natural that biological and computer vision research would interact continuously since they target the same goals at task leve

Contents

Further reading

Knowledge-based systems

- **Article**
- **Talk**
- **Read**
- **Edit**
- **View history**

Tools

-
-
-
-
-
-
-
-
-
-
-
-
-

Appearance
Text

-

Small
Standard
Large
Width

-

Standard
Wide
Color (beta)

-

Automatic
Light
Dark
From Wikipedia, the free encyclopedia

This article has multiple issues. Please help improve it or discuss these issues on the talk page. (Learn how and when to remove these messages)
This article needs additional citations for verification. (May 2024)
This article needs to be updated. (May 2024)

For the academic journal, see Knowledge-Based Systems (journal).
A knowledge-based system (KBS) is a computer program that reasons and uses a knowledge base to solve complex problems. Knowledge-based systems were the focus of early artificial intelligence researchers in the 1980s. The term can refer to a broad range of systems. However, all knowledge-based systems have two defining components: an attempt to represent knowledge explicitly, called a knowledge base, and a reasoning system that allows them to derive new knowledge, known as an inference engine.
Components
The knowledge base contains domain-specific facts and rules[1] about a problem domain (rather

than knowledge implicitly embedded in procedural code, as in a conventional computer program). In addition, the knowledge may be structured by means of a subsumption ontology, frames, conceptual graph, or logical assertions.[2]
The inference engine uses general-purpose reasoning methods to infer new knowledge and to solve problems in the problem domain. Most commonly, it employs forward chaining or backward chaining. Other approaches include the use of automated theorem proving, logic programming, blackboard systems, and term rewriting systems such as Constraint Handling Rules (CHR). These more formal approaches are covered in detail in the Wikipedia article on knowledge representation and reasoning.
Aspects and development of early systems
Knowledge-based vs. expert systems
See also: Expert system
The term "knowledge-based system" was often used interchangeably with "expert system", possibly because almost all of the earliest knowledge-based systems were designed for expert tasks. However, these terms tell us about different aspects of a system:
• expert: describes only the task the system is designed for – its purpose is to aid replace a human expert in a task typically requiring specialised knowledge
• knowledge-based: refers only to the system's architecture – it represents knowledge explicitly, rather than as procedural code
Today, virtually all expert systems are knowledge-based, whereas knowledge-based system architecture is used in a wide range of types of system designed for a variety of tasks.
Rule-based systems
Main article: Rule-based system
The first knowledge-based systems were primarily rule-based expert systems. These represented facts about the world as simple assertions in a flat database and used domain-specific rules to reason about these assertions, and then to add to them. One of the most famous of these early systems was Mycin, a program for medical diagnosis.
Representing knowledge explicitly via rules had several advantages:
1. Acquisition and maintenance. Using rules meant that domain experts could often define and maintain the rules themselves rather than via a programmer.
2. Explanation. Representing knowledge explicitly allowed systems to reason about how they came to a conclusion and use this information to explain results to users. For example, to follow the chain of inferences that led to a diagnosis and use these facts to explain the diagnosis.
3. Reasoning. Decoupling the knowledge from the processing of that knowledge enabled general purpose inference engines to be developed. These systems could develop conclusions that followed from a data set that the initial developers may not have even been aware of.[3]
Meta-reasoning
Later[when?] architectures for knowledge-based reasoning, such as the BB1 blackboard architecture (a blackboard system),[4] allowed the reasoning process itself to be affected by new inferences, providing meta-level reasoning. BB1 allowed the problem-solving process itself to be monitored. Different kinds of problem-solving (e.g., top-down, bottom-up, and opportunistic problem-solving) could be selectively mixed based on the current state of problem solving. Essentially, the problem-solver was being used both to solve a domain-level problem along with its own control problem, which could depend on the former.
Other examples of knowledge-based system architectures supporting meta-level reasoning are MRS[5] and SOAR.
Widening of application
In the 1980s and 1990s, in addition to expert systems, other applications of knowledge-based systems included real-time process control,[6] intelligent tutoring systems,[7] and problem-solvers for specific domains such as protein structure analysis,[8] construction-site layout,[9] and computer system fault diagnosis.[10]
Advances driven by enhanced architecture
As knowledge-based systems became more complex, the techniques used to represent the knowledge base became more sophisticated and included logic, term-rewriting systems,

conceptual graphs, and frames.
Frames, for example, are a way representing world knowledge using techniques that can be seen as analogous to object-oriented programming, specifically classes and subclasses, hierarchies and relations between classes, and behavior[clarification needed] of objects. With the knowledge base more structured, reasoning could now occur not only by independent rules and logical inference, but also based on interactions within the knowledge base itself. For example, procedures stored as daemons on[clarification needed] objects could fire and could replicate the chaining behavior of rules.[11]
Advances in automated reasoning
Another advancement in the 1990s was the development of special purpose automated reasoning systems called classifiers. Rather than statically declare the subsumption relations in a knowledge-base, a classifier allows the developer to simply declare facts about the world and let the classifier deduce the relations. In this way a classifier also can play the role of an inference engine.[12]
The most recent[as of?] advancement of knowledge-based systems was to adopt the technologies, especially a kind of logic called description logic, for the development of systems that use the internet. The internet often has to deal with complex, unstructured data that cannot be relied on to fit a specific data model. The technology of knowledge-based systems, and especially the ability to classify objects on demand, is ideal for such systems. The model for these kinds of knowledge-based internet systems is known as the Semantic Web.[13]
See also
- **Knowledge representation and reasoning**
- **Knowledge modeling**
- **Knowledge engine**
- **Information retrieval**
- **Reasoning system**
- **Case-based reasoning**
- **Conceptual graph**
- **Neural networks**
- **Skip to main contentAccessibility help**
- **Accessibility feedback**
-
-
- **All**
-
- **Images**
-
-
- **Videos**
-
-
- **Shopping**
-
-
- **Web**
-
-
- **News**
-
-
- **Books**
-
- **More**
- **Tools**
-

• **Knowledge-based systems**
•
• **Wikipedia**
• **https://en.wikipedia.org › wiki › Knowledge-based_syst...**
•
• **A knowledge-based system (KBS) is a computer program that reasons and uses a knowledge base to solve complex problems.**
• **Knowledge-based vs. expert... • Rule-based systems • Meta-reasoning**
• **Discussions and forums**
•
• **How would you describe a knowledge-based system and explain how ...**
•
• **Quora •**
• **3 answers**
• **• 1 year ago**
•
•
• **What are some examples of knowledge based systems?**
•
• **Quora •**
• **2 answers**
• **• 7 years ago**
•
•
• ____
• **See more**
•
• **People also ask**
• **What is a knowledge-based system example?**
• **What are knowledge base systems?**
• **What is the meaning of knowledge-based engineering?**
• **What are the two basic parts of most knowledge-based systems?**
• **Feedback**
•

• **Knowledge-based engineering**
•
• **Wikipedia**
• **https://en.wikipedia.org › wiki › Knowledge-based_eng...**
•
• **Knowledge-based engineering (KBE) is the application of knowledge-based systems technology to the domain of manufacturing design and production.**
• **Overview • KBE and CAx • KBE and knowledge... • KBE methodology**
•

• **Intelligent knowledge based systems in electrical power ...**
•
• **Springer**
• **https://link.springer.com › book**
•
• **by JR McDonald • Cited by 53 — Intelligent Knowledge Based Systems in Electrical Power Engineering details how intelligent applications can be used in the power industry.**
•

- (PDF) A Knowledge-Based System Engineering Process ...
-
- ResearchGate
- https://www.researchgate.net › publication › 25888494...
-
- This paper discusses the system architecture of the Knowledge-driven Automation (KDA) program — established at Parker in 2002 ...
-

- A knowledge-based system for the analysis of electrical ...
-
- ScienceDirect.com
- https://www.sciencedirect.com › science › article › pii
-
- by A Micarelli • 1996 • Cited by 1 — A knowledge-based system for the analysis of electrical networks in alternate current in steady condition is presented. It is a glass box expert system, ...
-

- What is a Knowledge-based System? | Definition from ...
-
- TechTarget
- https://www.techtarget.com › searchcio › knowledge-ba...
-
- Knowledge-based systems are a form of artificial intelligence (AI) designed to capture the knowledge of human experts to support decision-making. An expert ...
-

- a . EXPERT SYSTEM APPLICATIONS IN POWER SYSTEMS
-
- ResearchGate
- https://www.researchgate.net › figure › a-EXPERT-SYS...
-
- Knowledge-based systems (KBSs) or expert systems (ESs) are able to solve problems generally through the application of knowledge representing a domain and a set ...
-

- What Is a Knowledge-Based System? (With Types and Uses)
-
- Indeed
- https://www.indeed.com › ... › Career development
-
- 15 Aug 2024 — A knowledge-based system (KBS) is a type of computer system that analyzes knowledge, data and other information from sources to generate new knowledge.
-

- WHAT IS KNOWLEDGE BASED ENGINEERING (KBE)
-
- LinkedIn • PARAMCS Software Services LLP
- 6 reactions • 1 year ago
-
- Knowledge-Based Engineering (KBE) refers to the use of computer-based systems that enable the automation of engineering design processes.
- People also search for

-
- Knowledge based systems in electrical engineering pdf
-
-
- Knowledge based systems in electrical engineering ppt
-
-
- knowledge-based systems examples
-
-
- knowledge-based system tutorialspoint
-
-
- knowledge-based systems in artificial intelligence
-
-
- Components of knowledge-based system
-
-
- knowledge-based system geeksforgeeks
-
-
- Knowledge-Based system impact factor
-

1 2
3
4
5
6
7
8
9
10
Next

- Intelligent Knowledge Based Systems in Electrical Power Engineering
- Book
- Book preview
- 26/233 pages available
-
-
-
- Preview
-
- Did you like this book?
- Intelligent Knowledge Based Systems in Electrical Power Engineering details how intelligent applications can be used in the power industry. ... Google Books
- Originally published: 30 November 1997
- Editors: Stephen McArthur, Graeme Burt, Jerry Zielinski, J.R. McDonald
- Read now
-
-
- Borrow
- Find in a library

-
- Already read?
- Reading list
-
-
-
- Find in a library
- Search in WorldCat
- Search
-
- All read options
- Intelligent Knowledge Based Systems in Electrical Power Engineering
- Feedback
- South Africa
-
- Johannesburg Metropolitan Area
- - From your device
-
- -
- Update location
- HelpSend feedbackPrivacyTerms
-

•
Further reading

•
External links

Fiber-optic communication

• **Article**
• **Talk**
• **Read**
• **Edit**
• **View history**
Tools
•
•
•
•
•
•
•
•
•
•
•
•
•
•
Appearance
Text
•
Small
Standard
Large
Width
•
Standard
Wide
Color (beta)
•
Automatic
Light
Dark
From Wikipedia, the free encyclopedia
An optical fiber patching cabinet. The yellow cables are single mode fibers; the orange and blue cables are multi-mode fibers: 62.5/125 µm OM1 and 50/125 µm OM3 fibers, respectively. Stealth Communications fiber crew installing a 432-count dark fiber cable underneath the streets of Midtown Manhattan, New York City
Fiber-optic communication is a method of transmitting information from one place to another by sending pulses of infrared or visible light through an optical fiber.[1][2] The light is a form of carrier wave that is modulated to carry information.[3] Fiber is preferred over electrical cabling when high bandwidth, long distance, or immunity to electromagnetic interference is required.[4] This type of communication can transmit voice, video, and telemetry through local area networks or across

long distances.[5]
Optical fiber is used by many telecommunications companies to transmit telephone signals, internet communication, and cable television signals. Researchers at Bell Labs have reached a record bandwidth–distance product of over 100 petabit × kilometers per second using fiber-optic communication.[6][better source needed]
Background
First developed in the 1970s, fiber-optics have revolutionized the telecommunications industry and have played a major role in the advent of the Information Age.[7] Because of its advantages over electrical transmission, optical fibers have largely replaced copper wire communications in backbone networks in the developed world.[8]
The process of communicating using fiber optics involves the following basic steps:
1. creating the optical signal involving the use of a transmitter,[9] usually from an electrical signal
2. relaying the signal along the fiber, ensuring that the signal does not become too distorted or weak
3. receiving the optical signal
4. converting it into an electrical signal
Applications
Optical fiber is used by telecommunications companies to transmit telephone signals, Internet communication and cable television signals. It is also used in other industries, including medical, defense, government, industrial and commercial. In addition to serving the purposes of telecommunications, it is used as light guides, for imaging tools, lasers, hydrophones for seismic waves, SONAR, and as sensors to measure pressure and temperature.
Due to lower attenuation and interference, optical fiber has advantages over copper wire in long-distance, high-bandwidth applications. However, infrastructure development within cities is relatively difficult and time-consuming, and fiber-optic systems can be complex and expensive to install and operate. Due to these difficulties, early fiber-optic communication systems were primarily installed in long-distance applications, where they can be used to their full transmission capacity, offsetting the increased cost. The prices of fiber-optic communications have dropped considerably since 2000.[10]
The price for rolling out fiber to homes has currently become more cost-effective than that of rolling out a copper-based network. Prices have dropped to $850 per subscriber in the US and lower in countries like The Netherlands, where digging costs are low and housing density is high. [citation needed]
Since 1990, when optical-amplification systems became commercially available, the telecommunications industry has laid a vast network of intercity and transoceanic fiber communication lines. By 2002, an intercontinental network of 250,000 km of submarine communications cable with a capacity of 2.56 Tb/s was completed, and although specific network capacities are privileged information, telecommunications investment reports indicate that network capacity has increased dramatically since 2004.[11] As of 2020, over 5 billion kilometers of fiber-optic cable has been deployed around the globe.[12]
History
In 1880 Alexander Graham Bell and his assistant Charles Sumner Tainter created a very early precursor to fiber-optic communications, the Photophone, at Bell's newly established Volta Laboratory in Washington, D.C. Bell considered it his most important invention. The device allowed for the transmission of sound on a beam of light. On June 3, 1880, Bell conducted the world's first wireless telephone transmission between two buildings, some 213 meters apart.[13][14] Due to its use of an atmospheric transmission medium, the Photophone would not prove practical until advances in laser and optical fiber technologies permitted the secure transport of light. The Photophone's first practical use came in military communication systems many decades later.[15]
In 1954 Harold Hopkins and Narinder Singh Kapany showed that rolled fiber glass allowed light to be transmitted.[16] Jun-ichi Nishizawa, a Japanese scientist at Tohoku University, proposed the use of optical fibers for communications in 1963.[17] Nishizawa invented the PIN diode and the static induction transistor, both of which contributed to the development of optical fiber communications. [18][19]

In 1966 Charles K. Kao and George Hockham at Standard Telecommunication Laboratories showed that the losses of 1,000 dB/km in existing glass (compared to 5–10 dB/km in coaxial cable) were due to contaminants which could potentially be removed.
Optical fiber with attenuation low enough for communication purposes (about 20 dB/km) was developed in 1970 by Corning Glass Works. At the same time, GaAs semiconductor lasers were developed that were compact and therefore suitable for transmitting light through fiber optic cables for long distances.
In 1973, Optelecom, Inc., co-founded by the inventor of the laser, Gordon Gould, received a contract from ARPA for one of the first optical communication systems. Developed for Army Missile Command in Huntsville, Alabama, the system was intended to allow a short-range missile with video processing to communicate by laser to the ground by means of a five-kilometer long optical fiber that unspooled from the missile as it flew.[20] Optelecom then delivered the first commercial optical communications system to Chevron.[21]
After a period of research starting from 1975, the first commercial fiber-optic telecommunications system was developed which operated at a wavelength around 0.8 μm and used GaAs semiconductor lasers. This first-generation system operated at a bit rate of 45 Mbit/s with repeater spacing of up to 10 km. Soon on 22 April 1977, General Telephone and Electronics sent the first live telephone traffic through fiber optics at a 6 Mbit/s throughput in Long Beach, California.[citation needed]
In October 1973, Corning Glass signed a development contract with CSELT and Pirelli aimed to test fiber optics in an urban environment: in September 1977, the second cable in this test series, named COS-2, was experimentally deployed in two lines (9 km) in Turin, for the first time in a big city, at a speed of 140 Mbit/s.[22]
The second generation of fiber-optic communication was developed for commercial use in the early 1980s, operated at 1.3 μm and used InGaAsP semiconductor lasers. These early systems were initially limited by multi-mode fiber dispersion, and in 1981 the single-mode fiber was revealed to greatly improve system performance, however practical connectors capable of working with single mode fiber proved difficult to develop. Canadian service provider SaskTel had completed construction of what was then the world's longest commercial fiber optic network, which covered 3,268 km (2,031 mi) and linked 52 communities.[23] By 1987, these systems were operating at bit rates of up to 1.7 Gbit/s with repeater spacing up to 50 km (31 mi).
The first transatlantic telephone cable to use optical fiber was TAT-8, based on Desurvire optimized laser amplification technology. It went into operation in 1988.
Third-generation fiber-optic systems operated at 1.55 μm and had losses of about 0.2 dB/km. This development was spurred by the discovery of indium gallium arsenide and the development of the indium gallium arsenide photodiode by Pearsall. Engineers overcame earlier difficulties with pulse-spreading using conventional InGaAsP semiconductor lasers at that wavelength by using dispersion-shifted fibers designed to have minimal dispersion at 1.55 μm or by limiting the laser spectrum to a single longitudinal mode. These developments eventually allowed third-generation systems to operate commercially at 2.5 Gbit/s with repeater spacing in excess of 100 km (62 mi).
The fourth generation of fiber-optic communication systems used optical amplification to reduce the need for repeaters and wavelength-division multiplexing (WDM) to increase data capacity. The introduction of WDM was the start of optical networking, as WDM became the technology of choice for fiber-optic bandwidth expansion.[24] The first to market with a dense WDM system was Ciena Corp., in June 1996.[25] The introduction of optical amplifiers and WDM caused system capacity to double every six months from 1992 until a bit rate of 10 Tb/s was reached by 2001. In 2006 a bit-rate of 14 Tb/s was reached over a single 160 km (99 mi) line using optical amplifiers.[26] As of 2021, Japanese scientists transmitted 319 terabits per second over 3,000 kilometers with four-core fiber cables with standard cable diameter.[27]
The focus of development for the fifth generation of fiber-optic communications is on extending the wavelength range over which a WDM system can operate. The conventional wavelength window, known as the C band, covers the wavelength range 1525–1565 nm, and dry fiber has a low-loss window promising an extension of that range to 1300–1650 nm.[citation needed] Other developments include the concept of optical solitons, pulses that preserve their shape by

counteracting the effects of dispersion with the nonlinear effects of the fiber by using pulses of a specific shape.
In the late 1990s through 2000, industry promoters, and research companies such as KMI, and RHK predicted massive increases in demand for communications bandwidth due to increased use of the Internet, and commercialization of various bandwidth-intensive consumer services, such as video on demand. Internet Protocol data traffic was increasing exponentially, at a faster rate than integrated circuit complexity had increased under Moore's Law. From the bust of the dot-com bubble through 2006, however, the main trend in the industry has been consolidation of firms and offshoring of manufacturing to reduce costs. Companies such as Verizon and AT&T have taken advantage of fiber-optic communications to deliver a variety of high-throughput data and broadband services to consumers' homes.
Technology
Modern fiber-optic communication systems generally include optical transmitters that convert electrical signals into optical signals, optical fiber cables to carry the signal, optical amplifiers, and optical receivers to convert the signal back into an electrical signal. The information transmitted is typically digital information generated by computers or telephone systems.
Transmitters
A GBIC module (shown here with its cover removed), is an optical and electrical transceiver, a device combining a transmitter and a receiver in a single housing. The electrical connector is at top right and the optical connectors are at bottom left.
The most commonly used optical transmitters are semiconductor devices such as light-emitting diodes (LEDs) and laser diodes. The difference between LEDs and laser diodes is that LEDs produce incoherent light, while laser diodes produce coherent light. For use in optical communications, semiconductor optical transmitters must be designed to be compact, efficient and reliable, while operating in an optimal wavelength range and directly modulated at high frequencies.
In its simplest form, an LED emits light through spontaneous emission, a phenomenon referred to as electroluminescence. The emitted light is incoherent with a relatively wide spectral width of 30–60 nm.[a] The large spectrum width of LEDs is subject to higher fiber dispersion, considerably limiting their bit rate-distance product (a common measure of usefulness). LEDs are suitable primarily for local-area-network applications with bit rates of 10–100 Mbit/s and transmission distances of a few kilometers.
LED light transmission is inefficient, with only about 1% of input power, or about 100 microwatts, eventually converted into launched power coupled into the optical fiber.[28]
LEDs have been developed that use several quantum wells to emit light at different wavelengths over a broad spectrum and are currently in use for local-area wavelength-division multiplexing (WDM) applications.
LEDs have been largely superseded by vertical-cavity surface-emitting laser (VCSEL) devices, which offer improved speed, power and spectral properties, at a similar cost. However, due to their relatively simple design, LEDs are very useful for very low-cost applications. Commonly used classes of semiconductor laser transmitters used in fiber optics include VCSEL, Fabry–Pérot and distributed-feedback laser.
A semiconductor laser emits light through stimulated emission rather than spontaneous emission, which results in high output power (~100 mW) as well as other benefits related to the nature of coherent light. The output of a laser is relatively directional, allowing high coupling efficiency (~50%) into single-mode fiber. Common VCSEL devices also couple well to multimode fiber. The narrow spectral width also allows for high bit rates since it reduces the effect of chromatic dispersion. Furthermore, semiconductor lasers can be modulated directly at high frequencies because of short recombination time.
Laser diodes are often directly modulated, that is the light output is controlled by a current applied directly to the device. For very high data rates or very long distance links, a laser source may be operated continuous wave, and the light modulated by an external device, an optical modulator, such as an electro-absorption modulator or Mach–Zehnder interferometer. External modulation increases the achievable link distance by eliminating laser chirp, which broadens the linewidth in

directly modulated lasers, increasing the chromatic dispersion in the fiber. For very high bandwidth efficiency, coherent modulation can be used to vary the phase of the light in addition to the amplitude, enabling the use of QPSK, QAM, and OFDM. "Dual-polarization quadrature phase shift keying is a modulation format that effectively sends four times as much information as traditional optical transmissions of the same speed."[29]
Receivers
The main component of an optical receiver is a photodetector which converts light into electricity using the photoelectric effect. The primary photodetectors for telecommunications are made from Indium gallium arsenide. The photodetector is typically a semiconductor-based photodiode. Several types of photodiodes include p–n photodiodes, p–i–n photodiodes, and avalanche photodiodes. Metal-semiconductor-metal (MSM) photodetectors are also used due to their suitability for circuit integration in regenerators and wavelength-division multiplexers.
Since light may be attenuated and distorted while passing through the fiber, photodetectors are typically coupled with a transimpedance amplifier and a limiting amplifier to produce a digital signal in the electrical domain recovered from the incoming optical signal. Further signal processing such as clock recovery from data performed by a phase-locked loop may also be applied before the data is passed on.
Coherent receivers use a local oscillator laser in combination with a pair of hybrid couplers and four photodetectors per polarization, followed by high-speed ADCs and digital signal processing to recover data modulated with QPSK, QAM, or OFDM.[citation needed]
Digital predistortion
An optical communication system transmitter consists of a digital-to-analog converter (DAC), a driver amplifier and a Mach–Zehnder modulator. The deployment of higher modulation formats (>4-QAM) or higher baud Rates (>32 GBd) diminishes the system performance due to linear and non-linear transmitter effects. These effects can be categorized as linear distortions due to DAC bandwidth limitation and transmitter I/Q skew as well as non-linear effects caused by gain saturation in the driver amplifier and the Mach–Zehnder modulator. Digital predistortion counteracts the degrading effects and enables Baud rates up to 56 GBd and modulation formats like 64-QAM and 128-QAM with the commercially available components. The transmitter digital signal processor performs digital predistortion on the input signals using the inverse transmitter model before sending the samples to the DAC.
Older digital predistortion methods only addressed linear effects. Recent publications also consider non-linear distortions. Berenguer et al models the Mach–Zehnder modulator as an independent Wiener system and the DAC and the driver amplifier are modeled by a truncated, time-invariant Volterra series.[30] Khanna et al use a memory polynomial to model the transmitter components jointly.[31] In both approaches the Volterra series or the memory polynomial coefficients are found using indirect-learning architecture. Duthel et al records, for each branch of the Mach-Zehnder modulator, several signals at different polarity and phases. The signals are used to calculate the optical field. Cross-correlating in-phase and quadrature fields identifies the timing skew. The frequency response and the non-linear effects are determined by the indirect-learning architecture.[32]
Fiber cable types
A cable reel trailer with conduit that can carry optical fiber Multi-mode optical fiber in an underground service pit
An optical fiber cable consists of a core, cladding, and a buffer (a protective outer coating), in which the cladding guides the light along the core by using the method of total internal reflection. The core and the cladding (which has a lower-refractive-index) are usually made of high-quality silica glass, although they can both be made of plastic as well. Connecting two optical fibers is done by fusion splicing or mechanical splicing and requires special skills and interconnection technology due to the microscopic precision required to align the fiber cores.[33]
Two main types of optical fiber used in optic communications include multi-mode optical fibers and single-mode optical fibers. A multi-mode optical fiber has a larger core (≥ 50 micrometers), allowing less precise, cheaper transmitters and receivers to connect to it as well as cheaper connectors. However, a multi-mode fiber introduces multimode distortion, which often limits the bandwidth and

length of the link. Furthermore, because of its higher dopant content, multi-mode fibers are usually expensive and exhibit higher attenuation. The core of a single-mode fiber is smaller (Detection
Detection theory is a means to quantify the ability to discern between information-bearing patterns and random patterns (called noise).
Typically boils down to a “hypothesis test” problem.
Introduction>
Modeling for Detection and Estimation
Introduction>
Estimation or Detection–
which comes first?
Introduction> Communication Examples
Introduction> Communication Examples
Introduction> Communication Examples
Introduction> System Identification
Introduction> Clustering in Social Networks
Introduction> Parameter Estimation Via
Sensor Networks
Next Lecture:
Basics- A Refresher
Digital control

- **Article**
- **Talk**
- **Read**
- **Edit**
- **View history**

Tools

-
-
-
-
-
-
-
-
-
-
-
-

Appearance
Text

-

Small
Standard
Large
Width

-

Standard
Wide
Color (beta)

-

Automatic
Light
Dark

From Wikipedia, the free encyclopedia
This article includes a list of general references, but it lacks sufficient corresponding inline citations. Please help to improve this article by introducing more precise citations. (September 2020) (Learn how and when to remove this message)

Digital control is a branch of control theory that uses digital computers to act as system controllers. Depending on the requirements, a digital control system can take the form of a microcontroller to an ASIC to a standard desktop computer. Since a digital computer is a discrete system, the Laplace transform is replaced with the Z-transform. Since a digital computer has finite precision (See quantization), extra care is needed to ensure the error in coefficients, analog-to-digital conversion, digital-to-analog conversion, etc. are not producing undesired or unplanned effects.
Since the creation of the first digital computer in the early 1940s the price of digital computers has dropped considerably, which has made them key pieces to control systems because they are easy to configure and reconfigure through software, can scale to the limits of the memory or storage space without extra cost, parameters of the program can change with time (See adaptive control) and digital computers are much less prone to environmental conditions than capacitors, inductors, etc.
Digital controller implementation
A digital controller is usually cascaded with the plant in a feedback system. The rest of the system can either be digital or analog.
Typically, a digital controller requires:
- **Analog-to-digital conversion to convert analog inputs to machine-readable (digital) format**
- **Digital-to-analog conversion to convert digital outputs to a form that can be input to a plant (analog)**
- **A program that relates the outputs to the inputs**

Output program
- **Outputs from the digital controller are functions of current and past input samples, as well as past output samples - this can be implemented by storing relevant values of input and output in registers. The output can then be formed by a weighted sum of these stored values.**

The programs can take numerous forms and perform many functions
- **A digital filter for low-pass filtering**
- **A state space model of a system to act as a state observer**
- **A telemetry system**

Stability
Although a controller may be stable when implemented as an analog controller, it could be unstable when implemented as a digital controller due to a large sampling interval. During sampling the aliasing modifies the cutoff parameters. Thus the sample rate characterizes the transient response and stability of the compensated system, and must update the values at the controller input often enough so as to not cause instability.
When substituting the frequency into the z operator, regular stability criteria still apply to discrete control systems. Nyquist criteria apply to z-domain transfer functions as well as being general for complex valued functions. Bode stability criteria apply similarly. Jury criterion determines the discrete system stability about its characteristic polynomial.
Design of digital controller in s-domain
The digital controller can also be designed in the s-domain (continuous). The Tustin transformation can transform the continuous compensator to the respective digital compensator. The digital compensator will achieve an output that approaches the output of its respective analog controller as the sampling interval is decreased.

Tustin transformation deduction
Tustin is the Padé(1,1) approximation of the exponential function :

And its inverse

Digital control theory is the technique to design strategies in discrete time, (and/or) quantized amplitude (and/or) in (binary) coded form to be implemented in computer systems (microcontrollers, microprocessors) that will control the analog (continuous in time and amplitude) dynamics of analog systems. From this consideration many errors from classical digital control were identified and solved and new methods were proposed:

- Marcelo Tredinnick and Marcelo Souza and their new type of analog-digital mapping[1][2][3]
- Yutaka Yamamoto and his "lifting function space model"[4]
- Alexander Sesekin and his studies about impulsive systems.[5]
- M.U. Akhmetov and his studies about impulsive and pulse control[6]

Design of digital controller in z-domain

The digital controller can also be designed in the z-domain (discrete). The Pulse Transfer Function (PTF) represents the digital viewpoint of the continuous process when interfaced with appropriate ADC and DAC, and for a specified sample time is obtained as:[7]

Where denotes z-Transform for the chosen sample time . There are many ways to directly design a digital controller to achieve a given specification.[7] For a type-0 system under unity negative feedback control, Michael Short and colleagues have shown that a relatively simple but effective method to synthesize a controller for a given (monic) closed-loop denominator polynomial and preserve the (scaled) zeros of the PTF numerator is to use the design equation:[8]

Where the scalar term ensures the controller exhibits integral action, and a steady-state gain of unity is achieved in the closed-loop. The resulting closed-loop discrete transfer function from the z-Transform of reference input to the z-Transform of process output is then given by:[8]

Since process time delay manifests as leading co-efficient(s) of zero in the process PTF numerator , the synthesis method above inherently yields a predictive controller if any such delay is present in the continuous plant.[8]

See also

- Sampled data systems
- Adaptive control
- Analog control
- Control theory
- Digital
- Feedback, Negative feedback, Positive feedback
- Laplace transform
- Real-time control
- Z-transform

References

1.
• "Referência Completa". mtc-m18.sid.inpe.br. Archived from the original on March 5, 2012.
• • "Discrete attitude control of artificial satellites with flexible appendages" (PDF). mtc-m05.sid.inpe.br. Archived from the original (PDF) on 6 July 2011. Retrieved 12 January 2022.
• • "An Analytical Approach for Discrete Controllers Design Using a New S-Z Mapping with Two Tuning Parameters". www.sae.org. Archived from the original on 13 January 2013. Retrieved 27 January 2022.
• • Yamamoto, Yutaka (1996). "A Retrospective View on Sampled-Data - Control Systems" (PDF). Kyoto University. Archived from the original (PDF) on 22 July 2011. Retrieved 12 January 2022.
• • Zavalishchin, S. T.; Sesekin, A. N. (28 February 1997). Dynamic Impulse Systems: Theory and Applications. Springer. ISBN 0792343948.
• • "Author page". Association for Computing Machinery. Retrieved 2009-03-20.[permanent dead link]
• • Åström, Karl J.; Wittenmark, Björn (2013-06-13). Computer-Controlled Systems: Theory and Design, Third Edition. Courier Corporation. ISBN 978-0-486-28404-0.

8. • Short, Michael; Abugchem, Fathi; Abrar, Usama (2015-02-11). "Dependable Control for Wireless Distributed Control Systems". Electronics. 4 (4): 857–878. doi:10.3390/electronics4040857.
• FRANKLIN, G.F.; POWELL, J.D., Emami-Naeini, A., Digital Control of Dynamical Systems, 3rd Ed (1998). Ellis-Kagle Press, Half Moon Bay, CA ISBN 978-0-9791226-1-3
• KATZ, P. Digital control using microprocessors. Englewood Cliffs: Prentice-Hall, 293p. 1981.
• OGATA, K. Discrete-time control systems. Englewood Cliffs: Prentice-Hall,984p. 1987.
• PHILLIPS, C.L.; NAGLE, H. T. Digital control system analysis and design. Englewood Cliffs, New Jersey: Prentice Hall International. 1995.
• M. Sami Fadali, Antonio Visioli, (2009) "Digital Control Engineering", Academic Press, ISBN 978-0-12-374498-2.
• JURY, E.I. Sampled-data control systems. New-York: John Wiley. 1958.

Category:
• Control theory
• This page was last edited on 26 Nov

•
•
•
•
•
•
•
•
•
•
•

• Donate
• Create account
• Log in

•
•
Contents
•
(Top)

•
Structure

• •
Special-purpose designs

•
Speed and power considerations

•
Embedded applications

•
History

•
o

First projects

Four-Phase Systems AL1 (1969)

Garrett AiResearch CADC (1970)

Gilbert Hyatt (1970)

Texas Instruments TMX 1795 (1970–1971)

Texas Instruments TMS 1802NC (1971)

Pico/General Instrument (1971)

Intel 4004 (1971)

- **8-bit designs**
- **12-bit designs**
- **16-bit designs**
- **32-bit designs**
- **64-bit designs in personal computers**
- **RISC**
- **SMP and multi-core design**

- **Market statistics**
- **See also**
- **Notes**
-

References

•
External links

Microprocessor

• **Article**
• **Talk**
• **Read**
• **Edit**
• **View history**
Tools
•
•
•
•
•
•
•
•
•
•
•
•
•
•
•
Appearance
Text
•
Small
Standard
Large
Width
•
Standard
Wide
Color (beta)
•
Automatic
Light
Dark
From Wikipedia, the free encyclopedia
Texas Instruments TMS1000 Intel 4004 Motorola 6800 (MC6800) A modern 64-bit x86-64 processor (AMD Ryzen Threadripper 7970X, based on Zen 4, 2023) AMD Ryzen 7 1800X (2017, based on Zen) processor in an AM4 socket on a motherboard
A microprocessor is a computer processor for which the data processing logic and control is included on a single integrated circuit (IC), or a small number of ICs. The microprocessor contains the arithmetic, logic, and control circuitry required to perform the functions of a computer's central processing unit (CPU). The IC is capable of interpreting and executing program instructions and performing arithmetic operations.[1] The microprocessor is a multipurpose, clock-driven, register-based, digital integrated circuit that accepts binary data as input, processes it according to instructions stored in its memory, and provides results (also in binary form) as output.

Microprocessors contain both combinational logic and sequential digital logic, and operate on numbers and symbols represented in the binary number system.

The integration of a whole CPU onto a single or a few integrated circuits using Very-Large-Scale Integration (VLSI) greatly reduced the cost of processing power. Integrated circuit processors are produced in large numbers by highly automated metal–oxide–semiconductor (MOS) fabrication processes, resulting in a relatively low unit price. Single-chip processors increase reliability because there are fewer electrical connections that can fail. As microprocessor designs improve, the cost of manufacturing a chip (with smaller components built on a semiconductor chip the same size) generally stays the same according to Rock's law.

Before microprocessors, small computers had been built using racks of circuit boards with many medium- and small-scale integrated circuits, typically of TTL type. Microprocessors combined this into one or a few large-scale ICs. While there is disagreement over who deserves credit for the invention of the microprocessor, the first commercially available microprocessor was the Intel 4004, designed by Federico Faggin and introduced in 1971.[2]

Continued increases in microprocessor capacity have since rendered other forms of computers almost completely obsolete (see history of computing hardware), with one or more microprocessors used in everything from the smallest embedded systems and handheld devices to the largest mainframes and supercomputers.

A microprocessor is distinct from a microcontroller including a system on a chip.[3][4] A microprocessor is related but distinct from a digital signal processor, a specialized microprocessor chip, with its architecture optimized for the operational needs of digital signal processing.[5]: 104–107 [6]

Structure

A block diagram of the architecture of the Z80 microprocessor, showing the arithmetic and logic section, register file, control logic section, and buffers to external address and data lines

The complexity of an integrated circuit is bounded by physical limitations on the number of transistors that can be put onto one chip, the number of package terminations that can connect the processor to other parts of the system, the number of interconnections it is possible to make on the chip, and the heat that the chip can dissipate. Advancing technology makes more complex and powerful chips feasible to manufacture.

A minimal hypothetical microprocessor might include only an arithmetic logic unit (ALU), and a control logic section. The ALU performs addition, subtraction, and operations such as AND or OR. Each operation of the ALU sets one or more flags in a status register, which indicate the results of the last operation (zero value, negative number, overflow, or others). The control logic retrieves instruction codes from memory and initiates the sequence of operations required for the ALU to carry out the instruction. A single operation code might affect many individual data paths, registers, and other elements of the processor.

As integrated circuit technology advanced, it was feasible to manufacture more and more complex processors on a single chip. The size of data objects became larger; allowing more transistors on a chip allowed word sizes to increase from 4- and 8-bit words up to today's 64-bit words. Additional features were added to the processor architecture; more on-chip registers sped up programs, and complex instructions could be used to make more compact programs. Floating-point arithmetic, for example, was often not available on 8-bit microprocessors, but had to be carried out in software. Integration of the floating-point unit, first as a separate integrated circuit and then as part of the same microprocessor chip, sped up floating-point calculations.

Occasionally, physical limitations of integrated circuits made such practices as a bit slice approach necessary. Instead of processing all of a long word on one integrated circuit, multiple circuits in parallel processed subsets of each word. While this required extra logic to handle, for example, carry and overflow within each slice, the result was a system that could handle, for example, 32-bit words using integrated circuits with a capacity for only four bits each.

The ability to put large numbers of transistors on one chip makes it feasible to integrate memory on the same die as the processor. This CPU cache has the advantage of faster access than off-chip memory and increases the processing speed of the system for many applications. Processor clock frequency has increased more rapidly than external memory speed, so cache memory is necessary

if the processor is not to be delayed by slower external memory.
The design of some processors has become complicated enough to be difficult to fully test, and this has caused problems at large cloud providers.[7]
Special-purpose designs
A microprocessor is a general purpose processing entity. Several specialized processing devices have followed:
• A digital signal processor (DSP) is specialized for signal processing.
• Graphics processing units (GPUs) are processors designed primarily for real-time rendering of images.
• Other specialized units exist for video processing and machine vision. (See: Hardware acceleration.)
• Microcontrollers in embedded systems and peripheral devices.
• Systems on chip (SoCs) often integrate one or more microprocessor and microcontroller cores with other components such as radio modems, and are used in smartphones and tablet computers.
Speed and power considerations
Intel Core i9-9900K (2018, based on Coffee Lake)
Microprocessors can be selected for differing applications based on their word size, which is a measure of their complexity. Longer word sizes allow each clock cycle of a processor to carry out more computation, but correspond to physically larger integrated circuit dies with higher standby and operating power consumption.[8] 4-, 8- or 12-bit processors are widely integrated into microcontrollers operating embedded systems. Where a system is expected to handle larger volumes of data or require a more flexible user interface, 16-, 32- or 64-bit processors are used. An 8- or 16-bit processor may be selected over a 32-bit processor for system on a chip or microcontroller applications that require extremely low-power electronics, or are part of a mixed-signal integrated circuit with noise-sensitive on-chip analog electronics such as high-resolution analog to digital converters, or both. Some people say that running 32-bit arithmetic on an 8-bit chip could end up using more power, as the chip must execute software with multiple instructions.[9] However, others say that modern 8-bit chips are always more power-efficient than 32-bit chips when running equivalent software routines.[10]
Embedded applications
Thousands of items that were traditionally not computer-related include microprocessors. These include household appliances, vehicles (and their accessories), tools and test instruments, toys, light switches/dimmers and electrical circuit breakers, smoke alarms, battery packs, and hi-fi audio/visual components (from DVD players to phonograph turntables). Such products as cellular telephones, DVD video system and HDTV broadcast systems fundamentally require consumer devices with powerful, low-cost, microprocessors. Increasingly stringent pollution control standards effectively require automobile manufacturers to use microprocessor engine management systems to allow optimal control of emissions over the widely varying operating conditions of an automobile. Non-programmable controls would require bulky, or costly implementation to achieve the results possible with a microprocessor.
A microprocessor control program (embedded software) can be tailored to fit the needs of a product line, allowing upgrades in performance with minimal redesign of the product. Unique features can be implemented in product line's various models at negligible production cost.
Microprocessor control of a system can provide control strategies that would be impractical to implement using electromechanical controls or purpose-built electronic controls. For example, an internal combustion engine's control system can adjust ignition timing based on engine speed, load, temperature, and any observed tendency for knocking—allowing the engine to operate on a range of fuel grades.
History
See also: Microprocessor chronology
The advent of low-cost computers on integrated circuits has transformed modern society. General-purpose microprocessors in personal computers are used for computation, text editing, multimedia display, and communication over the Internet. Many more microprocessors are part of embedded systems, providing digital control over myriad objects from appliances to automobiles to cellular

phones and industrial process control. Microprocessors perform binary operations based on Boolean logic, named after George Boole. The ability to operate computer systems using Boolean Logic was first proven in a 1938 thesis by master's student Claude Shannon, who later went on to become a professor. Shannon is considered "The Father of Information Theory". In 1951 Microprogramming was invented by Maurice Wilkes at the University of Cambridge, UK, from the realisation that the central processor could be controlled by a specialised program in a dedicated ROM.[11] Wilkes is also credited with the idea of symbolic labels, macros and subroutine libraries.[12]

Following the development of MOS integrated circuit chips in the early 1960s, MOS chips reached higher transistor density and lower manufacturing costs than bipolar integrated circuits by 1964. MOS chips further increased in complexity at a rate predicted by Moore's law, leading to large-scale integration (LSI) with hundreds of transistors on a single MOS chip by the late 1960s. The application of MOS LSI chips to computing was the basis for the first microprocessors, as engineers began recognizing that a complete computer processor could be contained on several MOS LSI chips.[13] Designers in the late 1960s were striving to integrate the central processing unit (CPU) functions of a computer onto a handful of MOS LSI chips, called microprocessor unit (MPU) chipsets.

While there is disagreement over who invented the microprocessor,[2][14] the first commercially available microprocessor was the Intel 4004, released as a single MOS LSI chip in 1971.[15] The single-chip microprocessor was made possible with the development of MOS silicon-gate technology (SGT).[16] The earliest MOS transistors had aluminium metal gates, which Italian physicist Federico Faggin replaced with silicon self-aligned gates to develop the first silicon-gate MOS chip at Fairchild Semiconductor in 1968.[16] Faggin later joined Intel and used his silicon-gate MOS technology to develop the 4004, along with Marcian Hoff, Stanley Mazor and Masatoshi Shima in 1971.[17] The 4004 was designed for Busicom, which had earlier proposed a multi-chip design in 1969, before Faggin's team at Intel changed it into a new single-chip design. Intel introduced the first commercial microprocessor, the 4-bit Intel 4004, in 1971. It was soon followed by the 8-bit microprocessor Intel 8008 in 1972. The MP944 chipset used in the F-14 Central Air Data Computer in 1970 has also been cited as an early microprocessor, but was not known to the public until declassified in 1998.

Other embedded uses of 4-bit and 8-bit microprocessors, such as terminals, printers, various kinds of automation etc., followed soon after. Affordable 8-bit microprocessors with 16-bit addressing also led to the first general-purpose microcomputers from the mid-1970s on.

The first use of the term "microprocessor" is attributed to Viatron Computer Systems[18] describing the custom integrated circuit used in their System 21 small computer system announced in 1968.

Since the early 1970s, the increase in capacity of microprocessors has followed Moore's law; this originally suggested that the number of components that can be fitted onto a chip doubles every year. With present technology, it is actually every two years,[19] [obsolete source] and as a result Moore later changed the period to two years.[20]

First projects

These projects delivered a microprocessor at about the same time: Garrett AiResearch's Central Air Data Computer (CADC) (1970), Texas Instruments' TMS 1802NC (September 1971) and Intel's 4004 (November 1971, based on an earlier 1969 Busicom design). Arguably, Four-Phase Systems AL1 microprocessor was also delivered in 1969.

Four-Phase Systems AL1 (1969)

The Four-Phase Systems AL1 was an 8-bit bit slice chip containing eight registers and an ALU.[21] It was designed by Lee Boysel in 1969.[22][23][24] At the time, it formed part of a nine-chip, 24-bit CPU with three AL1s. It was later called a microprocessor when, in response to 1990s litigation by Texas Instruments, Boysel constructed a demonstration system where a single AL1 formed part of a courtroom demonstration computer system, together with RAM, ROM, and an input-output device.[25]

Garrett AiResearch CADC (1970)

This section relies excessively on references to primary sources. Please improve this section by adding secondary or tertiary sources.
Find sources: "Microprocessor" – news • newspapers • books • scholar • JSTOR (March 2010) (Learn how and when to remove this message)

Further information: F-14 CADC
In 1968, Garrett AiResearch (who employed designers Ray Holt and Steve Geller) was invited to produce a digital computer to compete with electromechanical systems then under development for the main flight control computer in the US Navy's new F-14 Tomcat fighter. The design was complete by 1970, and used a MOS-based chipset as the core CPU. The design was significantly (approximately 20 times) smaller and much more reliable than the mechanical systems it competed against and was used in all of the early Tomcat models. This system contained "a 20-bit, pipelined, parallel multi-microprocessor". The Navy refused to allow publication of the design until 1997. Released in 1998, the documentation on the CADC, and the MP944 chipset, are well known. Ray Holt's autobiographical story of this design and development is presented in the book: The Accidental Engineer.[26][27]
Ray Holt graduated from California State Polytechnic University, Pomona in 1968, and began his computer design career with the CADC.[28] From its inception, it was shrouded in secrecy until 1998 when at Holt's request, the US Navy allowed the documents into the public domain. Holt has claimed that no one has compared this microprocessor with those that came later.[29] According to Parab et al. (2007),
The scientific papers and literature published around 1971 reveal that the MP944 digital processor used for the F-14 Tomcat aircraft of the US Navy qualifies as the first microprocessor. Although interesting, it was not a single-chip processor, as was not the Intel 4004 – they both were more like a set of parallel building blocks you could use to make a general-purpose form. It contains a CPU, RAM, ROM, and two other support chips like the Intel 4004. It was made from the same P-channel technology, operated at military specifications and had larger chips – an excellent computer engineering design by any standards. Its design indicates a major advance over Intel, and two year earlier. It actually worked and was flying in the F-14 when the Intel 4004 was announced. It indicates that today's industry theme of converging DSP-microcontroller architectures was started in 1971.[30]
This convergence of DSP and microcontroller architectures is known as a digital signal controller.[31]
Gilbert Hyatt (1970)
In 1990, American engineer Gilbert Hyatt was awarded U.S. Patent No. 4,942,516,[32] which was based on a 16-bit serial computer he built at his Northridge, California, home in 1969 from boards of bipolar chips after quitting his job at Teledyne in 1968;[2][33] though the patent had been submitted in December 1970 and prior to Texas Instruments' filings for the TMX 1795 and TMS 0100, Hyatt's invention was never manufactured.[33][34][35] This nonetheless led to claims that Hyatt was the inventor of the microprocessor and the payment of substantial royalties through a Philips N.V. subsidiary,[36] until Texas Instruments prevailed in a complex legal battle in 1996, when the U.S. Patent Office overturned key parts of the patent, while allowing Hyatt to keep it.[2][37] Hyatt said in a 1990 Los Angeles Times article that his invention would have been created had his prospective investors backed him, and that the venture investors leaked details of his chip to the industry, though he did not elaborate with evidence to support this claim.[33] In the same article, The Chip author T.R. Reid was quoted as saying that historians may ultimately place Hyatt as a co-inventor of the microprocessor, in the way that Intel's Noyce and TI's Kilby share credit for the invention of the chip in 1958: "Kilby got the idea first, but Noyce made it practical. The legal ruling finally favored Noyce, but they are considered co-inventors. The same could happen here."[33] Hyatt would go on to fight a decades-long legal battle with the state of California over alleged unpaid taxes on his patent's windfall after 1990, which would culminate in a landmark Supreme Court case addressing states' sovereign immunity in Franchise Tax Board of California v. Hyatt (2019).
Texas Instruments TMX 1795 (1970–1971)
Along with Intel (who developed the 8008), Texas Instruments developed in 1970–1971 a one-chip

CPU replacement for the Datapoint 2200 terminal, the TMX 1795 (later TMC 1795.) Like the 8008, it was rejected by customer Datapoint. According to Gary Boone, the TMX 1795 never reached production. Still it reached a working prototype state at 1971 February 24, therefore it is the world's first 8-bit microprocessor.[38] Since it was built to the same specification, its instruction set was very similar to the Intel 8008.[39][40]
Texas Instruments TMS 1802NC (1971)
The TMS1802NC was announced September 17, 1971, and implemented a four-function calculator. The TMS1802NC, despite its designation, was not part of the TMS 1000 series; it was later redesignated as part of the TMS 0100 series, which was used in the TI Datamath calculator. Although marketed as a calculator-on-a-chip, the TMS1802NC was fully programmable, including on the chip a CPU with an 11-bit instruction word, 3520 bits (320 instructions) of ROM and 182 bits of RAM.[39][41][40][42]
Pico/General Instrument (1971)
The PICO1/GI250 chip introduced in 1971: It was designed by Pico Electronics (Glenrothes, Scotland) and manufactured by General Instrument of Hicksville NY.
In 1971, Pico Electronics[43] and General Instrument (GI) introduced their first collaboration in ICs, a complete single-chip calculator IC for the Monroe/Litton Royal Digital III calculator. This chip could also arguably lay claim to be one of the first microprocessors or microcontrollers having ROM, RAM and a RISC instruction set on-chip. The layout for the four layers of the PMOS process was hand drawn at x500 scale on mylar film, a significant task at the time given the complexity of the chip.
Pico was a spinout by five GI design engineers whose vision was to create single-chip calculator ICs. They had significant previous design experience on multiple calculator chipsets with both GI and Marconi-Elliott.[44] The key team members had originally been tasked by Elliott Automation to create an 8-bit computer in MOS and had helped establish a MOS Research Laboratory in Glenrothes, Scotland in 1967.
Calculators were becoming the largest single market for semiconductors so Pico and GI went on to have significant success in this burgeoning market. GI continued to innovate in microprocessors and microcontrollers with products including the CP1600, IOB1680 and PIC1650.[45] In 1987, the GI Microelectronics business was spun out into the Microchip PIC microcontroller business.
Intel 4004 (1971)
Main article: Intel 4004
The 4004 with cover removed (left) and as actually used (right)
The Intel 4004 is often (falsely) regarded as the first true microprocessor built on a single chip,[46][47] priced at US$60 (equivalent to $450 in 2023).[48] The claim of being the first is definitely false, as the earlier TMS1802NC was also a true microprocessor built on a single chip and the same applies for the - prototype only - 8-bit TMX 1795.[38] The first known advertisement for the 4004 is dated November 15, 1971, and appeared in Electronic News.[citation needed] The microprocessor was designed by a team consisting of Italian engineer Federico Faggin, American engineers Marcian Hoff and Stanley Mazor, and Japanese engineer Masatoshi Shima.[49]
The project that produced the 4004 originated in 1969, when Busicom, a Japanese calculator manufacturer, asked Intel to build a chipset for high-performance desktop calculators. Busicom's original design called for a programmable chip set consisting of seven different chips. Three of the chips were to make a special-purpose CPU with its program stored in ROM and its data stored in shift register read-write memory. Ted Hoff, the Intel engineer assigned to evaluate the project, believed the Busicom design could be simplified by using dynamic RAM storage for data, rather than shift register memory, and a more traditional general-purpose CPU architecture. Hoff came up with a four-chip architectural proposal: a ROM chip for storing the programs, a dynamic RAM chip for storing data, a simple I/O device, and a 4-bit central processing unit (CPU). Although not a chip designer, he felt the CPU could be integrated into a single chip, but as he lacked the technical know-how the idea remained just a wish for the time being.
First microprocessor by Intel, the 4004
While the architecture and specifications of the MCS-4 came from the interaction of Hoff with Stanley Mazor, a software engineer reporting to him, and with Busicom engineer Masatoshi Shima,

during 1969, Mazor and Hoff moved on to other projects. In April 1970, Intel hired Italian engineer Federico Faggin as project leader, a move that ultimately made the single-chip CPU final design a reality (Shima meanwhile designed the Busicom calculator firmware and assisted Faggin during the first six months of the implementation). Faggin, who originally developed the silicon gate technology (SGT) in 1968 at Fairchild Semiconductor[50] and designed the world's first commercial integrated circuit using SGT, the Fairchild 3708, had the correct background to lead the project into what would become the first commercial general purpose microprocessor. Since SGT was his very own invention, Faggin also used it to create his new methodology for random logic design that made it possible to implement a single-chip CPU with the proper speed, power dissipation and cost. The manager of Intel's MOS Design Department was Leslie L. Vadász at the time of the MCS-4 development but Vadász's attention was completely focused on the mainstream business of semiconductor memories so he left the leadership and the management of the MCS-4 project to Faggin, who was ultimately responsible for leading the 4004 project to its realization. Production units of the 4004 were first delivered to Busicom in March 1971 and shipped to other customers in late 1971.[citation needed]

8-bit designs

This section and the sections below needs additional citations for verification. Please help improve this article by adding citations to reliable sources in this section and the sections below. Unsourced material may be challenged and removed.
Find sources: "Microprocessor" – news • newspapers • books • scholar • JSTOR (June 2011) (Learn how and when to remove this message)

The Intel 4004 was followed in 1972 by the Intel 8008, intel's first 8-bit microprocessor.[51] The 8008 was not, however, an extension of the 4004 design, but instead the culmination of a separate design project at Intel, arising from a contract with Computer Terminals Corporation, of San Antonio TX, for a chip for a terminal they were designing,[52] the Datapoint 2200—fundamental aspects of the design came not from Intel but from CTC. In 1968, CTC's Vic Poor and Harry Pyle developed the original design for the instruction set and operation of the processor. In 1969, CTC contracted two companies, Intel and Texas Instruments, to make a single-chip implementation, known as the CTC 1201.[53] In late 1970 or early 1971, TI dropped out being unable to make a reliable part. In 1970, with Intel yet to deliver the part, CTC opted to use their own implementation in the Datapoint 2200, using traditional TTL logic instead (thus the first machine to run "8008 code" was not in fact a microprocessor at all and was delivered a year earlier). Intel's version of the 1201 microprocessor arrived in late 1971, but was too late, slow, and required a number of additional support chips. CTC had no interest in using it. CTC had originally contracted Intel for the chip, and would have owed them US$50,000 (equivalent to $376,171 in 2023) for their design work.[53] To avoid paying for a chip they did not want (and could not use), CTC released Intel from their contract and allowed them free use of the design.[53] Intel marketed it as the 8008 in April, 1972, as the world's first 8-bit microprocessor. It was the basis for the famous "Mark-8" computer kit advertised in the magazine Radio-Electronics in 1974. This processor had an 8-bit data bus and a 14-bit address bus.[54]
The 8008 was the precursor to the successful Intel 8080 (1974), which offered improved performance over the 8008 and required fewer support chips. Federico Faggin conceived and designed it using high voltage N channel MOS. The Zilog Z80 (1976) was also a Faggin design, using low voltage N channel with depletion load and derivative Intel 8-bit processors: all designed with the methodology Faggin created for the 4004. Motorola released the competing 6800 in August 1974, and the similar MOS Technology 6502 was released in 1975 (both designed largely by the same people). The 6502 family rivaled the Z80 in popularity during the 1980s.
A low overall cost, little packaging, simple computer bus requirements, and sometimes the integration of extra circuitry (e.g. the Z80's built-in memory refresh circuitry) allowed the home computer "revolution" to accelerate sharply in the early 1980s. This delivered such inexpensive machines as the Sinclair ZX81, which sold for US$99 (equivalent to $331.79 in 2023). A variation of the 6502, the MOS Technology 6510 was used in the Commodore 64 and yet another variant, the 8502, powered the Commodore 128.

The Western Design Center, Inc (WDC) introduced the CMOS WDC 65C02 in 1982 and licensed the design to several firms. It was used as the CPU in the Apple IIe and IIc personal computers as well as in medical implantable grade pacemakers and defibrillators, automotive, industrial and consumer devices. WDC pioneered the licensing of microprocessor designs, later followed by ARM (32-bit) and other microprocessor intellectual property (IP) providers in the 1990s.
Motorola introduced the MC6809 in 1978. It was an ambitious and well thought-through 8-bit design that was source compatible with the 6800, and implemented using purely hard-wired logic (subsequent 16-bit microprocessors typically used microcode to some extent, as CISC design requirements were becoming too complex for pure hard-wired logic).
Another early 8-bit microprocessor was the Signetics 2650, which enjoyed a brief surge of interest due to its innovative and powerful instruction set architecture.
A seminal microprocessor in the world of spaceflight was RCA's RCA 1802 (aka CDP1802, RCA COSMAC) (introduced in 1976), which was used on board the Galileo probe to Jupiter (launched 1989, arrived 1995). RCA COSMAC was the first to implement CMOS technology. The CDP1802 was used because it could be run at very low power, and because a variant was available fabricated using a special production process, silicon on sapphire (SOS), which provided much better protection against cosmic radiation and electrostatic discharge than that of any other processor of the era. Thus, the SOS version of the 1802 was said to be the first radiation-hardened microprocessor.
The RCA 1802 had a static design, meaning that the clock frequency could be made arbitrarily low, or even stopped. This let the Galileo spacecraft use minimum electric power for long uneventful stretches of a voyage. Timers or sensors would awaken the processor in time for important tasks, such as navigation updates, attitude control, data acquisition, and radio communication. Current versions of the Western Design Center 65C02 and 65C816 also have static cores, and thus retain data even when the clock is completely halted.
12-bit designs
The Intersil 6100 family consisted of a 12-bit microprocessor (the 6100) and a range of peripheral support and memory ICs. The microprocessor recognised the DEC PDP-8 minicomputer instruction set. As such it was sometimes referred to as the CMOS-PDP8. Since it was also produced by Harris Corporation, it was also known as the Harris HM-6100. By virtue of its CMOS technology and associated benefits, the 6100 was being incorporated into some military designs until the early 1980s.
16-bit designs
Part of a series on
Microprocessor modes for the x86 architecture

- **Real mode (Intel 8086)**
- **8080 emulation mode (NEC V20/V30 only)**
- **Protected mode (Intel 80286)**
- **Unreal mode (Intel 80286)**
- **Virtual 8086 mode (Intel 80386)**
- **System Management Mode (Intel 386SL)**
- **Long mode (AMD Athlon 64)**
- **x86 virtualization (Intel Pentium 4, AMD Athlon 64)**
- **AIS mode (VIA C3 only)**

First supported platform shown in parentheses
- **v**
- **t**
- **e**

The first multi-chip 16-bit microprocessor was the National Semiconductor IMP-16, introduced in early 1973. An 8-bit version of the chipset was introduced in 1974 as the IMP-8.
Other early multi-chip 16-bit microprocessors include the MCP-1600 that Digital Equipment

Corporation (DEC) used in the LSI-11 OEM board set and the packaged PDP-11/03 minicomputer—and the Fairchild Semiconductor MicroFlame 9440, both introduced in 1975–76. In late 1974, National introduced the first 16-bit single-chip microprocessor, the National Semiconductor PACE, [55] which was later followed by an NMOS version, the INS8900.
Next in list is the General Instrument CP1600, released in February 1975,[56] which was used mainly in the Intellivision console.
Another early single-chip 16-bit microprocessor was TI's TMS 9900, which was also compatible with their TI-990 line of minicomputers. The 9900 was used in the TI 990/4 minicomputer, the TI-99/4A home computer, and the TM990 line of OEM microcomputer boards. The chip was packaged in a large ceramic 64-pin DIP package, while most 8-bit microprocessors such as the Intel 8080 used the more common, smaller, and less expensive plastic 40-pin DIP. A follow-on chip, the TMS 9980, was designed to compete with the Intel 8080, had the full TI 990 16-bit instruction set, used a plastic 40-pin package, moved data 8 bits at a time, but could only address 16 KB. A third chip, the TMS 9995, was a new design. The family later expanded to include the 99105 and 99110.
The Western Design Center (WDC) introduced the CMOS 65816 16-bit upgrade of the WDC CMOS 65C02 in 1984. The 65816 16-bit microprocessor was the core of the Apple IIGS and later the Super Nintendo Entertainment System, making it one of the most popular 16-bit designs of all time.
Intel "upsized" their 8080 design into the 16-bit Intel 8086, the first member of the x86 family, which powers most modern PC type computers. Intel introduced the 8086 as a cost-effective way of porting software from the 8080 lines, and succeeded in winning much business on that premise. The 8088, a version of the 8086 that used an 8-bit external data bus, was the microprocessor in the first IBM PC. Intel then released the 80186 and 80188, the 80286 and, in 1985, the 32-bit 80386, cementing their PC market dominance with the processor family's backwards compatibility. The 80186 and 80188 were essentially versions of the 8086 and 8088, enhanced with some onboard peripherals and a few new instructions. Although Intel's 80186 and 80188 were not used in IBM PC type designs,[dubious – discuss] second source versions from NEC, the V20 and V30 frequently were. The 8086 and successors had an innovative but limited method of memory segmentation, while the 80286 introduced a full-featured segmented memory management unit (MMU). The 80386 introduced a flat 32-bit memory model with paged memory management.
The 16-bit Intel x86 processors up to and including the 80386 do not include floating-point units (FPUs). Intel introduced the 8087, 80187, 80287 and 80387 math coprocessors to add hardware floating-point and transcendental function capabilities to the 8086 through 80386 CPUs. The 8087 works with the 8086/8088 and 80186/80188,[57] the 80187 works with the 80186 but not the 80188, [58] the 80287 works with the 80286 and the 80387 works with the 80386. The combination of an x86 CPU and an x87 coprocessor forms a single multi-chip microprocessor; the two chips are programmed as a unit using a single integrated instruction set.[59] The 8087 and 80187 coprocessors are connected in parallel with the data and address buses of their parent processor and directly execute instructions intended for them. The 80287 and 80387 coprocessors are interfaced to the CPU through I/O ports in the CPU's address space, this is transparent to the program, which does not need to know about or access these I/O ports directly; the program accesses the coprocessor and its registers through normal instruction opcodes.
32-bit designs
Upper interconnect layers on an Intel 80486DX2 die
16-bit designs had only been on the market briefly when 32-bit implementations started to appear. The most significant of the 32-bit designs is the Motorola MC68000, introduced in 1979. The 68k, as it was widely known, had 32-bit registers in its programming model but used 16-bit internal data paths, three 16-bit Arithmetic Logic Units, and a 16-bit external data bus (to reduce pin count), and externally supported only 24-bit addresses (internally it worked with full 32 bit addresses). In PC-based IBM-compatible mainframes the MC68000 internal microcode was modified to emulate the 32-bit System/370 IBM mainframe.[60] Motorola generally described it as a 16-bit processor. The combination of high performance, large (16 megabytes or 224 bytes) memory space and fairly low cost made it the most popular CPU design of its class. The Apple Lisa and Macintosh designs made use of the 68000, as did other designs in the mid-1980s, including the Atari ST and Amiga.
The world's first single-chip fully 32-bit microprocessor, with 32-bit data paths, 32-bit buses, and 32-

bit addresses, was the AT&T Bell Labs BELLMAC-32A, with first samples in 1980, and general production in 1982.[61][62] After the divestiture of AT&T in 1984, it was renamed the WE 32000 (WE for Western Electric), and had two follow-on generations, the WE 32100 and WE 32200. These microprocessors were used in the AT&T 3B5 and 3B15 minicomputers; in the 3B2, the world's first desktop super microcomputer; in the "Companion", the world's first 32-bit laptop computer; and in "Alexander", the world's first book-sized super microcomputer, featuring ROM-pack memory cartridges similar to today's gaming consoles. All these systems ran the UNIX System V operating system.
The first commercial, single chip, fully 32-bit microprocessor available on the market was the HP FOCUS.
Intel's first 32-bit microprocessor was the iAPX 432, which was introduced in 1981, but was not a commercial success. It had an advanced capability-based object-oriented architecture, but poor performance compared to contemporary architectures such as Intel's own 80286 (introduced 1982), which was almost four times as fast on typical benchmark tests. However, the results for the iAPX432 was partly due to a rushed and therefore suboptimal Ada compiler.[citation needed]
Motorola's success with the 68000 led to the MC68010, which added virtual memory support. The MC68020, introduced in 1984 added full 32-bit data and address buses. The 68020 became hugely popular in the Unix supermicrocomputer market, and many small companies (e.g., Altos, Charles River Data Systems, Cromemco) produced desktop-size systems. The MC68030 was introduced next, improving upon the previous design by integrating the MMU into the chip. The continued success led to the MC68040, which included an FPU for better math performance. The 68050 failed to achieve its performance goals and was not released, and the follow-up MC68060 was released into a market saturated by much faster RISC designs. The 68k family faded from use in the early 1990s.
Other large companies designed the 68020 and follow-ons into embedded equipment. At one point, there were more 68020s in embedded equipment than there were Intel Pentiums in PCs.[63] The ColdFire processor cores are derivatives of the 68020.
During this time (early to mid-1980s), National Semiconductor introduced a very similar 16-bit pinout, 32-bit internal microprocessor called the NS 16032 (later renamed 32016), the full 32-bit version named the NS 32032. Later, National Semiconductor produced the NS 32132, which allowed two CPUs to reside on the same memory bus with built in arbitration. The NS32016/32 outperformed the MC68000/10, but the NS32332—which arrived at approximately the same time as the MC68020—did not have enough performance. The third generation chip, the NS32532, was different. It had about double the performance of the MC68030, which was released around the same time. The appearance of RISC processors like the AM29000 and MC88000 (now both dead) influenced the architecture of the final core, the NS32764. Technically advanced—with a superscalar RISC core, 64-bit bus, and internally overclocked—it could still execute Series 32000 instructions through real-time translation.
When National Semiconductor decided to leave the Unix market, the chip was redesigned into the Swordfish Embedded processor with a set of on-chip peripherals. The chip turned out to be too expensive for the laser printer market and was killed. The design team went to Intel and there designed the Pentium processor, which is very similar to the NS32764 core internally. The big success of the Series 32000 was in the laser printer market, where the NS32CG16 with microcoded BitBlt instructions had very good price/performance and was adopted by large companies like Canon. By the mid-1980s, Sequent introduced the first SMP server-class computer using the NS 32032. This was one of the design's few wins, and it disappeared in the late 1980s. The MIPS R2000 (1984) and R3000 (1989) were highly successful 32-bit RISC microprocessors. They were used in high-end workstations and servers by SGI, among others. Other designs included the Zilog Z80000, which arrived too late to market to stand a chance and disappeared quickly.
The ARM first appeared in 1985.[64] This is a RISC processor design, which has since come to dominate the 32-bit embedded systems processor space due in large part to its power efficiency, its licensing model, and its wide selection of system development tools. Semiconductor manufacturers generally license cores and integrate them into their own system on a chip products; only a few such vendors such as Apple are licensed to modify the ARM cores or create

their own. Most cell phones include an ARM processor, as do a wide variety of other products. There are microcontroller-oriented ARM cores without virtual memory support, as well as symmetric multiprocessor (SMP) applications processors with virtual memory.
From 1993 to 2003, the 32-bit x86 architectures became increasingly dominant in desktop, laptop, and server markets, and these microprocessors became faster and more capable. Intel had licensed early versions of the architecture to other companies, but declined to license the Pentium, so AMD and Cyrix built later versions of the architecture based on their own designs. During this span, these processors increased in complexity (transistor count) and capability (instructions/second) by at least three orders of magnitude. Intel's Pentium line is probably the most famous and recognizable 32-bit processor model, at least with the public at broad.
64-bit designs in personal computers
While 64-bit microprocessor designs have been in use in several markets since the early 1990s (including the Nintendo 64 gaming console in 1996), the early 2000s saw the introduction of 64-bit microprocessors targeted at the PC market.
With AMD's introduction of a 64-bit architecture backwards-compatible with x86, x86-64 (also called AMD64), in September 2003, followed by Intel's near fully compatible 64-bit extensions (first called IA-32e or EM64T, later renamed Intel 64), the 64-bit desktop era began. Both versions can run 32-bit legacy applications without any performance penalty as well as new 64-bit software. With operating systems Windows XP x64, Windows Vista x64, Windows 7 x64, Linux, BSD, and macOS that run 64-bit natively, the software is also geared to fully utilize the capabilities of such processors. The move to 64 bits is more than just an increase in register size from the IA-32 as it also doubles the number of general-purpose registers.
The move to 64 bits by PowerPC had been intended since the architecture's design in the early 90s and was not a major cause of incompatibility. Existing integer registers are extended as are all related data pathways, but, as was the case with IA-32, both floating-point and vector units had been operating at or above 64 bits for several years. Unlike what happened when IA-32 was extended to x86-64, no new general purpose registers were added in 64-bit PowerPC, so any performance gained when using the 64-bit mode for applications making no use of the larger address space is minimal.[citation needed]
In 2011, ARM introduced the new 64-bit ARM architecture.
RISC
Main article: Reduced instruction set computer
In the mid-1980s to early 1990s, a crop of new high-performance reduced instruction set computer (RISC) microprocessors appeared, influenced by discrete RISC-like CPU designs such as the IBM 801 and others. RISC microprocessors were initially used in special-purpose machines and Unix workstations, but then gained wide acceptance in other roles.
The first commercial RISC microprocessor design was released in 1984, by MIPS Computer Systems, the 32-bit R2000 (the R1000 was not released). In 1986, HP released its first system with a PA-RISC CPU. In 1987, in the non-Unix Acorn computers' 32-bit, then cache-less, ARM2-based Acorn Archimedes became the first commercial success using the ARM architecture, then known as Acorn RISC Machine (ARM); first silicon ARM1 in 1985. The R3000 made the design truly practical, and the R4000 introduced the world's first commercially available 64-bit RISC microprocessor. Competing projects would result in the IBM POWER and Sun SPARC architectures. Soon every major vendor was releasing a RISC design, including the AT&T CRISP, AMD 29000, Intel i860 and Intel i960, Motorola 88000, DEC Alpha.
In the late 1990s, only two 64-bit RISC architectures were still produced in volume for non-embedded applications: SPARC and Power ISA, but as ARM has become increasingly powerful, in the early 2010s, it became the third RISC architecture in the general computing segment.
SMP and multi-core design
ABIT BP6 motherboard supported two Intel Celeron 366Mhz processors picture shows Zalman heatsinks. Abit BP6 dual-socket motherboard shown with Zalman Flower heatsinks
SMP symmetric multiprocessing[65] is a configuration of two, four, or more CPU's (in pairs) that are typically used in servers, certain workstations and in desktop personal computers, since the 1990s. A multi-core processor is a single CPU that contains more than one microprocessor core.

This popular two-socket motherboard from Abit was released in 1999 as the first SMP enabled PC motherboard, the Intel Pentium Pro was the first commercial CPU offered to system builders and enthusiasts. The Abit BP9 supports two Intel Celeron CPU's and when used with a SMP enabled operating system (Windows NT/2000/Linux) many applications obtain much higher performance than a single CPU. The early Celerons are easily overclockable and hobbyists used these relatively inexpensive CPU's clocked as high as 533Mhz - far beyond Intel's specification. After discovering the capacity of these motherboards Intel removed access to the multiplier in later CPU's.

In 2001 IBM released the POWER4 CPU, it was a processor that was developed over five years of research, began in 1996 using a team of 250 researchers. The effort to accomplish the impossible was buttressed by development of and through—remote-collaboration and assigning younger engineers to work with more experienced engineers. The teams work achieved success with the new microprocessor, Power4. It is a two-in-one CPU that more than doubled performance at half the price of the competition, and a major advance in computing. The business magazine eWeek wrote: "The newly designed 1GHz Power4 represents a tremendous leap over its predecessor". An industry analyst, Brad Day of Giga Information Group said: "IBM is getting very aggressive, and this server is a game changer".

The Power4 won "Analysts’ Choice Award for Best Workstation/Server Processor of 2001", and it broke notable records, including winning a contest against the best players on the Jeopardy![66] U.S. television show.

Intel's codename Yonah CPU's launched on Jan 6, 2006, and were manufactured with two dies packaged on a multi-chip module. In a hotly-contested marketplace AMD and others released new versions of multi-core CPU's, AMD's SMP enabled Athlon MP CPU's from the AthlonXP line in 2001, Sun released the Niagara and Niagara 2 with eight-cores, AMD's Athlon X2 was released in June 2007. The companies were engaged in a never-ending race for speed, indeed more demanding software mandated more processing power and faster CPU speeds.

By 2012 dual and quad-core processors became widely used in PCs and laptops, newer processors - similar to the higher cost professional level Intel Xeon's - with additional cores that execute instructions in parallel so software performance typically increases, provided the software is designed to utilize advanced hardware. Operating systems provided support for multiple-cores and SMD CPU's, many software applications including large workload and resource intensive applications - such as 3-D games - are programmed to take advantage of multiple core and multi-CPU systems.

Apple, Intel, and AMD currently lead the market with multiple core desktop and workstation CPU's. Although they frequently leapfrog each other for the lead in the performance tier. Intel retains higher frequencies and thus has the fastest single core performance,[67] while AMD is often the leader in multi-threaded routines due to a more advanced ISA and the process node the CPU's are fabricated on.

Multiprocessing concepts for multi-core/multi-cpu configurations are related to Amdahl's law.

Market statistics

In 1997, about 55% of all CPUs sold in the world were 8-bit microcontrollers, of which over 2 billion were sold.[68]

In 2002, less than 10% of all the CPUs sold in the world were 32-bit or more. Of all the 32-bit CPUs sold, about 2% are used in desktop or laptop personal computers. Most microprocessors are used in embedded control applications such as household appliances, automobiles, and computer peripherals. Taken as a whole, the average price for a microprocessor, microcontroller, or DSP is just over US$6 (equivalent to $10.16 in 2023).[69]

In 2003, about $44 billion (equivalent to about $73 billion in 2023) worth of microprocessors were manufactured and sold.[70] Although about half of that money was spent on CPUs used in desktop or laptop personal computers, those count for only about 2% of all CPUs sold.[69] The quality-adjusted price of laptop microprocessors improved −25% to −35% per year in 2004–2010, and the rate of improvement slowed to −15% to −25% per year in 2010–2013.[71]

About 10 billion CPUs were manufactured in 2008. Most new CPUs produced each year are embedded.[72]

See also

- **Comparison of instruction set architectures**
- **Computer architecture**
- **Computer engineering**
- **Heterogeneous computing**
- **List of microprocessors**
- **Microarchitecture**
- **Microprocessor chronology**

Notes

1.
• Orion, Veritas (23 August 2024). "What distinguishes a microprocessor from a microcontroller?". Ampheo Electronics. Orion Veritas.
• • "The Surprising Story of the First Microprocessors". 30 August 2016. Archived from the original on 4 October 2022. Retrieved 4 October 2022.
• • Warnes, Lionel (2003). "Microprocessors and microcontrollers". Electronic and Electrical Engineering. London: Macmillan Education UK. pp. 443–477. doi:10.1007/978-0-230-21633-4_23. ISBN 978-0-333-99040-7. microprocessor is not a stand-alone computer, since it lacks memory and input/output control. These are the missing parts that the microcontroller supplies, making it more nearly a complete computer on a chip.
• • Morris, Noel M. (1985). Microelectronic and Microprocessor-based Systems. London: Macmillan Education UK. p. 16. doi:10.1007/978-1-349-06978-1. ISBN 978-0-333-36190-0. A microprocessor itself is incapable of performing calculations and requires a support system in order to do so. The CPU support system includes a storage system in which not only the operating instructions but also the data (operands) are stored.
• • Dyer, Stephen A.; Harms, Brian K. (13 August 1993). "Digital Signal Processing". In Yovits, Marshall C. (ed.). Advances in Computers. Vol. 37. Academic Press. pp. 59–118. doi:10.1016/S0065-2458(08)60403-9. ISBN 978-0120121373. ISSN 0065-2458. LCCN 59015761. OCLC 858439915. OL 10070096M.
• • Liptak, B. G. (2006). Process Control and Optimization. Instrument Engineers' Handbook. Vol. 2 (4th ed.). CRC Press. pp. 11–12. ISBN 978-0849310812 – via Google Books.
• • "FYI: Today's computer chips are so advanced, they are more 'mercurial' than precise – and here's the proof". Archived from the original on 13 February 2024. Retrieved 13 February 2024.
• • CMicrotek. "8-bit vs 32-bit Micros" Archived 2014-07-14 at the Wayback Machine.
• • "Managing the Impact of Increasing Microprocessor Power Consumption" (PDF). Rice University. Archived (PDF) from the original on 3 October 2015. Retrieved 1 October 2015.
• • Wayne Freeman. "11 Myths About 8-Bit Microcontrollers" Archived 12 August 2022 at the Wayback Machine. 2016. quote: "Basically, by getting your work done faster, you can put the CPU in sleep mode for longer periods of time. Thus, 32-bit MCUs are more power-efficient than 8-bit MCUs, right? Wrong."
• • Wilkes, M. V. (1969). "The Growth of Interest in Microprogramming: A Literature Survey". ACM Computing Surveys. 1 (3): 139–145. doi:10.1145/356551.356553. S2CID 10673679.
• • "Sir Maurice Wilkes, The Father Of Computing, Dies". Silicon UK. 3 December 2010. Retrieved 28 November 2023.
• • Shirriff, Ken (30 August 2016). "The Surprising Story of the First Microprocessors". IEEE Spectrum. 53 (9). Institute of Electrical and Electronics Engineers: 48–54. doi:10.1109/MSPEC.2016.7551353. S2CID 32003640. Archived from the original on 24 November 2017. Retrieved 13 October 2019.
• • Laws, David (20 September 2018). "Who Invented the Microprocessor?". Computer History Museum. Archived from the original on 19 January 2024. Retrieved 19 January 2024.
• • "1971: Microprocessor Integrates CPU Function onto a Single Chip". The Silicon Engine. Computer History Museum. Archived from the original on 12 August 2021. Retrieved 22 July 2019.
• • "1968: Silicon Gate Technology Developed for ICs | The Silicon Engine | Computer History Museum". www.computerhistory.org. Archived from the original on 29 July 2020. Retrieved 24 October 2019.
• • "1971: Microprocessor Integrates CPU Function onto a Single Chip | The Silicon Engine |

Computer History Museum". www.computerhistory.org. Archived from the original on 12 August 2021. Retrieved 24 October 2019.
• • Viatron Computer Systems. "System 21 is Now!" Archived 2011-03-21 at the Wayback Machine (PDF).
• • Moore, Gordon (19 April 1965). "Cramming more components onto integrated circuits" (PDF). Electronics. 38 (8). Archived from the original (PDF) on 18 February 2008. Retrieved 23 December 2009.
• • "Excerpts from A Conversation with Gordon Moore: Moore's Law" (PDF). Intel. 2005. Archived from the original (PDF) on 29 October 2012. Retrieved 23 December 2009. {{cite journal}}: Cite journal requires |journal= (help)
• • Basset, Ross (2003). "When is a Microprocessor not a Microprocessor? The Industrial Construction of Semiconductor Innovation". In Finn, Bernard (ed.). Exposing Electronics. Michigan State University Press. p. 121. ISBN 978-0-87013-658-0. Archived from the original on 30 March 2014.
• • "1971 - Microprocessor Integrates CPU Function onto a Single Chip". The Silicon Engine. Computer History Museum. Archived from the original on 8 June 2010. Retrieved 25 July 2010.
• • Shaller, Robert R. (15 April 2004). "Technological Innovation in the Semiconductor Industry: A Case Study of the International Technology Roadmap for Semiconductors" (PDF). George Mason University. Archived (PDF) from the original on 19 December 2006. Retrieved 25 July 2010.
• • RW (3 March 1995). "Interview with Gordon E. Moore". LAIR History of Science and Technology Collections. Los Altos Hills, California: Stanford University. Archived from the original on 4 February 2012.
• • Bassett 2003. pp. 115, 122.
• • "First Microprocessor". First Microprocessor | 50th Anniversary of the Microprocessor 2020. Archived from the original on 6 January 2014.
• • Holt, Ray M. "World's First Microprocessor Chip Set". Ray M. Holt website. Archived from the original on 6 January 2014. Retrieved 25 July 2010.
• • Fallon, Sarah. "The Secret History of the First Microprocessor, the F-14, and Me". Wired. ISSN 1059-1028. Archived from the original on 18 January 2024. Retrieved 21 January 2024.
• • Holt, Ray (27 September 2001). Lecture: Microprocessor Design and Development for the US Navy F14 FighterJet (Speech). Room 8220, Wean Hall, Carnegie Mellon University, Pittsburgh, PA, US. Archived from the original on 1 October 2011. Retrieved 25 July 2010.
• • Parab, Jivan S.; Shelake, Vinod G.; Kamat, Rajanish K.; Naik, Gourish M. (2007). Exploring C for Microcontrollers: A Hands on Approach (PDF). Springer. p. 4. ISBN 978-1-4020-6067-0. Archived (PDF) from the original on 20 July 2011. Retrieved 25 July 2010.
• • Dyer, S. A.; Harms, B. K. (1993). "Digital Signal Processing". In Yovits, M. C. (ed.). Advances in Computers. Vol. 37. Academic Press. pp. 104–107. doi:10.1016/S0065-2458(08)60403-9. ISBN 9780120121373. Archived from the original on 29 December 2016.
• • US 4942516, Hyatt, Gilbert P, "Single chip integrated circuit computer architecture", issued 1990-07-17 Archived 25 May 2012 at the Wayback Machine
• • "Chip Designer's 20-Year Quest : Computers: Gilbert Hyatt's solitary battle to patent the microprocessor appears to have paid off, if it can withstand legal challenges. Here's his story". Los Angeles Times. 21 October 1990. Archived from the original on 4 October 2022. Retrieved 4 October 2022.
• • Markoff, John (20 June 1996). "For Texas Instruments, Some Bragging Rights". The New York Times. Archived from the original on 28 September 2022. Retrieved 4 October 2022.
• • "The Birth of the Microprocessor". Archived from the original on 4 October 2022. Retrieved 4 October 2022.
• • "Microprocessor Patent Holder Signs Contract : Invention: La Palma inventor signs with Dutch electronics giant, the first company to accord validity to his patent". Los Angeles Times. 7 November 1991. Archived from the original on 4 October 2022. Retrieved 4 October 2022.
• • "Inventor's fight for recognition ongoing but not all-consuming - Las Vegas Sun Newspaper". 21 December 2014. Archived from the original on 20 October 2022. Retrieved 4 October 2022.
• • "The Texas Instruments TMX 1795: The (Almost) first, forgotten microprocessor".

• • Seitz, Frederick; Einspruch, Norman G. (1998). "19. The 1970s and the Microprocessor § Texas Instruments". Electronic Genie: The Tangled History of Silicon. University of Illinois Press. pp. 228–9. ISBN 0252023838. Archived from the original on 19 February 2023. Retrieved 14 August 2022.
• • Shirriff, Ken (2016). "The Surprising Story of the First Microprocessors". IEEE Spectrum. 53 (9): 48–54. doi:10.1109/MSPEC.2016.7551353. S2CID 32003640. Archived from the original on 14 August 2022. Retrieved 14 August 2022.
• • U.S. Patent no. 4,074,351 (TMS1802NC.)
• • "STANDARD CALCULATOR ON A CHIP ANNOUNCED BY TEXAS INSTRUMENTS", press release. TI, Sep. 19, 1971. Originally on ti.com but now archived at archive.org.
• • McGonigal, James (20 September 2006). "Microprocessor History: Foundations in Glenrothes, Scotland". McGonigal personal website. Archived from the original on 20 July 2011. Retrieved 23 December 2009.
• • Tout, Nigel. "ANITA at its Zenith". Bell Punch Company and the ANITA calculators. Archived from the original on 11 August 2010. Retrieved 25 July 2010.
• • 16 Bit Microprocessor Handbook by Gerry Kane, Adam Osborne ISBN 0-07-931043-5 (0-07-931043-5)
• • Mack, Pamela E. (30 November 2005). "The Microcomputer Revolution". Archived from the original on 14 January 2010. Retrieved 23 December 2009.
• • "History in the Computing Curriculum" (PDF). Archived from the original (PDF) on 19 July 2011. Retrieved 23 December 2009. {{cite journal}}: Cite journal requires |journal= (help)
• • Bright, Peter (15 November 2011). "The 40th birthday of—maybe—the first microprocessor, the Intel 4004". arstechnica.com. Archived from the original on 6 January 2017.
• • Faggin, Federico; Hoff, Marcian E. Jr.; Mazor, Stanley; Shima, Masatoshi (December 1996). "The History of the 4004". IEEE Micro. 16 (6): 10–20. doi:10.1109/40.546561.
• • Faggin, F.; Klein, T.; Vadasz, L. (23 October 1968). Insulated Gate Field Effect Transistor Integrated Circuits with Silicon Gates (JPEG image). International Electronic Devices Meeting. IEEE Electron Devices Group. Archived from the original on 19 February 2010. Retrieved 23 December 2009.
• • "Intel Microprocessor Quick Reference Guide - Year". www.intel.com. Archived from the original on 6 October 2021. Retrieved 21 September 2021.
• • Ceruzzi, Paul E. (May 2003). A History of Modern Computing (2nd ed.). MIT Press. pp. 220–221. ISBN 978-0-262-53203-7.
• • Wood, Lamont (August 2008). "Forgotten history: the true origins of the PC". Computerworld. Archived from the original on 6 June 2022. Retrieved 7 January 2011.
• • Intel 8008 data sheet.
• • "National Semiconductor PACE CPU family". Archived from the original on 25 November 2022. Retrieved 25 November 2022.
• • EDN Staff (1 January 2000). "General Instrument's microprocessor aimed at minicomputer market". EDN. Archived from the original on 25 November 2022. Retrieved 1 January 2023.
• • Intel 8087 datasheet, pg. 1
• • The 80187 only has a 16-bit data bus because it used the 80387SX core.
• • "Essentially, the 80C187 can be treated as an additional resource or an extension to the CPU. The 80C186 CPU together with an 80C187 can be used as a single unified system." Intel 80C187 datasheet, p. 3, November 1992 (Order Number: 270640-004).
• • "Implementation of IBM System 370 Via Co-Microprocessors/The Co-Processor Interface on priorart.ip.com". priorart.ip.com. 1 January 1986. Archived from the original on 11 December 2015. Retrieved 23 July 2020.
• • "Shoji, M. Bibliography". Bell Laboratories. 7 October 1998. Archived from the original on 16 October 2008. Retrieved 23 December 2009.
• • "Timeline: 1982–1984". Physical Sciences & Communications at Bell Labs. Bell Labs, Alcatel-Lucent. 17 January 2001. Archived from the original on 14 May 2011. Retrieved 23 December 2009.
• • Turley, Jim (July 1998). "MCore: Does Motorola Need Another Processor Family?". Embedded Systems Design. TechInsights (United Business Media). Archived from the original on 2 July 1998. Retrieved 23 December 2009.

• • Garnsey, Elizabeth; Lorenzoni, Gianni; Ferriani, Simone (March 2008). "Speciation through entrepreneurial spin-off: The Acorn-ARM story" (PDF). Research Policy. 37 (2): 210–224. doi:10.1016/j.respol.2007.11.006. S2CID 73520408. Retrieved 2 June 2011. [...] the first silicon was run on April 26th 1985.
• • "Difference Between Symmetric and Asymmetric Multiprocessing (With Comparison Chart)". 22 September 2016. Archived from the original on 18 July 2021. Retrieved 18 July 2021.
• • "IBM100 - A Computer Called Watson". IBM. 7 March 2012. Archived from the original on 19 July 2021. Retrieved 19 July 2021.
• • Tarasov, Katie (22 November 2022). "How AMD became a chip giant and leapfrogged Intel after years of playing catch-up". CNBC. Archived from the original on 1 June 2023. Retrieved 17 May 2023.
• • Cantrell, Tom (1998). "Microchip on the March". Archived from the original on 20 February 2007.
• • Turley, Jim (18 December 2002). "The Two Percent Solution". Embedded Systems Design. TechInsights (United Business Media). Archived from the original on 3 April 2015. Retrieved 23 December 2009.
• • WSTS Board Of Directors. "WSTS Semiconductor Market Forecast World Release Date: 1 June 2004 - 6:00 UTC". Miyazaki, Japan, Spring Forecast Meeting 18–21 May 2004 (Press release). World Semiconductor Trade Statistics. Archived from the original on 7 December 2004.
• • Sun, Liyang (25 April 2014). "What We Are Paying for: A Quality Adjusted Price Index for Laptop Microprocessors". Wellesley College. Archived from the original on 11 November 2014. Retrieved 7 November 2014. … compared with -25% to -35% per year over 2004-2010, the annual decline plateaus around -15% to -25% over 2010-2013.
72. • Barr, Michael (1 August 2009). "Real men program in C". Embedded Systems Design. TechInsights (United Business Media). p. 2. Archived from the original on 22 October 2012. Retrieved 23 December 2009.

References

• Ray, A. K.; Bhurchand, K.M. (2013). Advanced Microprocessors and Peripherals (3rd ed.). India: Tata McGraw-Hill. ISBN 978-1-259-02977-6. OCLC 878079623.

External links

Wikiversity has learning resources about Introduction to Computers/Processor

Wikimedia Commons has media related to Microprocessors.

• Patent problems
• Dirk Oppelt. "The CPU Collection". Retrieved 23 December 2009.
• Gennadiy Shvets. "CPU-World". Retrieved 23 December 2009.
• Jérôme Cremet. "The Gecko's CPU Library". Retrieved 23 December 2009.
• "How Microprocessors Work". April 2000. Retrieved 23 December 2009.
• William Blair. "IC Die Photography". Retrieved 23 December 2009.
• John Bayko (December 2003). "Great Microprocessors of the Past and Present". Archived from the original on 15 April 2013. Retrieved 23 December 2009.
• Wade Warner (22 December 2004). "Great moments in microprocessor history". IBM. Retrieved 7 March 2013.
• Arthur Daemmrich and Sharon Klotz (12 December 2019). "Gilbert Hyatt" (PDF). Smithsonian Institution.

• v
• t
• e

Processor technologies

Authority control databases

Categories:
- **Microprocessors**
- **American inventions**
- **Japanese inventions**
- **Digital electronics**
- **Microcomputers**
- **Telecommunications engineering**
- **1971 introductions**
- **This page was last edited on 26 November 2024, at 12:18 (UTC).**

Contents

• •
Further examples

• •
History

• •
Mathematical construction

• •
Application

• •
See also

•
Notes

•
References

•
Further reading

•
•
External links

Stochastic process

- **Article**
- **Talk**
- **Read**
- **Edit**
- **View history**

Tools
•
•
•
•
•
•
•
•
•
•
•
•
•
•
Appearance
Text
•

Small
Standard
Large
Width
•
Standard
Wide
Color (beta)
•
Automatic
Light
Dark
From Wikipedia, the free encyclopedia
Part of a series on statistics

Probability theory

• Probability
o Axioms
• Determinism
o System
• Indeterminism
• Randomness

• Probability space
• Sample space
• Event
o Collectively exhaustive events
o Elementary event
o Mutual exclusivity
o Outcome
o Singleton
• Experiment
o Bernoulli trial
• Probability distribution
o Bernoulli distribution
o Binomial distribution
o Exponential distribution
o Normal distribution
o Pareto distribution
o Poisson distribution
• Probability measure
• Random variable
o Bernoulli process
o Continuous or discrete
o Expected value
o Variance
o Markov chain
o Observed value
o Random walk
o Stochastic process
• Complementary event

- **Joint probability**
- **Marginal probability**
- **Conditional probability**

- **Independence**
- **Conditional independence**
- **Law of total probability**
- **Law of large numbers**
- **Bayes' theorem**
- **Boole's inequality**

- **Venn diagram**
- **Tree diagram**

- **v**
- **t**
- **e**

A computer-simulated realization of a Wiener or Brownian motion process on the surface of a sphere. The Wiener process is widely considered the most studied and central stochastic process in probability theory.[1][2][3]

In probability theory and related fields, a stochastic (/stəˈkæstɪk/) or random process is a mathematical object usually defined as a family of random variables in a probability space, where the index of the family often has the interpretation of time. Stochastic processes are widely used as mathematical models of systems and phenomena that appear to vary in a random manner. Examples include the growth of a bacterial population, an electrical current fluctuating due to thermal noise, or the movement of a gas molecule.[1][4][5] Stochastic processes have applications in many disciplines such as biology,[6] chemistry,[7] ecology,[8] neuroscience,[9] physics,[10] image processing, signal processing,[11] control theory,[12] information theory,[13] computer science,[14] and telecommunications.[15] Furthermore, seemingly random changes in financial markets have motivated the extensive use of stochastic processes in finance.[16][17][18]

Applications and the study of phenomena have in turn inspired the proposal of new stochastic processes. Examples of such stochastic processes include the Wiener process or Brownian motion process,[a] used by Louis Bachelier to study price changes on the Paris Bourse,[21] and the Poisson process, used by A. K. Erlang to study the number of phone calls occurring in a certain period of time.[22] These two stochastic processes are considered the most important and central in the theory of stochastic processes,[1][4][23] and were invented repeatedly and independently, both before and after Bachelier and Erlang, in different settings and countries.[21][24]

The term random function is also used to refer to a stochastic or random process,[25][26] because a stochastic process can also be interpreted as a random element in a function space.[27][28] The terms stochastic process and random process are used interchangeably, often with no specific mathematical space for the set that indexes the random variables.[27][29] But often these two terms are used when the random variables are indexed by the integers or an interval of the real line.[5][29] If the random variables are indexed by the Cartesian plane or some higher-dimensional Euclidean space, then the collection of random variables is usually called a random field instead.[5][30] The values of a stochastic process are not always numbers and can be vectors or other mathematical objects.[5][28]

Based on their mathematical properties, stochastic processes can be grouped into various categories, which include random walks,[31] martingales,[32] Markov processes,[33] Lévy processes,[34] Gaussian processes,[35] random fields,[36] renewal processes, and branching processes.[37] The study of stochastic processes uses mathematical knowledge and techniques from probability, calculus, linear algebra, set theory, and topology[38][39][40] as well as branches of mathematical analysis such as real analysis, measure theory, Fourier analysis, and functional analysis.[41][42][43] The theory of stochastic processes is considered to be an important

contribution to mathematics[44] and it continues to be an active topic of research for both theoretical reasons and applications.[45][46][47]

Introduction

A stochastic or random process can be defined as a collection of random variables that is indexed by some mathematical set, meaning that each random variable of the stochastic process is uniquely associated with an element in the set.[4][5] The set used to index the random variables is called the index set. Historically, the index set was some subset of the real line, such as the natural numbers, giving the index set the interpretation of time.[1] Each random variable in the collection takes values from the same mathematical space known as the state space. This state space can be, for example, the integers, the real line or -dimensional Euclidean space.[1][5] An increment is the amount that a stochastic process changes between two index values, often interpreted as two points in time.[48][49] A stochastic process can have many outcomes, due to its randomness, and a single outcome of a stochastic process is called, among other names, a sample function or realization.[28][50]

A single computer-simulated sample function or realization, among other terms, of a three-dimensional Wiener or Brownian motion process for time $0 \leq t \leq 2$. The index set of this stochastic process is the non-negative numbers, while its state space is three-dimensional Euclidean space.

Classifications

A stochastic process can be classified in different ways, for example, by its state space, its index set, or the dependence among the random variables. One common way of classification is by the cardinality of the index set and the state space.[51][52][53]

When interpreted as time, if the index set of a stochastic process has a finite or countable number of elements, such as a finite set of numbers, the set of integers, or the natural numbers, then the stochastic process is said to be in discrete time.[54][55] If the index set is some interval of the real line, then time is said to be continuous. The two types of stochastic processes are respectively referred to as discrete-time and continuous-time stochastic processes.[48][56][57] Discrete-time stochastic processes are considered easier to study because continuous-time processes require more advanced mathematical techniques and knowledge, particularly due to the index set being uncountable.[58][59] If the index set is the integers, or some subset of them, then the stochastic process can also be called a random sequence.[55]

If the state space is the integers or natural numbers, then the stochastic process is called a discrete or integer-valued stochastic process. If the state space is the real line, then the stochastic process is referred to as a real-valued stochastic process or a process with continuous state space. If the state space is -dimensional Euclidean space, then the stochastic process is called a -dimensional vector process or -vector process.[51][52]

Etymology

The word stochastic in English was originally used as an adjective with the definition "pertaining to conjecturing", and stemming from a Greek word meaning "to aim at a mark, guess", and the Oxford English Dictionary gives the year 1662 as its earliest occurrence.[60] In his work on probability Ars Conjectandi, originally published in Latin in 1713, Jakob Bernoulli used the phrase "Ars Conjectandi sive Stochastice", which has been translated to "the art of conjecturing or stochastics".[61] This phrase was used, with reference to Bernoulli, by Ladislaus Bortkiewicz[62] who in 1917 wrote in German the word stochastik with a sense meaning random. The term stochastic process first appeared in English in a 1934 paper by Joseph Doob.[60] For the term and a specific mathematical definition, Doob cited another 1934 paper, where the term stochastischer Prozeß was used in German by Aleksandr Khinchin,[63][64] though the German term had been used earlier, for example, by Andrei Kolmogorov in 1931.[65]

According to the Oxford English Dictionary, early occurrences of the word random in English with its current meaning, which relates to chance or luck, date back to the 16th century, while earlier recorded usages started in the 14th century as a noun meaning "impetuosity, great speed, force, or violence (in riding, running, striking, etc.)". The word itself comes from a Middle French word meaning "speed, haste", and it is probably derived from a French verb meaning "to run" or "to gallop". The first written appearance of the term random process pre-dates stochastic process, which the Oxford English Dictionary also gives as a synonym, and was used in an article by Francis

Edgeworth published in 1888.[66]
Terminology
The definition of a stochastic process varies,[67] but a stochastic process is traditionally defined as a collection of random variables indexed by some set.[68][69] The terms random process and stochastic process are considered synonyms and are used interchangeably, without the index set being precisely specified.[27][29][30][70][71][72] Both "collection",[28][70] or "family" are used[4][73] while instead of "index set", sometimes the terms "parameter set"[28] or "parameter space"[30] are used.
The term random function is also used to refer to a stochastic or random process,[5][74][75] though sometimes it is only used when the stochastic process takes real values.[28][73] This term is also used when the index sets are mathematical spaces other than the real line,[5][76] while the terms stochastic process and random process are usually used when the index set is interpreted as time,[5][76][77] and other terms are used such as random field when the index set is -dimensional Euclidean space or a manifold.[5][28][30]
Notation
A stochastic process can be denoted, among other ways, by ,[56] ,[69] [78] or simply as . Some authors mistakenly write even though it is an abuse of function notation.[79] For example, or are used to refer to the random variable with the index , and not the entire stochastic process.[78] If the index set is , then one can write, for example, to denote the stochastic process.[29]
Examples
Bernoulli process
Main article: Bernoulli process
One of the simplest stochastic processes is the Bernoulli process,[80] which is a sequence of independent and identically distributed (iid) random variables, where each random variable takes either the value one or zero, say one with probability and zero with probability . This process can be linked to an idealisation of repeatedly flipping a coin, where the probability of obtaining a head is taken to be and its value is one, while the value of a tail is zero.[81] In other words, a Bernoulli process is a sequence of iid Bernoulli random variables,[82] where each idealised coin flip is an example of a Bernoulli trial.[83]
Random walk
Main article: Random walk
Random walks are stochastic processes that are usually defined as sums of iid random variables or random vectors in Euclidean space, so they are processes that change in discrete time.[84][85][86][87][88] But some also use the term to refer to processes that change in continuous time,[89] particularly the Wiener process used in financial models, which has led to some confusion, resulting in its criticism.[90] There are various other types of random walks, defined so their state spaces can be other mathematical objects, such as lattices and groups, and in general they are highly studied and have many applications in different disciplines.[89][91]
A classic example of a random walk is known as the simple random walk, which is a stochastic process in discrete time with the integers as the state space, and is based on a Bernoulli process, where each Bernoulli variable takes either the value positive one or negative one. In other words, the simple random walk takes place on the integers, and its value increases by one with probability, say, , or decreases by one with probability , so the index set of this random walk is the natural numbers, while its state space is the integers. If , this random walk is called a symmetric random walk.[92][93]
Wiener process
Main article: Wiener process
The Wiener process is a stochastic process with stationary and independent increments that are normally distributed based on the size of the increments.[2][94] The Wiener process is named after Norbert Wiener, who proved its mathematical existence, but the process is also called the Brownian motion process or just Brownian motion due to its historical connection as a model for Brownian movement in liquids.[95][96][97]
Realizations of Wiener processes (or Brownian motion processes) with drift (blue) and without drift (red)

Playing a central role in the theory of probability, the Wiener process is often considered the most important and studied stochastic process, with connections to other stochastic processes.[1][2][3][98][99][100][101] Its index set and state space are the non-negative numbers and real numbers, respectively, so it has both continuous index set and states space.[102] But the process can be defined more generally so its state space can be -dimensional Euclidean space.[91][99][103] If the mean of any increment is zero, then the resulting Wiener or Brownian motion process is said to have zero drift. If the mean of the increment for any two points in time is equal to the time difference multiplied by some constant , which is a real number, then the resulting stochastic process is said to have drift .[104][105][106]
Almost surely, a sample path of a Wiener process is continuous everywhere but nowhere differentiable. It can be considered as a continuous version of the simple random walk.[49][105] The process arises as the mathematical limit of other stochastic processes such as certain random walks rescaled,[107][108] which is the subject of Donsker's theorem or invariance principle, also known as the functional central limit theorem.[109][110][111]
The Wiener process is a member of some important families of stochastic processes, including Markov processes, Lévy processes and Gaussian processes.[2][49] The process also has many applications and is the main stochastic process used in stochastic calculus.[112][113] It plays a central role in quantitative finance,[114][115] where it is used, for example, in the Black–Scholes–Merton model.[116] The process is also used in different fields, including the majority of natural sciences as well as some branches of social sciences, as a mathematical model for various random phenomena.[3][117][118]
Poisson process
Main article: Poisson process
The Poisson process is a stochastic process that has different forms and definitions.[119][120] It can be defined as a counting process, which is a stochastic process that represents the random number of points or events up to some time. The number of points of the process that are located in the interval from zero to some given time is a Poisson random variable that depends on that time and some parameter. This process has the natural numbers as its state space and the non-negative numbers as its index set. This process is also called the Poisson counting process, since it can be interpreted as an example of a counting process.[119]
If a Poisson process is defined with a single positive constant, then the process is called a homogeneous Poisson process.[119][121] The homogeneous Poisson process is a member of important classes of stochastic processes such as Markov processes and Lévy processes.[49]
The homogeneous Poisson process can be defined and generalized in different ways. It can be defined such that its index set is the real line, and this stochastic process is also called the stationary Poisson process.[122][123] If the parameter constant of the Poisson process is replaced with some non-negative integrable function of , the resulting process is called an inhomogeneous or nonhomogeneous Poisson process, where the average density of points of the process is no longer constant.[124] Serving as a fundamental process in queueing theory, the Poisson process is an important process for mathematical models, where it finds applications for models of events randomly occurring in certain time windows.[125][126]
Defined on the real line, the Poisson process can be interpreted as a stochastic process,[49][127] among other random objects.[128][129] But then it can be defined on the -dimensional Euclidean space or other mathematical spaces,[130] where it is often interpreted as a random set or a random counting measure, instead of a stochastic process.[128][129] In this setting, the Poisson process, also called the Poisson point process, is one of the most important objects in probability theory, both for applications and theoretical reasons.[22][131] But it has been remarked that the Poisson process does not receive as much attention as it should, partly due to it often being considered just on the real line, and not on other mathematical spaces.[131][132]
Definitions
Stochastic process
A stochastic process is defined as a collection of random variables defined on a common probability space , where is a sample space, is a -algebra, and is a probability measure; and the random variables, indexed by some set , all take values in the same mathematical space , which

must be measurable with respect to some -algebra .[28]
In other words, for a given probability space and a measurable space , a stochastic process is a collection of -valued random variables, which can be written as:[80]

Historically, in many problems from the natural sciences a point had the meaning of time, so is a random variable representing a value observed at time .[133] A stochastic process can also be written as to reflect that it is actually a function of two variables, and .[28][134]
There are other ways to consider a stochastic process, with the above definition being considered the traditional one.[68][69] For example, a stochastic process can be interpreted or defined as a -valued random variable, where is the space of all the possible functions from the set into the space .[27][68] However this alternative definition as a "function-valued random variable" in general requires additional regularity assumptions to be well-defined.[135]
Index set
The set is called the index set[4][51] or parameter set[28][136] of the stochastic process. Often this set is some subset of the real line, such as the natural numbers or an interval, giving the set the interpretation of time.[1] In addition to these sets, the index set can be another set with a total order or a more general set,[1][54] such as the Cartesian plane or -dimensional Euclidean space, where an element can represent a point in space.[48][137] That said, many results and theorems are only possible for stochastic processes with a totally ordered index set.[138]
State space
The mathematical space of a stochastic process is called its state space. This mathematical space can be defined using integers, real lines, -dimensional Euclidean spaces, complex planes, or more abstract mathematical spaces. The state space is defined using elements that reflect the different values that the stochastic process can take.[1][5][28][51][56]
Sample function
A sample function is a single outcome of a stochastic process, so it is formed by taking a single possible value of each random variable of the stochastic process.[28][139] More precisely, if is a stochastic process, then for any point , the mapping

is called a sample function, a realization, or, particularly when is interpreted as time, a sample path of the stochastic process .[50] This means that for a fixed , there exists a sample function that maps the index set to the state space .[28] Other names for a sample function of a stochastic process include trajectory, path function[140] or path.[141]
Increment
An increment of a stochastic process is the difference between two random variables of the same stochastic process. For a stochastic process with an index set that can be interpreted as time, an increment is how much the stochastic process changes over a certain time period. For example, if is a stochastic process with state space and index set , then for any two non-negative numbers and such that , the difference is a -valued random variable known as an increment.[48][49] When interested in the increments, often the state space is the real line or the natural numbers, but it can be -dimensional Euclidean space or more abstract spaces such as Banach spaces.[49]
Further definitions
Law
For a stochastic process defined on the probability space , the law of stochastic process is defined as the image measure:

where is a probability measure, the symbol denotes function composition and is the pre-image of the measurable function or, equivalently, the -valued random variable , where is the space of all the possible -valued functions of , so the law of a stochastic process is a probability measure.[27][68][142][143]
For a measurable subset of , the pre-image of gives

so the law of a can be written as:[28]

The law of a stochastic process or a random variable is also called the probability law, probability distribution, or the distribution.[133][142][144][145][146]
Finite-dimensional probability distributions
Main article: Finite-dimensional distribution
For a stochastic process with law , its finite-dimensional distribution for is defined as:

This measure is the joint distribution of the random vector ; it can be viewed as a "projection" of the law onto a finite subset of .[27][147]
For any measurable subset of the -fold Cartesian power , the finite-dimensional distributions of a stochastic process can be written as:[28]

The finite-dimensional distributions of a stochastic process satisfy two mathematical conditions known as consistency conditions.[57]
Stationarity
Main article: Stationary process
Stationarity is a mathematical property that a stochastic process has when all the random variables of that stochastic process are identically distributed. In other words, if is a stationary stochastic process, then for any the random variable has the same distribution, which means that for any set of index set values , the corresponding random variables

all have the same probability distribution. The index set of a stationary stochastic process is usually interpreted as time, so it can be the integers or the real line.[148][149] But the concept of stationarity also exists for point processes and random fields, where the index set is not interpreted as time.[148][150][151]
When the index set can be interpreted as time, a stochastic process is said to be stationary if its finite-dimensional distributions are invariant under translations of time. This type of stochastic process can be used to describe a physical system that is in steady state, but still experiences random fluctuations.[148] The intuition behind stationarity is that as time passes the distribution of the stationary stochastic process remains the same.[152] A sequence of random variables forms a stationary stochastic process only if the random variables are identically distributed.[148]
A stochastic process with the above definition of stationarity is sometimes said to be strictly stationary, but there are other forms of stationarity. One example is when a discrete-time or continuous-time stochastic process is said to be stationary in the wide sense, then the process has a finite second moment for all and the covariance of the two random variables and depends only on the number for all .[152][153] Khinchin introduced the related concept of stationarity in the wide sense, which has other names including covariance stationarity or stationarity in the broad sense. [153][154]
Filtration
A filtration is an increasing sequence of sigma-algebras defined in relation to some probability space and an index set that has some total order relation, such as in the case of the index set being some subset of the real numbers. More formally, if a stochastic process has an index set with a total order, then a filtration , on a probability space is a family of sigma-algebras such that for all , where and denotes the total order of the index set .[51] With the concept of a filtration, it is possible to study the amount of information contained in a stochastic process at , which can be interpreted as time .[51][155] The intuition behind a filtration is that as time passes, more and more information on is known or available, which is captured in , resulting in finer and finer partitions of .[156][157]
Modification
A modification of a stochastic process is another stochastic process, which is closely related to the original stochastic process. More precisely, a stochastic process that has the same index set , state space , and probability space as another stochastic process is said to be a modification of if for all the following

holds. Two stochastic processes that are modifications of each other have the same finite-dimensional law[158] and they are said to be stochastically equivalent or equivalent.[159]

Instead of modification, the term version is also used,[150][160][161][162] however some authors use the term version when two stochastic processes have the same finite-dimensional distributions, but they may be defined on different probability spaces, so two processes that are modifications of each other, are also versions of each other, in the latter sense, but not the converse.[163][142]
If a continuous-time real-valued stochastic process meets certain moment conditions on its increments, then the Kolmogorov continuity theorem says that there exists a modification of this process that has continuous sample paths with probability one, so the stochastic process has a continuous modification or version.[161][162][164] The theorem can also be generalized to random fields so the index set is -dimensional Euclidean space[165] as well as to stochastic processes with metric spaces as their state spaces.[166]
Indistinguishable
Two stochastic processes and defined on the same probability space with the same index set and set space are said be indistinguishable if the following

holds.[142][158] If two and are modifications of each other and are almost surely continuous, then and are indistinguishable.[167]
Separability
Separability is a property of a stochastic process based on its index set in relation to the probability measure. The property is assumed so that functionals of stochastic processes or random fields with uncountable index sets can form random variables. For a stochastic process to be separable, in addition to other conditions, its index set must be a separable space,[b] which means that the index set has a dense countable subset.[150][168]
More precisely, a real-valued continuous-time stochastic process with a probability space is separable if its index set has a dense countable subset and there is a set of probability zero, so , such that for every open set and every closed set , the two events and differ from each other at most on a subset of .[169][170][171] The definition of separability[c] can also be stated for other index sets and state spaces,[174] such as in the case of random fields, where the index set as well as the state space can be -dimensional Euclidean space.[30][150]
The concept of separability of a stochastic process was introduced by Joseph Doob,.[168] The underlying idea of separability is to make a countable set of points of the index set determine the properties of the stochastic process.[172] Any stochastic process with a countable index set already meets the separability conditions, so discrete-time stochastic processes are always separable.[175] A theorem by Doob, sometimes known as Doob's separability theorem, says that any real-valued continuous-time stochastic process has a separable modification.[168][170][176] Versions of this theorem also exist for more general stochastic processes with index sets and state spaces other than the real line.[136]
Independence
Two stochastic processes and defined on the same probability space with the same index set are said be independent if for all and for every choice of epochs , the random vectors and are independent.[177]: p. 515
Uncorrelatedness
Two stochastic processes and are called uncorrelated if their cross-covariance is zero for all times.[178]: p. 142 Formally:

.
Independence implies uncorrelatedness
If two stochastic processes and are independent, then they are also uncorrelated.[178]: p. 151
Orthogonality
Two stochastic processes and are called orthogonal if their cross-correlation is zero for all times.[178]: p. 142 Formally:

.
Skorokhod space
Main article: Skorokhod space
A Skorokhod space, also written as Skorohod space, is a mathematical space of all the functions

that are right-continuous with left limits, defined on some interval of the real line such as or , and take values on the real line or on some metric space.[179][180][181] Such functions are known as càdlàg or cadlag functions, based on the acronym of the French phrase continue à droite, limite à gauche.[179][182] A Skorokhod function space, introduced by Anatoliy Skorokhod,[181] is often denoted with the letter ,[179][180][181][182] so the function space is also referred to as space .[179][183][184] The notation of this function space can also include the interval on which all the càdlàg functions are defined, so, for example, denotes the space of càdlàg functions defined on the unit interval .[182][184][185]
Skorokhod function spaces are frequently used in the theory of stochastic processes because it often assumed that the sample functions of continuous-time stochastic processes belong to a Skorokhod space.[181][183] Such spaces contain continuous functions, which correspond to sample functions of the Wiener process. But the space also has functions with discontinuities, which means that the sample functions of stochastic processes with jumps, such as the Poisson process (on the real line), are also members of this space.[184][186]
Regularity
In the context of mathematical construction of stochastic processes, the term regularity is used when discussing and assuming certain conditions for a stochastic process to resolve possible construction issues.[187][188] For example, to study stochastic processes with uncountable index sets, it is assumed that the stochastic process adheres to some type of regularity condition such as the sample functions being continuous.[189][190]
Further examples
Markov processes and chains
Main article: Markov chain
Markov processes are stochastic processes, traditionally in discrete or continuous time, that have the Markov property, which means the next value of the Markov process depends on the current value, but it is conditionally independent of the previous values of the stochastic process. In other words, the behavior of the process in the future is stochastically independent of its behavior in the past, given the current state of the process.[191][192]
The Brownian motion process and the Poisson process (in one dimension) are both examples of Markov processes[193] in continuous time, while random walks on the integers and the gambler's ruin problem are examples of Markov processes in discrete time.[194][195]
A Markov chain is a type of Markov process that has either discrete state space or discrete index set (often representing time), but the precise definition of a Markov chain varies.[196] For example, it is common to define a Markov chain as a Markov process in either discrete or continuous time with a countable state space (thus regardless of the nature of time),[197][198][199][200] but it has been also common to define a Markov chain as having discrete time in either countable or continuous state space (thus regardless of the state space).[196] It has been argued that the first definition of a Markov chain, where it has discrete time, now tends to be used, despite the second definition having been used by researchers like Joseph Doob and Kai Lai Chung.[201]
Markov processes form an important class of stochastic processes and have applications in many areas.[39][202] For example, they are the basis for a general stochastic simulation method known as Markov chain Monte Carlo, which is used for simulating random objects with specific probability distributions, and has found application in Bayesian statistics.[203][204]
The concept of the Markov property was originally for stochastic processes in continuous and discrete time, but the property has been adapted for other index sets such as -dimensional Euclidean space, which results in collections of random variables known as Markov random fields.[205][206][207]
Martingale
Main article: Martingale (probability theory)
A martingale is a discrete-time or continuous-time stochastic process with the property that, at every instant, given the current value and all the past values of the process, the conditional expectation of every future value is equal to the current value. In discrete time, if this property holds for the next value, then it holds for all future values. The exact mathematical definition of a martingale requires two other conditions coupled with the mathematical concept of a filtration,

which is related to the intuition of increasing available information as time passes. Martingales are usually defined to be real-valued,[208][209][155] but they can also be complex-valued[210] or even more general.[211]

A symmetric random walk and a Wiener process (with zero drift) are both examples of martingales, respectively, in discrete and continuous time.[208][209] For a sequence of independent and identically distributed random variables with zero mean, the stochastic process formed from the successive partial sums is a discrete-time martingale.[212] In this aspect, discrete-time martingales generalize the idea of partial sums of independent random variables.[213]

Martingales can also be created from stochastic processes by applying some suitable transformations, which is the case for the homogeneous Poisson process (on the real line) resulting in a martingale called the compensated Poisson process.[209] Martingales can also be built from other martingales.[212] For example, there are martingales based on the martingale the Wiener process, forming continuous-time martingales.[208][214]

Martingales mathematically formalize the idea of a 'fair game' where it is possible form reasonable expectations for payoffs,[215] and they were originally developed to show that it is not possible to gain an 'unfair' advantage in such a game.[216] But now they are used in many areas of probability, which is one of the main reasons for studying them.[155][216][217] Many problems in probability have been solved by finding a martingale in the problem and studying it.[218] Martingales will converge, given some conditions on their moments, so they are often used to derive convergence results, due largely to martingale convergence theorems.[213][219][220]

Martingales have many applications in statistics, but it has been remarked that its use and application are not as widespread as it could be in the field of statistics, particularly statistical inference.[221] They have found applications in areas in probability theory such as queueing theory and Palm calculus[222] and other fields such as economics[223] and finance.[17]

Lévy process

Main article: Lévy process

Lévy processes are types of stochastic processes that can be considered as generalizations of random walks in continuous time.[49][224] These processes have many applications in fields such as finance, fluid mechanics, physics and biology.[225][226] The main defining characteristics of these processes are their stationarity and independence properties, so they were known as processes with stationary and independent increments. In other words, a stochastic process is a Lévy process if for non-negatives numbers, , the corresponding increments

are all independent of each other, and the distribution of each increment only depends on the difference in time.[49]

A Lévy process can be defined such that its state space is some abstract mathematical space, such as a Banach space, but the processes are often defined so that they take values in Euclidean space. The index set is the non-negative numbers, so , which gives the interpretation of time. Important stochastic processes such as the Wiener process, the homogeneous Poisson process (in one dimension), and subordinators are all Lévy processes.[49][224]

Random field

Main article: Random field

A random field is a collection of random variables indexed by a -dimensional Euclidean space or some manifold. In general, a random field can be considered an example of a stochastic or random process, where the index set is not necessarily a subset of the real line.[30] But there is a convention that an indexed collection of random variables is called a random field when the index has two or more dimensions.[5][28][227] If the specific definition of a stochastic process requires the index set to be a subset of the real line, then the random field can be considered as a generalization of stochastic process.[228]

Point process

Main article: Point process

A point process is a collection of points randomly located on some mathematical space such as the real line, -dimensional Euclidean space, or more abstract spaces. Sometimes the term point process is not preferred, as historically the word process denoted an evolution of some system in

time, so a point process is also called a random point field.[229] There are different interpretations of a point process, such a random counting measure or a random set.[230][231] Some authors regard a point process and stochastic process as two different objects such that a point process is a random object that arises from or is associated with a stochastic process,[232][233] though it has been remarked that the difference between point processes and stochastic processes is not clear.[233]

Other authors consider a point process as a stochastic process, where the process is indexed by sets of the underlying space[d] on which it is defined, such as the real line or -dimensional Euclidean space.[236][237] Other stochastic processes such as renewal and counting processes are studied in the theory of point processes.[238][233]

History

Early probability theory

Probability theory has its origins in games of chance, which have a long history, with some games being played thousands of years ago,[239] but very little analysis on them was done in terms of probability.[240] The year 1654 is often considered the birth of probability theory when French mathematicians Pierre Fermat and Blaise Pascal had a written correspondence on probability, motivated by a gambling problem.[241][242] But there was earlier mathematical work done on the probability of gambling games such as Liber de Ludo Aleae by Gerolamo Cardano, written in the 16th century but posthumously published later in 1663.[243]

After Cardano, Jakob Bernoulli[e] wrote Ars Conjectandi, which is considered a significant event in the history of probability theory. Bernoulli's book was published, also posthumously, in 1713 and inspired many mathematicians to study probability.[245][246] But despite some renowned mathematicians contributing to probability theory, such as Pierre-Simon Laplace, Abraham de Moivre, Carl Gauss, Siméon Poisson and Pafnuty Chebyshev,[247][248] most of the mathematical community[f] did not consider probability theory to be part of mathematics until the 20th century.[247][249][250][251]

Statistical mechanics

In the physical sciences, scientists developed in the 19th century the discipline of statistical mechanics, where physical systems, such as containers filled with gases, are regarded or treated mathematically as collections of many moving particles. Although there were attempts to incorporate randomness into statistical physics by some scientists, such as Rudolf Clausius, most of the work had little or no randomness.[252][253] This changed in 1859 when James Clerk Maxwell contributed significantly to the field, more specifically, to the kinetic theory of gases, by presenting work where he modelled the gas particles as moving in random directions at random velocities.[254][255] The kinetic theory of gases and statistical physics continued to be developed in the second half of the 19th century, with work done chiefly by Clausius, Ludwig Boltzmann and Josiah Gibbs, which would later have an influence on Albert Einstein's mathematical model for Brownian movement.[256]

Measure theory and probability theory

At the International Congress of Mathematicians in Paris in 1900, David Hilbert presented a list of mathematical problems, where his sixth problem asked for a mathematical treatment of physics and probability involving axioms.[248] Around the start of the 20th century, mathematicians developed measure theory, a branch of mathematics for studying integrals of mathematical functions, where two of the founders were French mathematicians, Henri Lebesgue and Émile Borel. In 1925, another French mathematician Paul Lévy published the first probability book that used ideas from measure theory.[248]

In the 1920s, fundamental contributions to probability theory were made in the Soviet Union by mathematicians such as Sergei Bernstein, Aleksandr Khinchin,[g] and Andrei Kolmogorov.[251] Kolmogorov published in 1929 his first attempt at presenting a mathematical foundation, based on measure theory, for probability theory.[257] In the early 1930s, Khinchin and Kolmogorov set up probability seminars, which were attended by researchers such as Eugene Slutsky and Nikolai Smirnov,[258] and Khinchin gave the first mathematical definition of a stochastic process as a set of random variables indexed by the real line.[63][259][h]

Birth of modern probability theory

In 1933, Andrei Kolmogorov published in German, his book on the foundations of probability theory titled Grundbegriffe der Wahrscheinlichkeitsrechnung,[i] where Kolmogorov used measure theory to develop an axiomatic framework for probability theory. The publication of this book is now widely considered to be the birth of modern probability theory, when the theories of probability and stochastic processes became parts of mathematics.[248][251]
After the publication of Kolmogorov's book, further fundamental work on probability theory and stochastic processes was done by Khinchin and Kolmogorov as well as other mathematicians such as Joseph Doob, William Feller, Maurice Fréchet, Paul Lévy, Wolfgang Doeblin, and Harald Cramér.[248][251] Decades later, Cramér referred to the 1930s as the "heroic period of mathematical probability theory".[251] World War II greatly interrupted the development of probability theory, causing, for example, the migration of Feller from Sweden to the United States of America[251] and the death of Doeblin, considered now a pioneer in stochastic processes.[261]
Mathematician Joseph Doob did early work on the theory of stochastic processes, making fundamental contributions, particularly in the theory of martingales.[262][260] His book Stochastic Processes is considered highly influential in the field of probability theory.[263]
Stochastic processes after World War II
After World War II, the study of probability theory and stochastic processes gained more attention from mathematicians, with significant contributions made in many areas of probability and mathematics as well as the creation of new areas.[251][264] Starting in the 1940s, Kiyosi Itô published papers developing the field of stochastic calculus, which involves stochastic integrals and stochastic differential equations based on the Wiener or Brownian motion process.[265]
Also starting in the 1940s, connections were made between stochastic processes, particularly martingales, and the mathematical field of potential theory, with early ideas by Shizuo Kakutani and then later work by Joseph Doob.[264] Further work, considered pioneering, was done by Gilbert Hunt in the 1950s, connecting Markov processes and potential theory, which had a significant effect on the theory of Lévy processes and led to more interest in studying Markov processes with methods developed by Itô.[21][266][267]
In 1953, Doob published his book Stochastic processes, which had a strong influence on the theory of stochastic processes and stressed the importance of measure theory in probability.[264] [263] Doob also chiefly developed the theory of martingales, with later substantial contributions by Paul-André Meyer. Earlier work had been carried out by Sergei Bernstein, Paul Lévy and Jean Ville, the latter adopting the term martingale for the stochastic process.[268][269] Methods from the theory of martingales became popular for solving various probability problems. Techniques and theory were developed to study Markov processes and then applied to martingales. Conversely, methods from the theory of martingales were established to treat Markov processes.[264]
Other fields of probability were developed and used to study stochastic processes, with one main approach being the theory of large deviations.[264] The theory has many applications in statistical physics, among other fields, and has core ideas going back to at least the 1930s. Later in the 1960s and 1970s, fundamental work was done by Alexander Wentzell in the Soviet Union and Monroe D. Donsker and Srinivasa Varadhan in the United States of America,[270] which would later result in Varadhan winning the 2007 Abel Prize.[271] In the 1990s and 2000s the theories of Schramm–Loewner evolution[272] and rough paths[142] were introduced and developed to study stochastic processes and other mathematical objects in probability theory, which respectively resulted in Fields Medals being awarded to Wendelin Werner[273] in 2008 and to Martin Hairer in 2014.[274]
The theory of stochastic processes still continues to be a focus of research, with yearly international conferences on the topic of stochastic processes.[45][225]
Discoveries of specific stochastic processes
Although Khinchin gave mathematical definitions of stochastic processes in the 1930s,[63][259] specific stochastic processes had already been discovered in different settings, such as the Brownian motion process and the Poisson process.[21][24] Some families of stochastic processes such as point processes or renewal processes have long and complex histories, stretching back centuries.[275]
Bernoulli process
The Bernoulli process, which can serve as a mathematical model for flipping a biased coin, is

possibly the first stochastic process to have been studied.[81] The process is a sequence of independent Bernoulli trials,[82] which are named after Jacob Bernoulli who used them to study games of chance, including probability problems proposed and studied earlier by Christiaan Huygens.[276] Bernoulli's work, including the Bernoulli process, were published in his book Ars Conjectandi in 1713.[277]

Random walks

In 1905, Karl Pearson coined the term random walk while posing a problem describing a random walk on the plane, which was motivated by an application in biology, but such problems involving random walks had already been studied in other fields. Certain gambling problems that were studied centuries earlier can be considered as problems involving random walks.[89][277] For example, the problem known as the Gambler's ruin is based on a simple random walk,[195][278] and is an example of a random walk with absorbing barriers.[241][279] Pascal, Fermat and Huyens all gave numerical solutions to this problem without detailing their methods,[280] and then more detailed solutions were presented by Jakob Bernoulli and Abraham de Moivre.[281]

For random walks in -dimensional integer lattices, George Pólya published, in 1919 and 1921, work where he studied the probability of a symmetric random walk returning to a previous position in the lattice. Pólya showed that a symmetric random walk, which has an equal probability to advance in any direction in the lattice, will return to a previous position in the lattice an infinite number of times with probability one in one and two dimensions, but with probability zero in three or higher dimensions.[282][283]

Wiener process

The Wiener process or Brownian motion process has its origins in different fields including statistics, finance and physics.[21] In 1880, Danish astronomer Thorvald Thiele wrote a paper on the method of least squares, where he used the process to study the errors of a model in time-series analysis.[284][285][286] The work is now considered as an early discovery of the statistical method known as Kalman filtering, but the work was largely overlooked. It is thought that the ideas in Thiele's paper were too advanced to have been understood by the broader mathematical and statistical community at the time.[286]

Norbert Wiener gave the first mathematical proof of the existence of the Wiener process. This mathematical object had appeared previously in the work of Thorvald Thiele, Louis Bachelier, and Albert Einstein.[21]

The French mathematician Louis Bachelier used a Wiener process in his 1900 thesis[287][288] in order to model price changes on the Paris Bourse, a stock exchange,[289] without knowing the work of Thiele.[21] It has been speculated that Bachelier drew ideas from the random walk model of Jules Regnault, but Bachelier did not cite him,[290] and Bachelier's thesis is now considered pioneering in the field of financial mathematics.[289][290]

It is commonly thought that Bachelier's work gained little attention and was forgotten for decades until it was rediscovered in the 1950s by the Leonard Savage, and then become more popular after Bachelier's thesis was translated into English in 1964. But the work was never forgotten in the mathematical community, as Bachelier published a book in 1912 detailing his ideas,[290] which was cited by mathematicians including Doob, Feller[290] and Kolmogorov.[21] The book continued to be cited, but then starting in the 1960s, the original thesis by Bachelier began to be cited more than his book when economists started citing Bachelier's work.[290]

In 1905, Albert Einstein published a paper where he studied the physical observation of Brownian motion or movement to explain the seemingly random movements of particles in liquids by using ideas from the kinetic theory of gases. Einstein derived a differential equation, known as a diffusion equation, for describing the probability of finding a particle in a certain region of space. Shortly after Einstein's first paper on Brownian movement, Marian Smoluchowski published work where he cited Einstein, but wrote that he had independently derived the equivalent results by using a different method.[291]

Einstein's work, as well as experimental results obtained by Jean Perrin, later inspired Norbert Wiener in the 1920s[292] to use a type of measure theory, developed by Percy Daniell, and Fourier analysis to prove the existence of the Wiener process as a mathematical object.[21]

Poisson process

The Poisson process is named after Siméon Poisson, due to its definition involving the Poisson distribution, but Poisson never studied the process.[22][293] There are a number of claims for early uses or discoveries of the Poisson process.[22][24] At the beginning of the 20th century, the Poisson process would arise independently in different situations.[22][24] In Sweden 1903, Filip Lundberg published a thesis containing work, now considered fundamental and pioneering, where he proposed to model insurance claims with a homogeneous Poisson process.[294][295]
Another discovery occurred in Denmark in 1909 when A.K. Erlang derived the Poisson distribution when developing a mathematical model for the number of incoming phone calls in a finite time interval. Erlang was not at the time aware of Poisson's earlier work and assumed that the number phone calls arriving in each interval of time were independent to each other. He then found the limiting case, which is effectively recasting the Poisson distribution as a limit of the binomial distribution.[22]
In 1910, Ernest Rutherford and Hans Geiger published experimental results on counting alpha particles. Motivated by their work, Harry Bateman studied the counting problem and derived Poisson probabilities as a solution to a family of differential equations, resulting in the independent discovery of the Poisson process.[22] After this time there were many studies and applications of the Poisson process, but its early history is complicated, which has been explained by the various applications of the process in numerous fields by biologists, ecologists, engineers and various physical scientists.[22]
Markov processes
Markov processes and Markov chains are named after Andrey Markov who studied Markov chains in the early 20th century. Markov was interested in studying an extension of independent random sequences. In his first paper on Markov chains, published in 1906, Markov showed that under certain conditions the average outcomes of the Markov chain would converge to a fixed vector of values, so proving a weak law of large numbers without the independence assumption,[296][297][298] which had been commonly regarded as a requirement for such mathematical laws to hold.[298] Markov later used Markov chains to study the distribution of vowels in Eugene Onegin, written by Alexander Pushkin, and proved a central limit theorem for such chains.
In 1912, Poincaré studied Markov chains on finite groups with an aim to study card shuffling. Other early uses of Markov chains include a diffusion model, introduced by Paul and Tatyana Ehrenfest in 1907, and a branching process, introduced by Francis Galton and Henry William Watson in 1873, preceding the work of Markov.[296][297] After the work of Galton and Watson, it was later revealed that their branching process had been independently discovered and studied around three decades earlier by Irénée-Jules Bienaymé.[299] Starting in 1928, Maurice Fréchet became interested in Markov chains, eventually resulting in him publishing in 1938 a detailed study on Markov chains.[296][300]
Andrei Kolmogorov developed in a 1931 paper a large part of the early theory of continuous-time Markov processes.[251][257] Kolmogorov was partly inspired by Louis Bachelier's 1900 work on fluctuations in the stock market as well as Norbert Wiener's work on Einstein's model of Brownian movement.[257][301] He introduced and studied a particular set of Markov processes known as diffusion processes, where he derived a set of differential equations describing the processes.[257][302] Independent of Kolmogorov's work, Sydney Chapman derived in a 1928 paper an equation, now called the Chapman–Kolmogorov equation, in a less mathematically rigorous way than Kolmogorov, while studying Brownian movement.[303] The differential equations are now called the Kolmogorov equations[304] or the Kolmogorov–Chapman equations.[305] Other mathematicians who contributed significantly to the foundations of Markov processes include William Feller, starting in the 1930s, and then later Eugene Dynkin, starting in the 1950s.[251]
Lévy processes
Lévy processes such as the Wiener process and the Poisson process (on the real line) are named after Paul Lévy who started studying them in the 1930s,[225] but they have connections to infinitely divisible distributions going back to the 1920s.[224] In a 1932 paper, Kolmogorov derived a characteristic function for random variables associated with Lévy processes. This result was later derived under more general conditions by Lévy in 1934, and then Khinchin independently gave an alternative form for this characteristic function in 1937.[251][306] In addition to Lévy, Khinchin and

Kolomogrov, early fundamental contributions to the theory of Lévy processes were made by Bruno de Finetti and Kiyosi Itô.[224]

Mathematical construction

In mathematics, constructions of mathematical objects are needed, which is also the case for stochastic processes, to prove that they exist mathematically.[57] There are two main approaches for constructing a stochastic process. One approach involves considering a measurable space of functions, defining a suitable measurable mapping from a probability space to this measurable space of functions, and then deriving the corresponding finite-dimensional distributions.[307] Another approach involves defining a collection of random variables to have specific finite-dimensional distributions, and then using Kolmogorov's existence theorem[j] to prove a corresponding stochastic process exists.[57][307] This theorem, which is an existence theorem for measures on infinite product spaces,[311] says that if any finite-dimensional distributions satisfy two conditions, known as consistency conditions, then there exists a stochastic process with those finite-dimensional distributions.[57]

Construction issues

When constructing continuous-time stochastic processes certain mathematical difficulties arise, due to the uncountable index sets, which do not occur with discrete-time processes.[58][59] One problem is that it is possible to have more than one stochastic process with the same finite-dimensional distributions. For example, both the left-continuous modification and the right-continuous modification of a Poisson process have the same finite-dimensional distributions.[312] This means that the distribution of the stochastic process does not, necessarily, specify uniquely the properties of the sample functions of the stochastic process.[307][313]

Another problem is that functionals of continuous-time process that rely upon an uncountable number of points of the index set may not be measurable, so the probabilities of certain events may not be well-defined.[168] For example, the supremum of a stochastic process or random field is not necessarily a well-defined random variable.[30][59] For a continuous-time stochastic process , other characteristics that depend on an uncountable number of points of the index set include:[168]

- **a sample function of a stochastic process is a continuous function of ;**
- **a sample function of a stochastic process is a bounded function of ; and**
- **a sample function of a stochastic process is an increasing function of .**

where the symbol ∈ can be read "a member of the set", as in a member of the set .

To overcome the two difficulties described above, i.e., "more than one..." and "functionals of...", different assumptions and approaches are possible.[69]

Resolving construction issues

One approach for avoiding mathematical construction issues of stochastic processes, proposed by Joseph Doob, is to assume that the stochastic process is separable.[314] Separability ensures that infinite-dimensional distributions determine the properties of sample functions by requiring that sample functions are essentially determined by their values on a dense countable set of points in the index set.[315] Furthermore, if a stochastic process is separable, then functionals of an uncountable number of points of the index set are measurable and their probabilities can be studied.[168][315]

Another approach is possible, originally developed by Anatoliy Skorokhod and Andrei Kolmogorov,[316] for a continuous-time stochastic process with any metric space as its state space. For the construction of such a stochastic process, it is assumed that the sample functions of the stochastic process belong to some suitable function space, which is usually the Skorokhod space consisting of all right-continuous functions with left limits. This approach is now more used than the separability assumption,[69][262] but such a stochastic process based on this approach will be automatically separable.[317]

Although less used, the separability assumption is considered more general because every stochastic process has a separable version.[262] It is also used when it is not possible to construct a stochastic process in a Skorokhod space.[173] For example, separability is assumed when constructing and studying random fields, where the collection of random variables is now indexed by sets other than the real line such as -dimensional Euclidean space.[30][318]

Application

Applications in Finance

Black-Scholes Model

One of the most famous applications of stochastic processes in finance is the Black-Scholes model for option pricing. Developed by Fischer Black, Myron Scholes, and Robert Solow, this model uses Geometric Brownian motion, a specific type of stochastic process, to describe the dynamics of asset prices.[319][320] The model assumes that the price of a stock follows a continuous-time stochastic process and provides a closed-form solution for pricing European-style options. The Black-Scholes formula has had a profound impact on financial markets, forming the basis for much of modern options trading.

The key assumption of the Black-Scholes model is that the price of a financial asset, such as a stock, follows a log-normal distribution, with its continuous returns following a normal distribution. Although the model has limitations, such as the assumption of constant volatility, it remains widely used due to its simplicity and practical relevance.

Stochastic Volatility Models

Another significant application of stochastic processes in finance is in stochastic volatility models, which aim to capture the time-varying nature of market volatility. The Heston model[321] is a popular example, allowing for the volatility of asset prices to follow its own stochastic process. Unlike the Black-Scholes model, which assumes constant volatility, stochastic volatility models provide a more flexible framework for modeling market dynamics, particularly during periods of high uncertainty or market stress.

Applications in Biology

Population Dynamics

One of the primary applications of stochastic processes in biology is in population dynamics. In contrast to deterministic models, which assume that populations change in predictable ways, stochastic models account for the inherent randomness in births, deaths, and migration. The birth-death process,[322] a simple stochastic model, describes how populations fluctuate over time due to random births and deaths. These models are particularly important when dealing with small populations, where random events can have large impacts, such as in the case of endangered species or small microbial populations.

Another example is the branching process,[323] which models the growth of a population where each individual reproduces independently. The branching process is often used to describe population extinction or explosion, particularly in epidemiology, where it can model the spread of infectious diseases within a population.

See also

- List of stochastic processes topics
- Covariance function
- Deterministic system
- Dynamics of Markovian particles
- Entropy rate (for a stochastic process)
- Ergodic process
- Gillespie algorithm
- Interacting particle system
- Markov chain
- Stochastic cellular automaton
- Random field
- Randomness
- Stationary process
- Statistical model
- Stochastic calculus
- Stochastic control
- Stochastic parrot
- Stochastic processes and boundary value problems

Notes

1.

• The term Brownian motion can refer to the physical process, also known as Brownian movement, and the stochastic process, a mathematical object, but to avoid ambiguity this article uses the terms Brownian motion process or Wiener process for the latter in a style similar to, for example, Gikhman and Skorokhod[19] or Rosenblatt.[20]
• • The term "separable" appears twice here with two different meanings, where the first meaning is from probability and the second from topology and analysis. For a stochastic process to be separable (in a probabilistic sense), its index set must be a separable space (in a topological or analytic sense), in addition to other conditions.[136]
• • The definition of separability for a continuous-time real-valued stochastic process can be stated in other ways.[172][173]
• • In the context of point processes, the term "state space" can mean the space on which the point process is defined such as the real line,[234][235] which corresponds to the index set in stochastic process terminology.
• • Also known as James or Jacques Bernoulli.[244]
• • It has been remarked that a notable exception was the St Petersburg School in Russia, where mathematicians led by Chebyshev studied probability theory.[249]
• • The name Khinchin is also written in (or transliterated into) English as Khintchine.[63]
• • Doob, when citing Khinchin, uses the term 'chance variable', which used to be an alternative term for 'random variable'.[260]
• • Later translated into English and published in 1950 as Foundations of the Theory of Probability[248]
10. • The theorem has other names including Kolmogorov's consistency theorem,[308] Kolmogorov's extension theorem[309] or the Daniell–Kolmogorov theorem.[310]

References

1.
• Joseph L. Doob (1990). Stochastic processes. Wiley. pp. 46, 47.
• • L. C. G. Rogers; David Williams (2000). Diffusions, Markov Processes, and Martingales: Volume 1, Foundations. Cambridge University Press. p. 1. ISBN 978-1-107-71749-7.
• • J. Michael Steele (2012). Stochastic Calculus and Financial Applications. Springer Science & Business Media. p. 29. ISBN 978-1-4684-9305-4.
• • Emanuel Parzen (2015). Stochastic Processes. Courier Dover Publications. pp. 7, 8. ISBN 978-0-486-79688-8.
• • Iosif Ilyich Gikhman; Anatoly Vladimirovich Skorokhod (1969). Introduction to the Theory of Random Processes. Courier Corporation. p. 1. ISBN 978-0-486-69387-3.
• • Bressloff, Paul C. (2014). Stochastic Processes in Cell Biology. Springer. ISBN 978-3-319-08488-6.
• • Van Kampen, N. G. (2011). Stochastic Processes in Physics and Chemistry. Elsevier. ISBN 978-0-08-047536-3.
• • Lande, Russell; Engen, Steinar; Sæther, Bernt-Erik (2003). Stochastic Population Dynamics in Ecology and Conservation. Oxford University Press. ISBN 978-0-19-852525-7.
• • Laing, Carlo; Lord, Gabriel J. (2010). Stochastic Methods in Neuroscience. Oxford University Press. ISBN 978-0-19-923507-0.
• • Paul, Wolfgang; Baschnagel, Jörg (2013). Stochastic Processes: From Physics to Finance. Springer Science+Business Media. ISBN 978-3-319-00327-6.
• • Dougherty, Edward R. (1999). Random processes for image and signal processing. SPIE Optical Engineering Press. ISBN 978-0-8194-2513-3.
• • Bertsekas, Dimitri P. (1996). Stochastic Optimal Control: The Discrete-Time Case. Athena Scientific. ISBN 1-886529-03-5.
• • Thomas M. Cover; Joy A. Thomas (2012). Elements of Information Theory. John Wiley & Sons. p. 71. ISBN 978-1-118-58577-1.
• • Baron, Michael (2015). Probability and Statistics for Computer Scientists (2nd ed.). CRC Press. p. 131. ISBN 978-1-4987-6060-7.
• • Baccelli, François; Blaszczyszyn, Bartlomiej (2009). Stochastic Geometry and Wireless Networks. Now Publishers Inc. ISBN 978-1-60198-264-3.

• • Steele, J. Michael (2001). Stochastic Calculus and Financial Applications. Springer Science+Business Media. ISBN 978-0-387-95016-7.
• • Musiela, Marek; Rutkowski, Marek (2006). Martingale Methods in Financial Modelling. Springer Science+Business Media. ISBN 978-3-540-26653-2.
• • Shreve, Steven E. (2004). Stochastic Calculus for Finance II: Continuous-Time Models. Springer Science+Business Media. ISBN 978-0-387-40101-0.
• • Iosif Ilyich Gikhman; Anatoly Vladimirovich Skorokhod (1969). Introduction to the Theory of Random Processes. Courier Corporation. ISBN 978-0-486-69387-3.
• • Murray Rosenblatt (1962). Random Processes. Oxford University Press.
• • Jarrow, Robert; Protter, Philip (2004). "A short history of stochastic integration and mathematical finance: the early years, 1880–1970". A Festschrift for Herman Rubin. Institute of Mathematical Statistics Lecture Notes - Monograph Series. pp. 75–80. CiteSeerX 10.1.1.114.632. doi:10.1214/lnms/1196285381. ISBN 978-0-940600-61-4. ISSN 0749-2170.
• • Stirzaker, David (2000). "Advice to Hedgehogs, or, Constants Can Vary". The Mathematical Gazette. 84 (500): 197–210. doi:10.2307/3621649. ISSN 0025-5572. JSTOR 3621649. S2CID 125163415.
• • Donald L. Snyder; Michael I. Miller (2012). Random Point Processes in Time and Space. Springer Science & Business Media. p. 32. ISBN 978-1-4612-3166-0.
• • Guttorp, Peter; Thorarinsdottir, Thordis L. (2012). "What Happened to Discrete Chaos, the Quenouille Process, and the Sharp Markov Property? Some History of Stochastic Point Processes". International Statistical Review. 80 (2): 253–268. doi:10.1111/j.1751-5823.2012.00181.x. ISSN 0306-7734. S2CID 80836.
• • Gusak, Dmytro; Kukush, Alexander; Kulik, Alexey; Mishura, Yuliya; Pilipenko, Andrey (2010). Theory of Stochastic Processes: With Applications to Financial Mathematics and Risk Theory. Springer Science & Business Media. p. 21. ISBN 978-0-387-87862-1.
• • Valeriy Skorokhod (2005). Basic Principles and Applications of Probability Theory. Springer Science & Business Media. p. 42. ISBN 978-3-540-26312-8.
• • Olav Kallenberg (2002). Foundations of Modern Probability. Springer Science & Business Media. pp. 24–25. ISBN 978-0-387-95313-7.
• • John Lamperti (1977). Stochastic processes: a survey of the mathematical theory. Springer-Verlag. pp. 1–2. ISBN 978-3-540-90275-1.
• • Loïc Chaumont; Marc Yor (2012). Exercises in Probability: A Guided Tour from Measure Theory to Random Processes, Via Conditioning. Cambridge University Press. p. 175. ISBN 978-1-107-60655-5.
• • Robert J. Adler; Jonathan E. Taylor (2009). Random Fields and Geometry. Springer Science & Business Media. pp. 7–8. ISBN 978-0-387-48116-6.
• • Gregory F. Lawler; Vlada Limic (2010). Random Walk: A Modern Introduction. Cambridge University Press. ISBN 978-1-139-48876-1.
• • David Williams (1991). Probability with Martingales. Cambridge University Press. ISBN 978-0-521-40605-5.
• • L. C. G. Rogers; David Williams (2000). Diffusions, Markov Processes, and Martingales: Volume 1, Foundations. Cambridge University Press. ISBN 978-1-107-71749-7.
• • David Applebaum (2004). Lévy Processes and Stochastic Calculus. Cambridge University Press. ISBN 978-0-521-83263-2.
• • Mikhail Lifshits (2012). Lectures on Gaussian Processes. Springer Science & Business Media. ISBN 978-3-642-24939-6.
• • Robert J. Adler (2010). The Geometry of Random Fields. SIAM. ISBN 978-0-89871-693-1.
• • Samuel Karlin; Howard E. Taylor (2012). A First Course in Stochastic Processes. Academic Press. ISBN 978-0-08-057041-9.
• • Bruce Hajek (2015). Random Processes for Engineers. Cambridge University Press. ISBN 978-1-316-24124-0.
• • G. Latouche; V. Ramaswami (1999). Introduction to Matrix Analytic Methods in Stochastic Modeling. SIAM. ISBN 978-0-89871-425-8.
• • D.J. Daley; David Vere-Jones (2007). An Introduction to the Theory of Point Processes: Volume II:

General Theory and Structure. Springer Science & Business Media. ISBN 978-0-387-21337-8.
• • Patrick Billingsley (2008). Probability and Measure. Wiley India Pvt. Limited. ISBN 978-81-265-1771-8.
• • Pierre Brémaud (2014). Fourier Analysis and Stochastic Processes. Springer. ISBN 978-3-319-09590-5.
• • Adam Bobrowski (2005). Functional Analysis for Probability and Stochastic Processes: An Introduction. Cambridge University Press. ISBN 978-0-521-83166-6.
• • Applebaum, David (2004). "Lévy processes: From probability to finance and quantum groups". Notices of the AMS. 51 (11): 1336–1347.
• • Jochen Blath; Peter Imkeller; Sylvie Roelly (2011). Surveys in Stochastic Processes. European Mathematical Society. ISBN 978-3-03719-072-2.
• • Michel Talagrand (2014). Upper and Lower Bounds for Stochastic Processes: Modern Methods and Classical Problems. Springer Science & Business Media. pp. 4–. ISBN 978-3-642-54075-2.
• • Paul C. Bressloff (2014). Stochastic Processes in Cell Biology. Springer. pp. vii–ix. ISBN 978-3-319-08488-6.
• • Samuel Karlin; Howard E. Taylor (2012). A First Course in Stochastic Processes. Academic Press. p. 27. ISBN 978-0-08-057041-9.
• • Applebaum, David (2004). "Lévy processes: From probability to finance and quantum groups". Notices of the AMS. 51 (11): 1337.
• • L. C. G. Rogers; David Williams (2000). Diffusions, Markov Processes, and Martingales: Volume 1, Foundations. Cambridge University Press. pp. 121–124. ISBN 978-1-107-71749-7.
• • Ionut Florescu (2014). Probability and Stochastic Processes. John Wiley & Sons. pp. 294, 295. ISBN 978-1-118-59320-2.
• • Samuel Karlin; Howard E. Taylor (2012). A First Course in Stochastic Processes. Academic Press. p. 26. ISBN 978-0-08-057041-9.
• • Donald L. Snyder; Michael I. Miller (2012). Random Point Processes in Time and Space. Springer Science & Business Media. pp. 24, 25. ISBN 978-1-4612-3166-0.
• • Patrick Billingsley (2008). Probability and Measure. Wiley India Pvt. Limited. p. 482. ISBN 978-81-265-1771-8.
• • Alexander A. Borovkov (2013). Probability Theory. Springer Science & Business Media. p. 527. ISBN 978-1-4471-5201-9.
• • Pierre Brémaud (2014). Fourier Analysis and Stochastic Processes. Springer. p. 120. ISBN 978-3-319-09590-5.
• • Jeffrey S Rosenthal (2006). A First Look at Rigorous Probability Theory. World Scientific Publishing Co Inc. pp. 177–178. ISBN 978-981-310-165-4.
• • Peter E. Kloeden; Eckhard Platen (2013). Numerical Solution of Stochastic Differential Equations. Springer Science & Business Media. p. 63. ISBN 978-3-662-12616-5.
• • Davar Khoshnevisan (2006). Multiparameter Processes: An Introduction to Random Fields. Springer Science & Business Media. pp. 153–155. ISBN 978-0-387-21631-7.
• • "Stochastic". Oxford English Dictionary (Online ed.). Oxford University Press. (Subscription or participating institution membership required.)
• • O. B. Sheĭnin (2006). Theory of probability and statistics as exemplified in short dictums. NG Verlag. p. 5. ISBN 978-3-938417-40-9.
• • Oscar Sheynin; Heinrich Strecker (2011). Alexandr A. Chuprov: Life, Work, Correspondence. V&R unipress GmbH. p. 136. ISBN 978-3-89971-812-6.
• • Doob, Joseph (1934). "Stochastic Processes and Statistics". Proceedings of the National Academy of Sciences of the United States of America. 20 (6): 376–379. Bibcode:1934PNAS...20..376D. doi:10.1073/pnas.20.6.376. PMC 1076423. PMID 16587907.
• • Khintchine, A. (1934). "Korrelationstheorie der stationeren stochastischen Prozesse". Mathematische Annalen. 109 (1): 604–615. doi:10.1007/BF01449156. ISSN 0025-5831. S2CID 122842868.
• • Kolmogoroff, A. (1931). "Über die analytischen Methoden in der Wahrscheinlichkeitsrechnung". Mathematische Annalen. 104 (1): 1. doi:10.1007/BF01457949. ISSN 0025-5831. S2CID 119439925.
• • "Random". Oxford English Dictionary (Online ed.). Oxford University Press. (Subscription or

participating institution membership required.)
• • Bert E. Fristedt; Lawrence F. Gray (2013). A Modern Approach to Probability Theory. Springer Science & Business Media. p. 580. ISBN 978-1-4899-2837-5.
• • L. C. G. Rogers; David Williams (2000). Diffusions, Markov Processes, and Martingales: Volume 1, Foundations. Cambridge University Press. pp. 121, 122. ISBN 978-1-107-71749-7.
• • Søren Asmussen (2003). Applied Probability and Queues. Springer Science & Business Media. p. 408. ISBN 978-0-387-00211-8.
• • David Stirzaker (2005). Stochastic Processes and Models. Oxford University Press. p. 45. ISBN 978-0-19-856814-8.
• • Murray Rosenblatt (1962). Random Processes. Oxford University Press. p. 91.
• • John A. Gubner (2006). Probability and Random Processes for Electrical and Computer Engineers. Cambridge University Press. p. 383. ISBN 978-1-139-45717-0.
• • Kiyosi Itō (2006). Essentials of Stochastic Processes. American Mathematical Soc. p. 13. ISBN 978-0-8218-3898-3.
• • M. Loève (1978). Probability Theory II. Springer Science & Business Media. p. 163. ISBN 978-0-387-90262-3.
• • Pierre Brémaud (2014). Fourier Analysis and Stochastic Processes. Springer. p. 133. ISBN 978-3-319-09590-5.
• • Gusak et al. (2010), p. 1
• • Richard F. Bass (2011). Stochastic Processes. Cambridge University Press. p. 1. ISBN 978-1-139-50147-7.
• • ,John Lamperti (1977). Stochastic processes: a survey of the mathematical theory. Springer-Verlag. p. 3. ISBN 978-3-540-90275-1.
• • Fima C. Klebaner (2005). Introduction to Stochastic Calculus with Applications. Imperial College Press. p. 55. ISBN 978-1-86094-555-7.
• • Ionut Florescu (2014). Probability and Stochastic Processes. John Wiley & Sons. p. 293. ISBN 978-1-118-59320-2.
• • Florescu, Ionut (2014). Probability and Stochastic Processes. John Wiley & Sons. p. 301. ISBN 978-1-118-59320-2.
• • Bertsekas, Dimitri P.; Tsitsiklis, John N. (2002). Introduction to Probability. Athena Scientific. p. 273. ISBN 978-1-886529-40-3.
• • Ibe, Oliver C. (2013). Elements of Random Walk and Diffusion Processes. John Wiley & Sons. p. 11. ISBN 978-1-118-61793-9.
• • Achim Klenke (2013). Probability Theory: A Comprehensive Course. Springer. p. 347. ISBN 978-1-4471-5362-7.
• • Gregory F. Lawler; Vlada Limic (2010). Random Walk: A Modern Introduction. Cambridge University Press. p. 1. ISBN 978-1-139-48876-1.
• • Olav Kallenberg (2002). Foundations of Modern Probability. Springer Science & Business Media. p. 136. ISBN 978-0-387-95313-7.
• • Ionut Florescu (2014). Probability and Stochastic Processes. John Wiley & Sons. p. 383. ISBN 978-1-118-59320-2.
• • Rick Durrett (2010). Probability: Theory and Examples. Cambridge University Press. p. 277. ISBN 978-1-139-49113-6.
• • Weiss, George H. (2006). "Random Walks". Encyclopedia of Statistical Sciences. p. 1. doi:10.1002/0471667196.ess2180.pub2. ISBN 978-0471667193.
• • Aris Spanos (1999). Probability Theory and Statistical Inference: Econometric Modeling with Observational Data. Cambridge University Press. p. 454. ISBN 978-0-521-42408-0.
• • Fima C. Klebaner (2005). Introduction to Stochastic Calculus with Applications. Imperial College Press. p. 81. ISBN 978-1-86094-555-7.
• • Allan Gut (2012). Probability: A Graduate Course. Springer Science & Business Media. p. 88. ISBN 978-1-4614-4708-5.
• • Geoffrey Grimmett; David Stirzaker (2001). Probability and Random Processes. OUP Oxford. p. 71. ISBN 978-0-19-857222-0.
• • Fima C. Klebaner (2005). Introduction to Stochastic Calculus with Applications. Imperial College

Press. p. 56. ISBN 978-1-86094-555-7.
• • Brush, Stephen G. (1968). "A history of random processes". Archive for History of Exact Sciences. 5 (1): 1–2. doi:10.1007/BF00328110. ISSN 0003-9519. S2CID 117623580.
• • Applebaum, David (2004). "Lévy processes: From probability to finance and quantum groups". Notices of the AMS. 51 (11): 1338.
• • Iosif Ilyich Gikhman; Anatoly Vladimirovich Skorokhod (1969). Introduction to the Theory of Random Processes. Courier Corporation. p. 21. ISBN 978-0-486-69387-3.
• • Ionut Florescu (2014). Probability and Stochastic Processes. John Wiley & Sons. p. 471. ISBN 978-1-118-59320-2.
• • Samuel Karlin; Howard E. Taylor (2012). A First Course in Stochastic Processes. Academic Press. pp. 21, 22. ISBN 978-0-08-057041-9.
• • Ioannis Karatzas; Steven Shreve (1991). Brownian Motion and Stochastic Calculus. Springer. p. VIII. ISBN 978-1-4612-0949-2.
• • Daniel Revuz; Marc Yor (2013). Continuous Martingales and Brownian Motion. Springer Science & Business Media. p. IX. ISBN 978-3-662-06400-9.
• • Jeffrey S Rosenthal (2006). A First Look at Rigorous Probability Theory. World Scientific Publishing Co Inc. p. 186. ISBN 978-981-310-165-4.
• • Donald L. Snyder; Michael I. Miller (2012). Random Point Processes in Time and Space. Springer Science & Business Media. p. 33. ISBN 978-1-4612-3166-0.
• • J. Michael Steele (2012). Stochastic Calculus and Financial Applications. Springer Science & Business Media. p. 118. ISBN 978-1-4684-9305-4.
• • Peter Mörters; Yuval Peres (2010). Brownian Motion. Cambridge University Press. pp. 1, 3. ISBN 978-1-139-48657-6.
• • Ioannis Karatzas; Steven Shreve (1991). Brownian Motion and Stochastic Calculus. Springer. p. 78. ISBN 978-1-4612-0949-2.
• • Ioannis Karatzas; Steven Shreve (1991). Brownian Motion and Stochastic Calculus. Springer. p. 61. ISBN 978-1-4612-0949-2.
• • Steven E. Shreve (2004). Stochastic Calculus for Finance II: Continuous-Time Models. Springer Science & Business Media. p. 93. ISBN 978-0-387-40101-0.
• • Olav Kallenberg (2002). Foundations of Modern Probability. Springer Science & Business Media. pp. 225, 260. ISBN 978-0-387-95313-7.
• • Ioannis Karatzas; Steven Shreve (1991). Brownian Motion and Stochastic Calculus. Springer. p. 70. ISBN 978-1-4612-0949-2.
• • Peter Mörters; Yuval Peres (2010). Brownian Motion. Cambridge University Press. p. 131. ISBN 978-1-139-48657-6.
• • Fima C. Klebaner (2005). Introduction to Stochastic Calculus with Applications. Imperial College Press. ISBN 978-1-86094-555-7.
• • Ioannis Karatzas; Steven Shreve (1991). Brownian Motion and Stochastic Calculus. Springer. ISBN 978-1-4612-0949-2.
• • Applebaum, David (2004). "Lévy processes: From probability to finance and quantum groups". Notices of the AMS. 51 (11): 1341.
• • Samuel Karlin; Howard E. Taylor (2012). A First Course in Stochastic Processes. Academic Press. p. 340. ISBN 978-0-08-057041-9.
• • Fima C. Klebaner (2005). Introduction to Stochastic Calculus with Applications. Imperial College Press. p. 124. ISBN 978-1-86094-555-7.
• • Ioannis Karatzas; Steven Shreve (1991). Brownian Motion and Stochastic Calculus. Springer. p. 47. ISBN 978-1-4612-0949-2.
• • Ubbo F. Wiersema (2008). Brownian Motion Calculus. John Wiley & Sons. p. 2. ISBN 978-0-470-02171-2.
• • Henk C. Tijms (2003). A First Course in Stochastic Models. Wiley. pp. 1, 2. ISBN 978-0-471-49881-0.
• • D.J. Daley; D. Vere-Jones (2006). An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods. Springer Science & Business Media. pp. 19–36. ISBN 978-0-387-21564-8.

• • Mark A. Pinsky; Samuel Karlin (2011). An Introduction to Stochastic Modeling. Academic Press. p. 241. ISBN 978-0-12-381416-6.
• • J. F. C. Kingman (1992). Poisson Processes. Clarendon Press. p. 38. ISBN 978-0-19-159124-2.
• • D.J. Daley; D. Vere-Jones (2006). An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods. Springer Science & Business Media. p. 19. ISBN 978-0-387-21564-8.
• • J. F. C. Kingman (1992). Poisson Processes. Clarendon Press. p. 22. ISBN 978-0-19-159124-2.
• • Samuel Karlin; Howard E. Taylor (2012). A First Course in Stochastic Processes. Academic Press. pp. 118, 119. ISBN 978-0-08-057041-9.
• • Leonard Kleinrock (1976). Queueing Systems: Theory. Wiley. p. 61. ISBN 978-0-471-49110-1.
• • Murray Rosenblatt (1962). Random Processes. Oxford University Press. p. 94.
• • Martin Haenggi (2013). Stochastic Geometry for Wireless Networks. Cambridge University Press. pp. 10, 18. ISBN 978-1-107-01469-5.
• • Sung Nok Chiu; Dietrich Stoyan; Wilfrid S. Kendall; Joseph Mecke (2013). Stochastic Geometry and Its Applications. John Wiley & Sons. pp. 41, 108. ISBN 978-1-118-65825-3.
• • J. F. C. Kingman (1992). Poisson Processes. Clarendon Press. p. 11. ISBN 978-0-19-159124-2.
• • Roy L. Streit (2010). Poisson Point Processes: Imaging, Tracking, and Sensing. Springer Science & Business Media. p. 1. ISBN 978-1-4419-6923-1.
• • J. F. C. Kingman (1992). Poisson Processes. Clarendon Press. p. v. ISBN 978-0-19-159124-2.
• • Alexander A. Borovkov (2013). Probability Theory. Springer Science & Business Media. p. 528. ISBN 978-1-4471-5201-9.
• • Georg Lindgren; Holger Rootzen; Maria Sandsten (2013). Stationary Stochastic Processes for Scientists and Engineers. CRC Press. p. 11. ISBN 978-1-4665-8618-5.
• • Aumann, Robert (December 1961). "Borel structures for function spaces". Illinois Journal of Mathematics. 5 (4). doi:10.1215/ijm/1255631584. S2CID 117171116.
• • Valeriy Skorokhod (2005). Basic Principles and Applications of Probability Theory. Springer Science & Business Media. pp. 93, 94. ISBN 978-3-540-26312-8.
• • Donald L. Snyder; Michael I. Miller (2012). Random Point Processes in Time and Space. Springer Science & Business Media. p. 25. ISBN 978-1-4612-3166-0.
• • Valeriy Skorokhod (2005). Basic Principles and Applications of Probability Theory. Springer Science & Business Media. p. 104. ISBN 978-3-540-26312-8.
• • Ionut Florescu (2014). Probability and Stochastic Processes. John Wiley & Sons. p. 296. ISBN 978-1-118-59320-2.
• • Patrick Billingsley (2008). Probability and Measure. Wiley India Pvt. Limited. p. 493. ISBN 978-81-265-1771-8.
• • Bernt Øksendal (2003). Stochastic Differential Equations: An Introduction with Applications. Springer Science & Business Media. p. 10. ISBN 978-3-540-04758-2.
• • Peter K. Friz; Nicolas B. Victoir (2010). Multidimensional Stochastic Processes as Rough Paths: Theory and Applications. Cambridge University Press. p. 571. ISBN 978-1-139-48721-4.
• • Sidney I. Resnick (2013). Adventures in Stochastic Processes. Springer Science & Business Media. pp. 40–41. ISBN 978-1-4612-0387-2.
• • Ward Whitt (2006). Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and Their Application to Queues. Springer Science & Business Media. p. 23. ISBN 978-0-387-21748-2.
• • David Applebaum (2004). Lévy Processes and Stochastic Calculus. Cambridge University Press. p. 4. ISBN 978-0-521-83263-2.
• • Daniel Revuz; Marc Yor (2013). Continuous Martingales and Brownian Motion. Springer Science & Business Media. p. 10. ISBN 978-3-662-06400-9.
• • L. C. G. Rogers; David Williams (2000). Diffusions, Markov Processes, and Martingales: Volume 1, Foundations. Cambridge University Press. p. 123. ISBN 978-1-107-71749-7.
• • John Lamperti (1977). Stochastic processes: a survey of the mathematical theory. Springer-Verlag. pp. 6 and 7. ISBN 978-3-540-90275-1.
• • Iosif I. Gikhman; Anatoly Vladimirovich Skorokhod (1969). Introduction to the Theory of Random Processes. Courier Corporation. p. 4. ISBN 978-0-486-69387-3.
• • Robert J. Adler (2010). The Geometry of Random Fields. SIAM. pp. 14, 15. ISBN 978-0-89871-693-

1.
• • Sung Nok Chiu; Dietrich Stoyan; Wilfrid S. Kendall; Joseph Mecke (2013). Stochastic Geometry and Its Applications. John Wiley & Sons. p. 112. ISBN 978-1-118-65825-3.
• • Joseph L. Doob (1990). Stochastic processes. Wiley. pp. 94–96.
• • Ionut Florescu (2014). Probability and Stochastic Processes. John Wiley & Sons. pp. 298, 299. ISBN 978-1-118-59320-2.
• • Iosif Ilyich Gikhman; Anatoly Vladimirovich Skorokhod (1969). Introduction to the Theory of Random Processes. Courier Corporation. p. 8. ISBN 978-0-486-69387-3.
• • David Williams (1991). Probability with Martingales. Cambridge University Press. pp. 93, 94. ISBN 978-0-521-40605-5.
• • Fima C. Klebaner (2005). Introduction to Stochastic Calculus with Applications. Imperial College Press. pp. 22–23. ISBN 978-1-86094-555-7.
• • Peter Mörters; Yuval Peres (2010). Brownian Motion. Cambridge University Press. p. 37. ISBN 978-1-139-48657-6.
• • L. C. G. Rogers; David Williams (2000). Diffusions, Markov Processes, and Martingales: Volume 1, Foundations. Cambridge University Press. p. 130. ISBN 978-1-107-71749-7.
• • Alexander A. Borovkov (2013). Probability Theory. Springer Science & Business Media. p. 530. ISBN 978-1-4471-5201-9.
• • Fima C. Klebaner (2005). Introduction to Stochastic Calculus with Applications. Imperial College Press. p. 48. ISBN 978-1-86094-555-7.
• • Bernt Øksendal (2003). Stochastic Differential Equations: An Introduction with Applications. Springer Science & Business Media. p. 14. ISBN 978-3-540-04758-2.
• • Ionut Florescu (2014). Probability and Stochastic Processes. John Wiley & Sons. p. 472. ISBN 978-1-118-59320-2.
• • Daniel Revuz; Marc Yor (2013). Continuous Martingales and Brownian Motion. Springer Science & Business Media. pp. 18–19. ISBN 978-3-662-06400-9.
• • David Applebaum (2004). Lévy Processes and Stochastic Calculus. Cambridge University Press. p. 20. ISBN 978-0-521-83263-2.
• • Hiroshi Kunita (1997). Stochastic Flows and Stochastic Differential Equations. Cambridge University Press. p. 31. ISBN 978-0-521-59925-2.
• • Olav Kallenberg (2002). Foundations of Modern Probability. Springer Science & Business Media. p. 35. ISBN 978-0-387-95313-7.
• • Monique Jeanblanc; Marc Yor; Marc Chesney (2009). Mathematical Methods for Financial Markets. Springer Science & Business Media. p. 11. ISBN 978-1-85233-376-8.
• • Kiyosi Itō (2006). Essentials of Stochastic Processes. American Mathematical Soc. pp. 32–33. ISBN 978-0-8218-3898-3.
• • Iosif Ilyich Gikhman; Anatoly Vladimirovich Skorokhod (1969). Introduction to the Theory of Random Processes. Courier Corporation. p. 150. ISBN 978-0-486-69387-3.
• • Petar Todorovic (2012). An Introduction to Stochastic Processes and Their Applications. Springer Science & Business Media. pp. 19–20. ISBN 978-1-4613-9742-7.
• • Ilya Molchanov (2005). Theory of Random Sets. Springer Science & Business Media. p. 340. ISBN 978-1-85233-892-3.
• • Patrick Billingsley (2008). Probability and Measure. Wiley India Pvt. Limited. pp. 526–527. ISBN 978-81-265-1771-8.
• • Alexander A. Borovkov (2013). Probability Theory. Springer Science & Business Media. p. 535. ISBN 978-1-4471-5201-9.
• • Gusak et al. (2010), p. 22
• • Joseph L. Doob (1990). Stochastic processes. Wiley. p. 56.
• • Davar Khoshnevisan (2006). Multiparameter Processes: An Introduction to Random Fields. Springer Science & Business Media. p. 155. ISBN 978-0-387-21631-7.
• • Lapidoth, Amos, A Foundation in Digital Communication, Cambridge University Press, 2009.
• • Kun Il Park, Fundamentals of Probability and Stochastic Processes with Applications to Communications, Springer, 2018, 978-3-319-68074-3
• • Ward Whitt (2006). Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and

Their Application to Queues. Springer Science & Business Media. pp. 78–79. ISBN 978-0-387-21748-2.
• • Gusak et al. (2010), p. 24
• • Vladimir I. Bogachev (2007). Measure Theory (Volume 2). Springer Science & Business Media. p. 53. ISBN 978-3-540-34514-5.
• • Fima C. Klebaner (2005). Introduction to Stochastic Calculus with Applications. Imperial College Press. p. 4. ISBN 978-1-86094-555-7.
• • Søren Asmussen (2003). Applied Probability and Queues. Springer Science & Business Media. p. 420. ISBN 978-0-387-00211-8.
• • Patrick Billingsley (2013). Convergence of Probability Measures. John Wiley & Sons. p. 121. ISBN 978-1-118-62596-5.
• • Richard F. Bass (2011). Stochastic Processes. Cambridge University Press. p. 34. ISBN 978-1-139-50147-7.
• • Nicholas H. Bingham; Rüdiger Kiesel (2013). Risk-Neutral Valuation: Pricing and Hedging of Financial Derivatives. Springer Science & Business Media. p. 154. ISBN 978-1-4471-3856-3.
• • Alexander A. Borovkov (2013). Probability Theory. Springer Science & Business Media. p. 532. ISBN 978-1-4471-5201-9.
• • Davar Khoshnevisan (2006). Multiparameter Processes: An Introduction to Random Fields. Springer Science & Business Media. pp. 148–165. ISBN 978-0-387-21631-7.
• • Petar Todorovic (2012). An Introduction to Stochastic Processes and Their Applications. Springer Science & Business Media. p. 22. ISBN 978-1-4613-9742-7.
• • Ward Whitt (2006). Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and Their Application to Queues. Springer Science & Business Media. p. 79. ISBN 978-0-387-21748-2.
• • Richard Serfozo (2009). Basics of Applied Stochastic Processes. Springer Science & Business Media. p. 2. ISBN 978-3-540-89332-5.
• • Y.A. Rozanov (2012). Markov Random Fields. Springer Science & Business Media. p. 58. ISBN 978-1-4613-8190-7.
• • Sheldon M. Ross (1996). Stochastic processes. Wiley. pp. 235, 358. ISBN 978-0-471-12062-9.
• • Ionut Florescu (2014). Probability and Stochastic Processes. John Wiley & Sons. pp. 373, 374. ISBN 978-1-118-59320-2.
• • Samuel Karlin; Howard E. Taylor (2012). A First Course in Stochastic Processes. Academic Press. p. 49. ISBN 978-0-08-057041-9.
• • Søren Asmussen (2003). Applied Probability and Queues. Springer Science & Business Media. p. 7. ISBN 978-0-387-00211-8.
• • Emanuel Parzen (2015). Stochastic Processes. Courier Dover Publications. p. 188. ISBN 978-0-486-79688-8.
• • Samuel Karlin; Howard E. Taylor (2012). A First Course in Stochastic Processes. Academic Press. pp. 29, 30. ISBN 978-0-08-057041-9.
• • John Lamperti (1977). Stochastic processes: a survey of the mathematical theory. Springer-Verlag. pp. 106–121. ISBN 978-3-540-90275-1.
• • Sheldon M. Ross (1996). Stochastic processes. Wiley. pp. 174, 231. ISBN 978-0-471-12062-9.
• • Sean Meyn; Richard L. Tweedie (2009). Markov Chains and Stochastic Stability. Cambridge University Press. p. 19. ISBN 978-0-521-73182-9.
• • Samuel Karlin; Howard E. Taylor (2012). A First Course in Stochastic Processes. Academic Press. p. 47. ISBN 978-0-08-057041-9.
• • Reuven Y. Rubinstein; Dirk P. Kroese (2011). Simulation and the Monte Carlo Method. John Wiley & Sons. p. 225. ISBN 978-1-118-21052-9.
• • Dani Gamerman; Hedibert F. Lopes (2006). Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Second Edition. CRC Press. ISBN 978-1-58488-587-0.
• • Y.A. Rozanov (2012). Markov Random Fields. Springer Science & Business Media. p. 61. ISBN 978-1-4613-8190-7.
• • Donald L. Snyder; Michael I. Miller (2012). Random Point Processes in Time and Space. Springer Science & Business Media. p. 27. ISBN 978-1-4612-3166-0.
• • Pierre Bremaud (2013). Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues.

Springer Science & Business Media. p. 253. ISBN 978-1-4757-3124-8.
• • Fima C. Klebaner (2005). Introduction to Stochastic Calculus with Applications. Imperial College Press. p. 65. ISBN 978-1-86094-555-7.
• • Ioannis Karatzas; Steven Shreve (1991). Brownian Motion and Stochastic Calculus. Springer. p. 11. ISBN 978-1-4612-0949-2.
• • Joseph L. Doob (1990). Stochastic processes. Wiley. pp. 292, 293.
• • Gilles Pisier (2016). Martingales in Banach Spaces. Cambridge University Press. ISBN 978-1-316-67946-3.
• • J. Michael Steele (2012). Stochastic Calculus and Financial Applications. Springer Science & Business Media. pp. 12, 13. ISBN 978-1-4684-9305-4.
• • P. Hall; C. C. Heyde (2014). Martingale Limit Theory and Its Application. Elsevier Science. p. 2. ISBN 978-1-4832-6322-9.
• • J. Michael Steele (2012). Stochastic Calculus and Financial Applications. Springer Science & Business Media. p. 115. ISBN 978-1-4684-9305-4.
• • Sheldon M. Ross (1996). Stochastic processes. Wiley. p. 295. ISBN 978-0-471-12062-9.
• • J. Michael Steele (2012). Stochastic Calculus and Financial Applications. Springer Science & Business Media. p. 11. ISBN 978-1-4684-9305-4.
• • Olav Kallenberg (2002). Foundations of Modern Probability. Springer Science & Business Media. p. 96. ISBN 978-0-387-95313-7.
• • J. Michael Steele (2012). Stochastic Calculus and Financial Applications. Springer Science & Business Media. p. 371. ISBN 978-1-4684-9305-4.
• • J. Michael Steele (2012). Stochastic Calculus and Financial Applications. Springer Science & Business Media. p. 22. ISBN 978-1-4684-9305-4.
• • Geoffrey Grimmett; David Stirzaker (2001). Probability and Random Processes. OUP Oxford. p. 336. ISBN 978-0-19-857222-0.
• • Glasserman, Paul; Kou, Steven (2006). "A Conversation with Chris Heyde". Statistical Science. 21 (2): 292, 293. arXiv:math/0609294. Bibcode:2006math......9294G. doi:10.1214/088342306000000088. ISSN 0883-4237. S2CID 62552177.
• • Francois Baccelli; Pierre Bremaud (2013). Elements of Queueing Theory: Palm Martingale Calculus and Stochastic Recurrences. Springer Science & Business Media. ISBN 978-3-662-11657-9.
• • P. Hall; C. C. Heyde (2014). Martingale Limit Theory and Its Application. Elsevier Science. p. x. ISBN 978-1-4832-6322-9.
• • Jean Bertoin (1998). Lévy Processes. Cambridge University Press. p. viii. ISBN 978-0-521-64632-1.
• • Applebaum, David (2004). "Lévy processes: From probability to finance and quantum groups". Notices of the AMS. 51 (11): 1336.
• • David Applebaum (2004). Lévy Processes and Stochastic Calculus. Cambridge University Press. p. 69. ISBN 978-0-521-83263-2.
• • Leonid Koralov; Yakov G. Sinai (2007). Theory of Probability and Random Processes. Springer Science & Business Media. p. 171. ISBN 978-3-540-68829-7.
• • David Applebaum (2004). Lévy Processes and Stochastic Calculus. Cambridge University Press. p. 19. ISBN 978-0-521-83263-2.
• • Sung Nok Chiu; Dietrich Stoyan; Wilfrid S. Kendall; Joseph Mecke (2013). Stochastic Geometry and Its Applications. John Wiley & Sons. p. 109. ISBN 978-1-118-65825-3.
• • Sung Nok Chiu; Dietrich Stoyan; Wilfrid S. Kendall; Joseph Mecke (2013). Stochastic Geometry and Its Applications. John Wiley & Sons. p. 108. ISBN 978-1-118-65825-3.
• • Martin Haenggi (2013). Stochastic Geometry for Wireless Networks. Cambridge University Press. p. 10. ISBN 978-1-107-01469-5.
• • D.J. Daley; D. Vere-Jones (2006). An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods. Springer Science & Business Media. p. 194. ISBN 978-0-387-21564-8.
• • Cox, D. R.; Isham, Valerie (1980). Point Processes. CRC Press. p. 3. ISBN 978-0-412-21910-8.
• • J. F. C. Kingman (1992). Poisson Processes. Clarendon Press. p. 8. ISBN 978-0-19-159124-2.

• • Jesper Moller; Rasmus Plenge Waagepetersen (2003). Statistical Inference and Simulation for Spatial Point Processes. CRC Press. p. 7. ISBN 978-0-203-49693-0.
• • Samuel Karlin; Howard E. Taylor (2012). A First Course in Stochastic Processes. Academic Press. p. 31. ISBN 978-0-08-057041-9.
• • Volker Schmidt (2014). Stochastic Geometry, Spatial Statistics and Random Fields: Models and Algorithms. Springer. p. 99. ISBN 978-3-319-10064-7.
• • D.J. Daley; D. Vere-Jones (2006). An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods. Springer Science & Business Media. ISBN 978-0-387-21564-8.
• • David, F. N. (1955). "Studies in the History of Probability and Statistics I. Dicing and Gaming (A Note on the History of Probability)". Biometrika. 42 (1/2): 1–15. doi:10.2307/2333419. ISSN 0006-3444. JSTOR 2333419.
• • L. E. Maistrov (2014). Probability Theory: A Historical Sketch. Elsevier Science. p. 1. ISBN 978-1-4832-1863-2.
• • Seneta, E. (2006). "Probability, History of". Encyclopedia of Statistical Sciences. p. 1. doi:10.1002/0471667196.ess2065.pub2. ISBN 978-0471667193.
• • John Tabak (2014). Probability and Statistics: The Science of Uncertainty. Infobase Publishing. pp. 24–26. ISBN 978-0-8160-6873-9.
• • Bellhouse, David (2005). "Decoding Cardano's Liber de Ludo Aleae". Historia Mathematica. 32 (2): 180–202. doi:10.1016/j.hm.2004.04.001. ISSN 0315-0860.
• • Anders Hald (2005). A History of Probability and Statistics and Their Applications before 1750. John Wiley & Sons. p. 221. ISBN 978-0-471-72517-6.
• • L. E. Maistrov (2014). Probability Theory: A Historical Sketch. Elsevier Science. p. 56. ISBN 978-1-4832-1863-2.
• • John Tabak (2014). Probability and Statistics: The Science of Uncertainty. Infobase Publishing. p. 37. ISBN 978-0-8160-6873-9.
• • Chung, Kai Lai (1998). "Probability and Doob". The American Mathematical Monthly. 105 (1): 28–35. doi:10.2307/2589523. ISSN 0002-9890. JSTOR 2589523.
• • Bingham, N. (2000). "Studies in the history of probability and statistics XLVI. Measure into probability: from Lebesgue to Kolmogorov". Biometrika. 87 (1): 145–156. doi:10.1093/biomet/87.1.145. ISSN 0006-3444.
• • Benzi, Margherita; Benzi, Michele; Seneta, Eugene (2007). "Francesco Paolo Cantelli. b. 20 December 1875 d. 21 July 1966". International Statistical Review. 75 (2): 128. doi:10.1111/j.1751-5823.2007.00009.x. ISSN 0306-7734. S2CID 118011380.
• • Doob, Joseph L. (1996). "The Development of Rigor in Mathematical Probability (1900-1950)". The American Mathematical Monthly. 103 (7): 586–595. doi:10.2307/2974673. ISSN 0002-9890. JSTOR 2974673.
• • Cramer, Harald (1976). "Half a Century with Probability Theory: Some Personal Recollections". The Annals of Probability. 4 (4): 509–546. doi:10.1214/aop/1176996025. ISSN 0091-1798.
• • Truesdell, C. (1975). "Early kinetic theories of gases". Archive for History of Exact Sciences. 15 (1): 22–23. doi:10.1007/BF00327232. ISSN 0003-9519. S2CID 189764116.
• • Brush, Stephen G. (1967). "Foundations of statistical mechanics 1845?1915". Archive for History of Exact Sciences. 4 (3): 150–151. doi:10.1007/BF00412958. ISSN 0003-9519. S2CID 120059181.
• • Truesdell, C. (1975). "Early kinetic theories of gases". Archive for History of Exact Sciences. 15 (1): 31–32. doi:10.1007/BF00327232. ISSN 0003-9519. S2CID 189764116.
• • Brush, S.G. (1958). "The development of the kinetic theory of gases IV. Maxwell". Annals of Science. 14 (4): 243–255. doi:10.1080/00033795800200147. ISSN 0003-3790.
• • Brush, Stephen G. (1968). "A history of random processes". Archive for History of Exact Sciences. 5 (1): 15–16. doi:10.1007/BF00328110. ISSN 0003-9519. S2CID 117623580.
• • Kendall, D. G.; Batchelor, G. K.; Bingham, N. H.; Hayman, W. K.; Hyland, J. M. E.; Lorentz, G. G.; Moffatt, H. K.; Parry, W.; Razborov, A. A.; Robinson, C. A.; Whittle, P. (1990). "Andrei Nikolaevich Kolmogorov (1903–1987)". Bulletin of the London Mathematical Society. 22 (1): 33. doi:10.1112/blms/22.1.31. ISSN 0024-6093.
• • Vere-Jones, David (2006). "Khinchin, Aleksandr Yakovlevich". Encyclopedia of Statistical Sciences. p. 1. doi:10.1002/0471667196.ess6027.pub2. ISBN 978-0471667193.

- • Vere-Jones, David (2006). "Khinchin, Aleksandr Yakovlevich". Encyclopedia of Statistical Sciences. p. 4. doi:10.1002/0471667196.ess6027.pub2. ISBN 978-0471667193.
- • Snell, J. Laurie (2005). "Obituary: Joseph Leonard Doob". Journal of Applied Probability. 42 (1): 251. doi:10.1239/jap/1110381384. ISSN 0021-9002.
- • Lindvall, Torgny (1991). "W. Doeblin, 1915-1940". The Annals of Probability. 19 (3): 929–934. doi:10.1214/aop/1176990329. ISSN 0091-1798.
- • Getoor, Ronald (2009). "J. L. Doob: Foundations of stochastic processes and probabilistic potential theory". The Annals of Probability. 37 (5): 1655. arXiv:0909.4213. Bibcode:2009arXiv0909.4213G. doi:10.1214/09-AOP465. ISSN 0091-1798. S2CID 17288507.
- • Bingham, N. H. (2005). "Doob: a half-century on". Journal of Applied Probability. 42 (1): 257–266. doi:10.1239/jap/1110381385. ISSN 0021-9002.
- • Meyer, Paul-André (2009). "Stochastic Processes from 1950 to the Present". Electronic Journal for History of Probability and Statistics. 5 (1): 1–42.
- • "Kiyosi Itô receives Kyoto Prize". Notices of the AMS. 45 (8): 981–982. 1998.
- • Jean Bertoin (1998). Lévy Processes. Cambridge University Press. p. viii and ix. ISBN 978-0-521-64632-1.
- • J. Michael Steele (2012). Stochastic Calculus and Financial Applications. Springer Science & Business Media. p. 176. ISBN 978-1-4684-9305-4.
- • P. Hall; C. C. Heyde (2014). Martingale Limit Theory and Its Application. Elsevier Science. pp. 1, 2. ISBN 978-1-4832-6322-9.
- • Dynkin, E. B. (1989). "Kolmogorov and the Theory of Markov Processes". The Annals of Probability. 17 (3): 822–832. doi:10.1214/aop/1176991248. ISSN 0091-1798.
- • Ellis, Richard S. (1995). "An overview of the theory of large deviations and applications to statistical mechanics". Scandinavian Actuarial Journal. 1995 (1): 98. doi:10.1080/03461238.1995.10413952. ISSN 0346-1238.
- • Raussen, Martin; Skau, Christian (2008). "Interview with Srinivasa Varadhan". Notices of the AMS. 55 (2): 238–246.
- • Malte Henkel; Dragi Karevski (2012). Conformal Invariance: an Introduction to Loops, Interfaces and Stochastic Loewner Evolution. Springer Science & Business Media. p. 113. ISBN 978-3-642-27933-1.
- • "2006 Fields Medals Awarded". Notices of the AMS. 53 (9): 1041–1044. 2015.
- • Quastel, Jeremy (2015). "The Work of the 2014 Fields Medalists". Notices of the AMS. 62 (11): 1341–1344.
- • D.J. Daley; D. Vere-Jones (2006). An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods. Springer Science & Business Media. pp. 1–4. ISBN 978-0-387-21564-8.
- • Anders Hald (2005). A History of Probability and Statistics and Their Applications before 1750. John Wiley & Sons. p. 226. ISBN 978-0-471-72517-6.
- • Joel Louis Lebowitz (1984). Nonequilibrium phenomena II: from stochastics to hydrodynamics. North-Holland Pub. pp. 8–10. ISBN 978-0-444-86806-0.
- • Ionut Florescu (2014). Probability and Stochastic Processes. John Wiley & Sons. p. 374. ISBN 978-1-118-59320-2.
- • Oliver C. Ibe (2013). Elements of Random Walk and Diffusion Processes. John Wiley & Sons. p. 5. ISBN 978-1-118-61793-9.
- • Anders Hald (2005). A History of Probability and Statistics and Their Applications before 1750. John Wiley & Sons. p. 63. ISBN 978-0-471-72517-6.
- • Anders Hald (2005). A History of Probability and Statistics and Their Applications before 1750. John Wiley & Sons. p. 202. ISBN 978-0-471-72517-6.
- • Ionut Florescu (2014). Probability and Stochastic Processes. John Wiley & Sons. p. 385. ISBN 978-1-118-59320-2.
- • Barry D. Hughes (1995). Random Walks and Random Environments: Random walks. Clarendon Press. p. 111. ISBN 978-0-19-853788-5.
- • Thiele, Thorwald N. (1880). "Om Anvendelse af mindste Kvadraterbs Methode i nogle Tilfælde, hvoren Komplikation af visse Slags uensartede tilfældige Fejlkilder giver Fejleneen "systematisk"

Karakter". Kongelige Danske Videnskabernes Selskabs Skrifter. Series 5 (12): 381–408.
• • Hald, Anders (1981). "T. N. Thiele's Contributions to Statistics". International Statistical Review / Revue Internationale de Statistique. 49 (1): 1–20. doi:10.2307/1403034. ISSN 0306-7734. JSTOR 1403034.
• • Lauritzen, Steffen L. (1981). "Time Series Analysis in 1880: A Discussion of Contributions Made by T.N. Thiele". International Statistical Review / Revue Internationale de Statistique. 49 (3): 319–320. doi:10.2307/1402616. ISSN 0306-7734. JSTOR 1402616.
• • Bachelier, Luis (1900). "Théorie de la spéculation" (PDF). Ann. Sci. Éc. Norm. Supér. Serie 3, 17: 21–89. doi:10.24033/asens.476. Archived (PDF) from the original on 2011-06-05.
• • Bachelier, Luis (1900). "The Theory of Speculation". Ann. Sci. Éc. Norm. Supér. Serie 3, 17: 21–89 (Engl. translation by David R. May, 2011). doi:10.24033/asens.476.
• • Courtault, Jean-Michel; Kabanov, Yuri; Bru, Bernard; Crepel, Pierre; Lebon, Isabelle; Le Marchand, Arnaud (2000). "Louis Bachelier on the Centenary of Theorie de la Speculation" (PDF). Mathematical Finance. 10 (3): 339–353. doi:10.1111/1467-9965.00098. ISSN 0960-1627. S2CID 14422885. Archived (PDF) from the original on 2018-07-21.
• • Jovanovic, Franck (2012). "Bachelier: Not the forgotten forerunner he has been depicted as. An analysis of the dissemination of Louis Bachelier's work in economics" (PDF). The European Journal of the History of Economic Thought. 19 (3): 431–451. doi:10.1080/09672567.2010.540343. ISSN 0967-2567. S2CID 154003579. Archived (PDF) from the original on 2018-07-21.
• • Brush, Stephen G. (1968). "A history of random processes". Archive for History of Exact Sciences. 5 (1): 25. doi:10.1007/BF00328110. ISSN 0003-9519. S2CID 117623580.
• • Brush, Stephen G. (1968). "A history of random processes". Archive for History of Exact Sciences. 5 (1): 1–36. doi:10.1007/BF00328110. ISSN 0003-9519. S2CID 117623580.
• • D.J. Daley; D. Vere-Jones (2006). An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods. Springer Science & Business Media. pp. 8–9. ISBN 978-0-387-21564-8.
• • Embrechts, Paul; Frey, Rüdiger; Furrer, Hansjörg (2001). "Stochastic processes in insurance and finance". Stochastic Processes: Theory and Methods. Handbook of Statistics. Vol. 19. p. 367. doi:10.1016/S0169-7161(01)19014-0. ISBN 978-0444500144. ISSN 0169-7161.
• • Cramér, Harald (1969). "Historical review of Filip Lundberg's works on risk theory". Scandinavian Actuarial Journal. 1969 (sup3): 6–12. doi:10.1080/03461238.1969.10404602. ISSN 0346-1238.
• • Charles Miller Grinstead; James Laurie Snell (1997). Introduction to Probability. American Mathematical Soc. pp. 464–466. ISBN 978-0-8218-0749-1.
• • Pierre Bremaud (2013). Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues. Springer Science & Business Media. p. ix. ISBN 978-1-4757-3124-8.
• • Hayes, Brian (2013). "First links in the Markov chain". American Scientist. 101 (2): 92–96. doi:10.1511/2013.101.92.
• • Seneta, E. (1998). "I.J. Bienaymé [1796-1878]: Criticality, Inequality, and Internationalization". International Statistical Review / Revue Internationale de Statistique. 66 (3): 291–292. doi:10.2307/1403518. ISSN 0306-7734. JSTOR 1403518.
• • Bru, B.; Hertz, S. (2001). "Maurice Fréchet". Statisticians of the Centuries. pp. 331–334. doi:10.1007/978-1-4613-0179-0_71. ISBN 978-0-387-95283-3.
• • Marc Barbut; Bernard Locker; Laurent Mazliak (2016). Paul Lévy and Maurice Fréchet: 50 Years of Correspondence in 107 Letters. Springer London. p. 5. ISBN 978-1-4471-7262-8.
• • Valeriy Skorokhod (2005). Basic Principles and Applications of Probability Theory. Springer Science & Business Media. p. 146. ISBN 978-3-540-26312-8.
• • Bernstein, Jeremy (2005). "Bachelier". American Journal of Physics. 73 (5): 398–396. Bibcode:2005AmJPh..73..395B. doi:10.1119/1.1848117. ISSN 0002-9505.
• • William J. Anderson (2012). Continuous-Time Markov Chains: An Applications-Oriented Approach. Springer Science & Business Media. p. vii. ISBN 978-1-4612-3038-0.
• • Kendall, D. G.; Batchelor, G. K.; Bingham, N. H.; Hayman, W. K.; Hyland, J. M. E.; Lorentz, G. G.; Moffatt, H. K.; Parry, W.; Razborov, A. A.; Robinson, C. A.; Whittle, P. (1990). "Andrei Nikolaevich Kolmogorov (1903–1987)". Bulletin of the London Mathematical Society. 22 (1): 57.

doi:10.1112/blms/22.1.31. ISSN 0024-6093.
• • David Applebaum (2004). Lévy Processes and Stochastic Calculus. Cambridge University Press. p. 67. ISBN 978-0-521-83263-2.
• • Robert J. Adler (2010). The Geometry of Random Fields. SIAM. p. 13. ISBN 978-0-89871-693-1.
• • Krishna B. Athreya; Soumendra N. Lahiri (2006). Measure Theory and Probability Theory. Springer Science & Business Media. ISBN 978-0-387-32903-1.
• • Bernt Øksendal (2003). Stochastic Differential Equations: An Introduction with Applications. Springer Science & Business Media. p. 11. ISBN 978-3-540-04758-2.
• • David Williams (1991). Probability with Martingales. Cambridge University Press. p. 124. ISBN 978-0-521-40605-5.
• • Rick Durrett (2010). Probability: Theory and Examples. Cambridge University Press. p. 410. ISBN 978-1-139-49113-6.
• • Patrick Billingsley (2008). Probability and Measure. Wiley India Pvt. Limited. pp. 493–494. ISBN 978-81-265-1771-8.
• • Alexander A. Borovkov (2013). Probability Theory. Springer Science & Business Media. pp. 529–530. ISBN 978-1-4471-5201-9.
• • Krishna B. Athreya; Soumendra N. Lahiri (2006). Measure Theory and Probability Theory. Springer Science & Business Media. p. 221. ISBN 978-0-387-32903-1.
• • Robert J. Adler; Jonathan E. Taylor (2009). Random Fields and Geometry. Springer Science & Business Media. p. 14. ISBN 978-0-387-48116-6.
• • Krishna B. Athreya; Soumendra N. Lahiri (2006). Measure Theory and Probability Theory. Springer Science & Business Media. p. 211. ISBN 978-0-387-32903-1.
• • Alexander A. Borovkov (2013). Probability Theory. Springer Science & Business Media. p. 536. ISBN 978-1-4471-5201-9.
• • Benjamin Yakir (2013). Extremes in Random Fields: A Theory and Its Applications. John Wiley & Sons. p. 5. ISBN 978-1-118-72062-2.
• • Black, Fischer; Scholes, Myron (1973). "The Pricing of Options and Corporate Liabilities". Journal of Political Economy. 81 (3): 637–654. doi:10.1086/260062. ISSN 0022-3808. JSTOR 1831029.
• • Merton, Robert C. (July 2005), "Theory of rational option pricing", Theory of Valuation (2 ed.), WORLD SCIENTIFIC, pp. 229–288, doi:10.1142/9789812701022_0008, ISBN 978-981-256-374-3, retrieved 2024-09-30
• • Heston, Steven L. (1993). "A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options". The Review of Financial Studies. 6 (2): 327–343. doi:10.1093/rfs/6.2.327. ISSN 0893-9454. JSTOR 2962057.
• • Ross, Sheldon M. (2010). Introduction to probability models (10th ed.). Amsterdam Heidelberg: Elsevier. ISBN 978-0-12-375686-2.
323. • Ross, Sheldon M. (2010). Introduction to probability models (10th ed.). Amsterdam Heidelberg: Elsevier. ISBN 978-0-12-375686-2.

Further reading

This "Further reading" section may need cleanup. Please read the editing guide and help improve the section. (July 2018) (Learn how and when to remove this message)

Articles

• Applebaum, David (2004). "Lévy processes: From probability to finance and quantum groups". Notices of the AMS. 51 (11): 1336–1347.
• Cramer, Harald (1976). "Half a Century with Probability Theory: Some Personal Recollections". The Annals of Probability. 4 (4): 509–546. doi:10.1214/aop/1176996025. ISSN 0091-1798.
• Guttorp, Peter; Thorarinsdottir, Thordis L. (2012). "What Happened to Discrete Chaos, the Quenouille Process, and the Sharp Markov Property? Some History of Stochastic Point Processes". International Statistical Review. 80 (2): 253–268. doi:10.1111/j.1751-5823.2012.00181.x. ISSN 0306-7734. S2CID 80836.
• Jarrow, Robert; Protter, Philip (2004). "A short history of stochastic integration and mathematical finance: the early years, 1880–1970". A Festschrift for Herman Rubin. Institute of Mathematical Statistics Lecture Notes - Monograph Series. pp. 75–91. doi:10.1214/lnms/1196285381. ISBN 978-0-

940600-61-4. ISSN 0749-2170.
• Meyer, Paul-André (2009). "Stochastic Processes from 1950 to the Present". Electronic Journal for History of Probability and Statistics. 5 (1): 1–42.

Books

• Robert J. Adler (2010). The Geometry of Random Fields. SIAM. ISBN 978-0-89871-693-1.
• Robert J. Adler; Jonathan E. Taylor (2009). Random Fields and Geometry. Springer Science & Business Media. ISBN 978-0-387-48116-6.
• Pierre Brémaud (2013). Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues. Springer Science & Business Media. ISBN 978-1-4757-3124-8.
• Joseph L. Doob (1990). Stochastic processes. Wiley.
• Anders Hald (2005). A History of Probability and Statistics and Their Applications before 1750. John Wiley & Sons. ISBN 978-0-471-72517-6.
• Crispin Gardiner (2010). Stochastic Methods. Springer. ISBN 978-3-540-70712-7.
• Iosif I. Gikhman; Anatoly Vladimirovich Skorokhod (1996). Introduction to the Theory of Random Processes. Courier Corporation. ISBN 978-0-486-69387-3.
• Emanuel Parzen (2015). Stochastic Processes. Courier Dover Publications. ISBN 978-0-486-79688-8.
• Murray Rosenblatt (1962). Random Processes. Oxford University Press.

External links

• Media related to Stochastic processes at Wikimedia Commons

• v
• t
• e

Stochastic processes

• v
• t
• e

Industrial and applied mathematics

Authority control databases

Categories:
• Stochastic processes
• Stochastic models
• Statistical data types

Full text links

Actions
Share
•
•
•

•
Page navigation

Review
J Healthc Eng
•
•
•
. 2018 Apr 30:2018:3182483.
doi: 10.1155/2018/3182483. eCollection 2018.
Optical Ultrasound Generation and Detection for Intravascular Imaging: A Review
Tianrui Zhao 1 , Lei Su 1 , Wenfeng Xia 2 3
Affiliations
• PMID: 29854358
• PMCID: PMC5952521
• DOI: 10.1155/2018/3182483
Abstract
Combined ultrasound and photoacoustic imaging has attracted significant interests for intravascular imaging such as atheromatous plaque detection, with ultrasound imaging providing spatial location and morphology and photoacoustic imaging highlighting molecular composition of the plaque. Conventional ultrasound imaging systems utilize piezoelectric ultrasound transducers, which suffer from limited frequency bandwidths and reduced sensitivity with miniature transducer elements. Recent advances on optical methods for both ultrasound generation and detection have shown great promise, as they provide efficient and ultrabroadband ultrasound generation and sensitive and ultrabroadband ultrasound detection. As such, all-optical ultrasound imaging has a great potential to become a next generation ultrasound imaging method. In this paper, we review recent developments on optical ultrasound transmitters, detectors, and all-optical ultrasound imaging systems, with a particular focus on fiber-based probes for intravascular imaging. We further discuss our thoughts on future directions on developing combined all-optical photoacoustic and ultrasound imaging systems for intravascular imaging.

Figures

Figure 1
Examples of IVPA and IVUS…

Figure 2
SEM images of multilayer CNTs-PDMS…

Figure 3
(a) SEM images of the…

Figure 4
SEM images of CNTs-PDMS-coated optical…

Figure 5
Major embodiments of fiber optic…

Figure 6
The principle for a FBG-based…

Similar articles

- **Intravascular photoacoustic imaging of exogenously labeled atherosclerotic plaque through luminal blood.**
Yeager D, Karpiouk A, Wang B, Amirian J, Sokolov K, Smalling R, Emelianov S. J Biomed Opt. 2012 Oct;17(10):106016. doi: 10.1117/1.JBO.17.10.106016. PMID: 23224013 Free PMC article.
- **Optical ultrasound sensors for photoacoustic imaging: a review.**
Zhu L, Cao H, Ma J, Wang L. J Biomed Opt. 2024 Jan;29(Suppl 1):S11523. doi: 10.1117/1.JBO.29.S1.S1

B3
Schneider Electric - Network Protection & Automation Guide 134
Industrial & Commercial Power System Protection
5. Industrial circuit breakers

a consequent risk of injury to the operator. Such types may be required to be replaced with modern equivalents. ACBs are normally fitted with integral overcurrent protection, thus avoiding the need for separate protection devices. However, the operating time characteristics of the integral protection are often designed to make discrimination with MCBs/MCCBs/fuses easier and so they may not be in accordance with the standard dependent time characteristics given in IEC 60255-3. Therefore, problems in co-ordination with discrete protection relays may still arise, but modern numerical relays have more flexible characteristics to alleviate such difficulties. ACBs will also have facilities for accepting an external trip signal, and this can be used in conjunction with an external relay if desired. Figure B3.6 illustrates the typical tripping characteristics available.

5.4 Oil Circuit Breakers (OCBs)

Oil circuit breakers have been very popular for many years for industrial supply systems at voltages of 3.3kV and above. They are found in both ‘bulk oil’ and ‘minimum oil’ types, the only significant difference being the volume of oil in the tank.

In this type of breaker, the main contacts are housed in an oil-filled tank, with the oil acting as the both the insulation and the arc-quenching medium. The arc produced during contact separation under fault conditions causes dissociation of the hydrocarbon insulating oil into hydrogen and carbon. The hydrogen extinguishes the arc. The carbon produced mixes with the oil. As the carbon is conductive, the oil must be changed after a prescribed number of fault clearances, when the degree of contamination reaches an unacceptable level.

Because of the fire risk involved with oil, precautions such as the construction of fire/blast walls may have to be taken when

OCBs are installed.
5.5 Vacuum Circuit Breakers (VCBs)
Since the introduction of vacuum switching technology in the 1960's, Vacuum switchgear has all but replaced Air Circuit Breaker (ACBs) and Oil Circuit Breaker (OCBs) at medium voltage levels. Vacuum switchgear is rated for fault level up to 63kA with continuous ratings of greater than 5000A.
The vacuum interrupter is a compact, inherently reliable and maintenance free device with an expected life of more than 10,000 operations and is capable of interrupting full fault currents up to 100 times.
These characteristics have resulted in a dramatic reduction in switchgear maintenance compared to ACBs or OCBs and are used in a wide range of applications, including Distribution networks and medium to large industry.
The reduction in maintenance requirements and smaller dimensions have allowed the configuration of switchgear to be adapted from the conventional withdrawable pattern to a fixed pattern Air Insulated Switchgear (AIS, See Fig B3.7). Fixed pattern switchgear is generally more compact, easier to install and has simpler operation.
0.01
1 10
Current (multiple of setting)
Time (s)
Inverse
Very Inverse
Ultra Inverse
Short Circuit
0.1
1
10
100
1000
20
Figure B3.6:
Typical tripping characteristics of an ACB
Figure B3.7:
Typical air insulated vacuum contactor switchgear
B3
Schneider Electric - Network Protection & Automation Guide135Industrial & Commercial Power System Protection
6. Protection relays
5. Industrial circuit breakers
When the circuit breaker itself does not have integral protection, then a suitable external relay will have to be provided. For an industrial system, the most common protection relays are time- delayed overcurrent and earth fault relays. Chapter [C1: Overcurrent Protection for Phase and Earth Faults] provides details of the application of overcurrent relays.
Traditionally, for three wire systems, overcurrent relays have often been applied to two phases only for relay element economy. Even with modern multi-element relay designs, economy is still a consideration in terms of the number of analogue current inputs that have to be provided. Two

overcurrent elements will detect any interphase fault, so it is conventional to apply two elements on the same phases at all relay locations. The phase CT residual current connections for an earth fault relay element are unaffected by this convention. Figure B3.9 illustrates the possible relay connections and limitations on settings.

The fixed pattern is also available in a gas insulated configuration (GIS) where the Vacuum Interrupter and main current carrying parts are insulated with SF6 gas. This further enhances the compact nature of the design. Typically 36kV GIS has similar dimensions to 12kV AIS (See Figure B3.8).

Gas Insulated Switchgear is normally found in higher voltage applications. i.e. 24kV and above.

A variation of vacuum switchgear is the vacuum contactor. This device has a limited fault interrupting rating and is used in conjunction with High Rupturing Capacity (HRC) fuses, The contactor has a very high operating duty – up to 1 million operations, and is typically used to switch MV motors.

5.6 SF6 circuit breakers

Circuit breakers using SF6 gas as the arc-quenching medium are also available and in some countries and for some applications are preferred. Generally these have similar ratings to those of vacuum switchgear and in some cases can be incorporated into the same cubicle as vacuum circuit breakers.

5.7 Improved safety

Changes in International Standards have resulted in improvements to operator safety. One area is Internal Arc (Arc Flash) protection. Many switchgear designs have passive protection ‘built in’ and are capable of controlling the effects of an internal arc fault even at the highest fault levels available. To supplement this, or to improve the performance of existing switchgear, active solutions, which detect the occurrence of an arc fault and then initiate the disconnection of the supply, are available in conjunction with protection relay systems. For more details on arc protection solutions please refer to Chapter [C11: Arc Protection].

Figure B3.8:
Typical gas insulated vacuum contactor switchgear

B3

6. Protection relays

Figure B3.9:
Overcurrent and earth fault relay connections

CT
connections
Phase
elements
Residual
current
elements
System Type of
fault Notes
(a)

A B C
3Ph. 3w Ph. - Ph. Petersen coil and
unearthed systems
(b)
CBA
3Ph. 3w (a) Ph. - Ph.
(b) Ph. - E* * Earth-fault protection
only if earth-fault
current is not less
than twice primary
operating
current(c) 3Ph. 4w
(a) Ph. - Ph.
(b) Ph. - E*
(c) Ph. - N
(d)
A B C
3Ph. 3w (a) Ph. - Ph.
(b) Ph. - E
Phase elements must
be in same phases at
all stations.
Earth-fault settings
may be less than full
load
(e)
B CA
3Ph. 3w (a) Ph. - Ph.
(b) Ph. - E
Earth-fault settings
may be less than full
load
(f) 3Ph. 4w
(a) Ph. - Ph.
(b) Ph. - E
(c) Ph. - N
Earth-fault settings
may be less than full
load, but must be
greater than largest
Ph. - N load
(g)
A B C N
3Ph. 4w
(a) Ph. - Ph.
(b) Ph. - E
(c) Ph. - N
Earth-fault settings
may be less than full
load
(h)
A B C N
3Ph. 3w
or

3Ph. 4w
Ph. - E
Earth-fault settings
may be less than full
load
Ph. = phase ; w = wire ; E = earth ; N = neutral
B3
Schneider Electric - Network Protection & Automation Guide141
Industrial & Commercial Power System Protection
Loads such as induction motors draw significant reactive power from the supply system, and a poor overall power factor may result. The flow of reactive power increases the voltage-drops through series reactances such as transformers and reactors, it uses up some of the current carrying capacity of power system plant and it increases the resistive losses in the power system.
To offset the losses and restrictions in plant capacity they incur and to assist with voltage regulation, Utilities usually apply tariff penalties to large industrial or commercial customers for running their plant at excessively low power factor. The customer is thereby induced to improve the power factor of his system and it may be cost-effective to install fixed or variable power factor correction equipment to raise or regulate the plant power factor to an acceptable level.
Shunt capacitors are often used to improve power factor. The basis for compensation is illustrated in Figure B3.14, where ∠φ1 represents the uncorrected power factor angle and ∠φ2 the angle relating to the desired power factor, after correction.
The following may be deduced from this vector diagram:
a. Uncorrected power factor = kW
kVA 1
= cos ∠φ1
b. Corrected power factor = kW
kVA 2
= cos ∠φ2
c. Reduction in kVA = kVA 1 - kVA 2
10. Voltage and phase reversal protection
11. Power factor correction and protection of capacitors
Magnetising kvar
kW
kVA 2 load current
with compensation
Compensating kvar
kVA1 load current without
compensation
V
Capacitor kvar
Figure B3.14:
Power factor correction principle
Voltage relays have been widely used in industrial power supply systems. The principle purposes are to detect undervoltage and/or overvoltage conditions at switchboards to disconnect supplies before damage can be caused from these conditions or to provide interlocking checks. Prolonged

overvoltage may cause damage to voltage-sensitive equipment (e.g. electronics), while undervoltage may cause excessive current to be drawn by motor loads. Motors are provided with thermal overload protection to prevent damage with excessive current, but undervoltage protection is commonly applied to disconnect motors after a prolonged voltage dip. With a voltage dip caused by a source system fault, a group of motors could decelerate to such a degree that their aggregate re-acceleration currents might keep the recovery voltage depressed to a level where the machines might stall. Modern numerical motor protection relays typically incorporate voltage protection functions, thus removing the need for discrete undervoltage relays for this purpose. See Chapter [C.9: A.C. Motor Protection]. Older installations may still utilise discrete undervoltage relays, but the setting criteria remain the same.
Reverse phase sequence voltage protection should be applied where it may be dangerous for a motor to be started with rotation in the opposite direction to that intended. Incorrect rotation due to reverse phase sequence might be set up following some error after power system maintenance or repairs, e.g. to a supply cable. Older motor control boards might have been fitted with discrete relays to detect this condition. Modern motor protection relays may incorporate this function. If reverse phase sequence is detected, motor starting can be blocked. If reverse phase sequence voltage protection is not provided, the high-set negative phase sequence current protection in the relay would quickly detect the condition once the starting device is closed – but initial reverse rotation of the motor could not be prevented.

If the kW load and uncorrected power factors are known, then the capacitor rating in kvar to achieve a given degree of correction may be calculated from:

Capacitor kvar = kW x ($\tan\angle\varphi 1$ - $\tan\angle\varphi 2$)

A spreadsheet can easily be constructed to calculate the required amount of compensation to achieve a desired power factor.

11.1 Capacitor control

Where the plant load or the plant power factor varies considerably, it is necessary to control the power factor correction, since over-correction will result in excessive system voltage and unnecessary losses. In a few industrial systems, capacitors are switched in manually when required, but automatic controllers are standard practice. A controller provides automatic power factor correction, by comparing the running power factor with the target value. Based on the available groupings, an appropriate amount of capacitance is switched in or out to maintain an optimum average power factor. The controller is fitted with a ’loss of voltage’ relay element to ensure that all selected capacitors are disconnected instantaneously if there is a supply voltage interruption. When the supply voltage is restored, the capacitors are reconnected

progressively as the plant starts up. To ensure that capacitor groups degrade at roughly the same rate, the controller usually rotates selection or randomly selects groups of the same size in order to even out the connected time. The provision of overvoltage protection to trip the capacitor bank is also desirable in some applications. This would be to prevent a severe system overvoltage if the power factor correction (PFC) controller fails to take fast corrective action.
The design of PFC installations must recognise that many industrial loads generate harmonic voltages, with the result that the PFC capacitors may sink significant harmonic currents. A harmonic study may be necessary to determine the capacitor thermal ratings or whether series filters are required.

11.2 Motor power factor correction

When dealing with power factor correction of motor loads, group correction is not always the most economical method. Some industrial consumers apply capacitors to selected motor substations rather than applying all of the correction at the main incoming substation busbars. Sometimes, power factor correction may even be applied to individual motors, resulting in optimum power factor being obtained under all conditions of aggregate motor load. In some instances, better motor starting may also result, from the improvement in the voltage regulation due to the capacitor. Motor capacitors are often six-terminal units, and a capacitor may be conveniently connected directly across each motor phase winding.
Capacitor sizing is important, such that a leading power factor does not occur under any load condition. If excess capacitance is applied to a motor, it may be possible for self-excitation to occur when the motor is switched off or suffers a supply

failure. This can result in the production of a high voltage or in mechanical damage if there is a sudden restoration of supply. Since most star/delta or auto-transformer starters other than the ‘Korndorffer’ types involve a transitional break in supply, it is generally recommended that the capacitor rating should not exceed 85% of the motor magnetising reactive power.

11.3 Capacitor protection

When considering protection for capacitors, allowance should be made for the transient inrush current occurring on switch-on, since this can reach peak values of around 20 times normal current. Switchgear for use with capacitors is usually de-rated considerably to allow for this. Inrush currents may be limited by a resistor in series with each capacitor or bank of capacitors.
Protection equipment is required to prevent rupture of the capacitor due to an internal fault and also to protect the cables and associated equipment from damage in case of a capacitor failure. If fuse protection is contemplated for a three-phase capacitor, HRC fuses should be employed with a current rating of not less than 1.5 times the rated capacitor current.
Medium voltage capacitor banks can be protected by the scheme shown in Figure B3.15. Since harmonics increase

capacitor current, the relay will respond more correctly if it does not have in-built tuning for harmonic rejection.
Double star capacitor banks are employed at medium voltage. As shown in Figure B3.16, a current transformer in the inter star-point connection can be used to drive a protection relay to detect the out-of-balance currents that will flow when capacitor elements become short-circuited or open-circuited. The relay will have adjustable current settings, and it might contain a bias circuit, fed from an external voltage transformer, that can be adjusted to compensate for steady-state spill current in the inter star-point connection.
Some industrial loads such as arc furnaces involve large inductive components and correction is often applied using very large high voltage capacitors in various configurations. Another high voltage capacitor configuration is the 'split phase' arrangement where the elements making up each phase of the capacitor are split into two parallel paths. Figure B3.17 shows two possible connection methods for the relay. A differential relay can be applied with a current transformer for each parallel branch, comparing the currents in the split phases. Alternatively an overcurrent relay can be applied with a current transformer in the bridge link, where normally no current should flow. Both relays use sensitive current settings but also adjustable compensation for the unbalance currents arising from initial capacitor mismatch.
The difference of current through the split phases or the increase of the current through the bridge link indicates a defect of a single capacitor unit 'can' where the amount of additional current is directly dependent on the can dimensions.
less than just after fault occurrence. In rare situations, it may have to be taken into account for correct time grading for through-fault protection considerations, and in the peak voltage calculation for high impedance differential protection configurations.
It is more important to take motor contribution into account when considering equipment fault rating (busbars, cables, switchgear, etc.). Typically, the initial AC motor current component at the instant of fault is of similar magnitude to the direct-on-line motor starting current. For LV motors, 5xFLC is assumed as the common fault current contribution (after considering the effect of motor cable impedance), with 5.5xFLC for HV motors, unless it is known that low starting current HV motors are installed. It is also common that similar motors connected to a busbar can be lumped together as one equivalent motor. In doing so, motor rated speed may need to be considered, as 2 or 4 pole motors have a longer fault current decay than motors with a higher number of poles. The kVA rating of the single equivalent motor is taken as the sum of the kVA ratings of the individual motors. It is still possible for motor contribution to be ignored in situations where the motor load on a busbar is insignificant in comparison to the total load (again IEC 60909 gives guidance in this respect). Nevertheless, big LV motor loads and all HV motors should be considered when calculating fault levels.

AUTOMATIC CHANGEOVER SYSTEMS

Induction motors are typically used to drive critical loads. In some plants, such as those involving the pumping of fluids and gases, this has led to the need for a power supply control configuration in which motor and other loads are automatically transferred on loss of the normal supply to an optional supply. A fast changeover, allowing the motor load to be re-accelerated, decreases the possibility of a process trip happening. Such configurations are typically used for big generating units to transfer unit loads from the

unit transformer to the station supply/start-up transformer. When the normal supply fails, induction motors that stay connected to the busbar slow down and the trapped rotor flux creates a residual voltage that exponentially decays. All motors connected to a busbar will start to decelerate at the same rate when the supply is lost if they stay connected to the busbar. This happens because the motors will exchange energy between themselves, so that they tend to stay 'synchronized' to each other.

Consequently, the residual voltages of all the motors decay at nearly the same rate. The magnitude of this voltage and its phase displacement with respect to the healthy alternative supply voltage is a function of time and the motor speed. The angular displacement between the residual motor voltage and the incoming voltage will be 180° at some instant. If the healthy back-up supply is switched on to motors which are running down under these conditions, very high inrush currents may happen, generating stresses which could be of adequate magnitude to cause mechanical stress, as well as a severe dip in the back-up supply voltage.

Two automatic transfer methods are applied:

- in-phase transfer system
- residual voltage system

The in-phase transfer method is presented in Figure 10(a). Normal and standby feeders from the same power source are used.

Phase angle measurement is applied to discover the relative phase angle between the standby feeder voltage and the motor busbar voltage. When the voltages are in phase or just prior to this condition, a high speed circuit breaker is used to complete the transfer. This approach is restricted to big high inertia drives where the gradual run down characteristic upon loss of normal feeder supply can be accurately anticipated.

Figure 10(b) presents the residual voltage method, which is more typical, particularly in the petrochemical industry. Two feeders are used, feeding two busbar sections connected by a normally open bus section breaker. Each line is capable of transferring the total busbar load. Each bus section voltage is monitored and loss of supply on either section causes the relevant incomer CB to open. Given there are no protection operations to show the presence of a busbar fault, the bus section breaker is automatically closed to restore the supply to the unpowered busbar section of after the residual voltage created by the motors running down on that section has decreased to a an acceptable level.

Figure 10. Auto-transfer systems (a) In phase transfer approach (b) Residual voltage approach

This is between 25% and 40%, of nominal voltage, dependent on the power system characteristics. The residual voltage setting selection will affect the re-acceleration current after the bus section breaker closes. For example, a setting of 25% may be anticipated to result in an inrush current of around 125% of the starting current at full voltage. Optionally, a time delay could be used as a residual voltage measurement substitute, which would be set with knowledge of the plant to make sure that the residual voltage would have sufficiently decreased before transfer is started. The protection relay settings for the switchboard must take account of the total load current

M M M M
Feeder No.1 Feeder No.2
Ursd< Ursd<
M
Preferred
feeder
Standby

feeder
Phase
angle
relay
φ U
ΔI>>
I >
ΔI>>
I >
ΔI>>
Capacitor bank
Metering
Id>
P2
P1
11 kV
From incoming
transformer
PFC/V
Controller
Lockout
I>> I >>
I> I >
Industrial System Protection – D04-003
29
Figure 14. Split phase capacitor bank differential protection
EXAMPLES
In next paragraphs, examples of the industrial system protection are presented.
FUSE CO-ORDINATION
Fuse application example is based on the configuration presented in Figure 15(a). This presents an unsatisfactory configuration with frequently encountered shortcomings. It can be noted that fuses B, C and D will discriminate with fuse A but the 400A sub-circuit fuse E may not discriminate, with the 500A sub-circuit fuse D at higher levels of fault current.
A
I >
B
I >
C
I >
Alarm Trip
Alarm Trip
C
I >
B
I >
A
I >
Industrial System Protection – D04-003
30
Contents
•
(Top)

- **History**
- **Features**
- **Examples**
- **Awards**
- **License**
- **Related projects**
-
  - **Design By Numbers**
  - **p5.js**
  - **ml5.js**
  - **Processing.js**
  - **P5Py**
  - **Processing.py**
  - **py5**
  - **Wiring, Arduino, and Fritzing**
  - **Mobile Processing**
  - **iProcessing**
  - **Spde**
  - **JRubyArt**

o
Quil

•
Media

•
See also

•
Footnotes

•
References

•
External links

Processing

- **Article**
- **Talk**
- **Read**
- **Edit**
- **View history**

Tools
•
•
•
•
•
•
•
•
•
•
•
•
•
•
Appearance
Text
•
Small
Standard
Large
Width
•
Standard
Wide
Color (beta)
•
Automatic

Light
Dark
From Wikipedia, the free encyclopedia
This article is about the graphics library. For the general term about a sequence of activities, see Process.
Processing

Paradigm
Object-oriented

Designed by
Casey Reas, Ben Fry

First appeared 2001; 23 years ago

Stable release
4.3 / July 26, 2023; 15 months ago[1]

License
GPL, LGPL

Filename extensions
.pde
Website processing.org

Processing is a free graphics library and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the purpose of teaching non-programmers the fundamentals of computer programming in a visual context.
Processing uses the Java language, with additional simplifications such as additional classes and aliased mathematical functions and operations. It also provides a graphical user interface for simplifying the compilation and execution stage.
The Processing language and IDE have been the precursor to other projects including Arduino and Wiring.
History
The project was initiated in 2001 by Casey Reas and Ben Fry, both formerly of the Aesthetics and Computation Group at the MIT Media Lab. In 2012, they started the Processing Foundation along with Daniel Shiffman, who joined as a third project lead. Johanna Hedva joined the Foundation in 2014 as Director of Advocacy.[2]
Originally, Processing had used the domain proce55ing.net, because the processing domain was taken; Reas and Fry eventually acquired the domain processing.org and moved the project to it in 2004.[3] While the original name had a combination of letters and numbers, it was always officially referred to as processing, but the abbreviated term p5 is still occasionally used (e.g. in "p5.js") in reference to the old domain name.[4]
In 2012 the Processing Foundation was established and received 501(c)(3) nonprofit status,[5] supporting the community around the tools and ideas that started with the Processing Project. The foundation encourages people around the world to meet annually in local events called Processing Community Day.[6]
Features
Processing IDE

Screenshot of Processing's integrated development environment.

Stable release
4.3 / July 26, 2023; 15 months ago
Repository
• github.com/processing/processing

Written in Java, GLSL, JavaScript

Operating system
Cross-platform

Type
Integrated development environment

Website processing.org

Processing includes a sketchbook, a minimal alternative to an integrated development environment (IDE) for organizing projects.[7]
Every Processing sketch is actually a subclass of the PApplet Java class (formerly a subclass of Java's built-in Applet) which implements most of the Processing language's features.[8]
When programming in Processing, all additional classes defined will be treated as inner classes when the code is translated into pure Java before compiling.[9] This means that the use of static variables and methods in classes is prohibited unless Processing is explicitly told to code in pure Java mode.
Processing also allows for users to create their own classes within the PApplet sketch. This allows for complex data types that can include any number of arguments and avoids the limitations of solely using standard data types such as: int (integer), char (character), float (real number), and color (RGB, RGBA, hex).
Examples
The simplest possible version of a "Hello World" program in Processing is:

```
// This prints "Hello World." to the IDE console.
println("Hello World.");
```

However, due to the more visually-oriented nature of Processing, the following code[10] is a better example of the look and feel of the language.

```
// Hello mouse.
void setup() {
size(400, 400);
stroke(255);
background(192, 64, 0);
}

void draw() {
line(150, 25, mouseX, mouseY);
}
```

Awards
In 2005 Reas and Fry won the Golden Nica award from Ars Electronica in its Net Vision category for their work on Processing.[11]
Ben Fry won the 2011 National Design Award given by the Smithsonian Cooper-Hewitt National Design Museum in the category of Interaction Design. The award statement says:
"Drawing on a background in graphic design and computer science, Ben Fry pursues a long-held fascination with visualizing data. As Principal of Fathom Information Design in Boston, Fry develops software, printed works, installations, and books that depict and explain topics from the human genome to baseball salaries to the evolution of text documents. With Casey Reas, he founded the Processing Project, an open-source programming environment for teaching computational design and sketching interactive-media software. It provides artists and designers

with accessible means of working with code while encouraging engineers and computer scientists to think about design concepts."[12]
License
Processing's core libraries, the code included in exported applications and applets, is licensed under the GNU Lesser General Public License, allowing users to release their original code with a choice of license.
The IDE is licensed under the GNU General Public License.
Related projects
Design By Numbers
Processing was based on the original work done on Design By Numbers project at MIT. It shares many of the same ideas and is a direct child of that experiment.
p5.js
In 2013, Lauren McCarthy created p5.js, a native JavaScript alternative to Processing.js that has the official support of the Processing Foundation. p5.js gained over 1.5 million users.[13]
Since April 2022, p5.js has been led by Qianqian Ye, an Adjunct Associate Professor of Media Arts at USC.[14]
ml5.js
ml5.js is a p5.js library developed by NYU's ITP/IMA with funding and support provided by a Google Education grant.
Daniel Shiffman has made videos demonstrating ml5 and is a notable code contributor.
Processing.js
Processing.js is a discontinued JavaScript port that enabled existing Processing Java code to run on web.
It was initially released in 2008 by John Resig. The project was later run through a partnership between the Mozilla Foundation and Seneca College, led by David Humphrey, Al MacDonald, and Corban Brook. Processing.js was kept at parity with Processing up to its API version 2.1 release. The project was discontinued in December 2018, two years after its active development had stopped.
P5Py
p5 is a Python library that provides high level drawing functionality to help you quickly create simulations and interactive art using Python. It combines the core ideas of Processing — learning to code in a visual context — with Python's readability to make programming more accessible to beginners, educators, and artists.[15]
Processing.py
Python Mode for Processing, or Processing.py is a Python interface to the underlying Java toolkit. It was chiefly developed by Jonathan Feinberg starting in 2010, with contributions from James Gilles and Ben Alkov.[16]
py5
py5 is a version of Processing for Python 3.8+. It makes the Java Processing jars available to the CPython interpreter using JPype. It can do just about everything Processing can do, except with Python instead of Java code.[17]
Wiring, Arduino, and Fritzing
Processing has spawned another project, Wiring, which uses the Processing IDE with a collection of libraries written in the C++ language as a way to teach artists how to program microcontrollers.[18] There are now two separate hardware projects, Wiring and Arduino, using the Wiring environment and language. Fritzing is another software environment of the same sort, which helps designers and artists to document their interactive prototypes and to take the step from physical prototyping to actual product.
Mobile Processing
Another spin-off project, now defunct, is Mobile Processing by Francis Li, which allowed software written using the Processing language and environment to run on Java powered mobile devices. Today some of the same functionality is provided by Processing itself.[19]
iProcessing
iProcessing was built to help people develop native iPhone applications using the Processing

language. It is an integration of the Processing.js library and a Javascript application framework for iPhone.[20]

Spde

Spde (Scala Processing Development Environment) replaces Processing's reduced Java syntax and custom preprocessor with the off-the-shelf Scala programming language which also runs on the Java platform and enforces some of the same restrictions such as disallowing static methods, while also allowing more concise code, and supporting functional programming.[21][22][23]

JRubyArt

JRubyArt (formerly named ruby-processing) is a wrapper for Processing in the Ruby language, that runs on the Java platform using JRuby.

Quil

Quil is an interactive animation library for Clojure and ClojureScript based on Processing.[24][25]

Media

The music video for "House of Cards" by Radiohead was created using Processing combined with data from lidar technology, along with using acrylic glass and mirrors to create scenes in which the image appears distorted, partially disappears, or disintegrate as if being carried by wind.[26]
Processing has also been used to create illustrations for publications such as Nature and The New York Times, to output sculptures for gallery exhibitions, to control huge video walls and to knit sweaters.[27]

See also

- Donate
- Create account
- Log in

•
Contents
•
(Top)

•
See also

•
References

Microelectronics

• Article
• Talk
• Read
• Edit
• View history
Tools
•
•
•
•
•
•
•
•
•
•
•
•
•
•
•
Appearance
Text
•
Small
Standard
Large
Width
•
Standard
Wide
Color (beta)
•
Automatic
Light
Dark
From Wikipedia, the free encyclopedia
For the American company, see Micro Electronics, Inc.
"Microminiaturisation" redirects here. For the art form, see Micro miniature.
An Integrated circuit (IC) as an example application in the field of microelectronics. The chip housing is opened to allow a view of the actual circuit. The golden connecting cables, which form

the electrical wiring between the IC and the housing contacts, can be seen on the sides.
Microelectronics is a subfield of electronics. As the name suggests, microelectronics relates to the study and manufacture (or microfabrication) of very small electronic designs and components. Usually, but not always, this means micrometre-scale or smaller. These devices are typically made from semiconductor materials. Many components of a normal electronic design are available in a microelectronic equivalent. These include transistors, capacitors, inductors, resistors, diodes and (naturally) insulators and conductors can all be found in microelectronic devices. Unique wiring techniques such as wire bonding are also often used in microelectronics because of the unusually small size of the components, leads and pads. This technique requires specialized equipment and is expensive.
Digital integrated circuits (ICs) consist of billions of transistors, resistors, diodes, and capacitors. [1] Analog circuits commonly contain resistors and capacitors as well. Inductors are used in some high frequency analog circuits, but tend to occupy larger chip area due to their lower reactance at low frequencies. Gyrators can replace them in many applications.
As techniques have improved, the scale of microelectronic components has continued to decrease. [2] At smaller scales, the relative impact of intrinsic circuit properties such as interconnections may become more significant. These are called parasitic effects, and the goal of the microelectronics design engineer is to find ways to compensate for or to minimize these effects, while delivering smaller, faster, and cheaper devices.
Today, microelectronics design is largely aided by Electronic Design Automation software.

See also
• Digital electronics
• Electrical engineering
• Kelvin probe force microscope
• Macroelectronics
• Microscale chemistry
• Nanoelectronics

References
• Veendrick, H.J.M. (2019). Bits on Chips. Springer. p. 284. ISBN 978-3-319-76095-7. https://link.springer.com/book/10.1007/978-3-319-76096-4/
1.
• Shamieh, Cathleen (2015-07-27). Electronics for dummies (3rd ed.). Hoboken, NJ. ISBN 9781119117971. OCLC 919482442.
2. • "The State of the Transistor in 3 Charts - IEEE Spectrum". IEEE. Retrieved 2024-02-20.

• v
• t
• e

Microtechnology

• Microelectromechanical systems
• Microtechnology
• Micromachinery

Basic structures • Interdigital transducer
• Cantilever
• Microchannel

Applications Sensors
• Microbolometer

Actuators
• Comb drive
• Scratch drive actuator

• Thermal actuator

Switches
• Digital micromirror device
• Optical switch

Other • Millipede memory
• Radio-frequency microelectromechanical systems
• Microoptoelectromechanical systems
• Microphotonics
• Biological microelectromechanical systems
• Microfluidics
• Micropower

Processes General • Surface micromachining
• Bulk micromachining
• HAR micromachining
• Deposition
• Lithography
• Etching
• Wire bonding
• 3D microfabrication

Specific • LOCOS
• Shallow trench isolation
• LIGA
• Lift-off
• Photolithography
• Silicon on insulator
• Smart cut

Wikibooks has a book on the topic of: Microtechnology
Authority control databases: National
• Germany
• United States
• France
• BnF data
• Japan
• Czech Republic
• Israel

Category:
• Electronics
• This page was last edited on 7 November 2024, at 14:47 (UTC).

- Code of Conduct
- Developers
- Statistics
- Cookie statement
- Mobile view
-
-
- Image Reconstruction Techniques

Lifeng Yu, PhD Mayo Clinic, Rochester, MN
Shuai Leng, PhD Mayo Clinic, Rochester, MN
(Updated Sept. 2016)

Image reconstruction in CT is a mathematical process that generates tomographic images from X-ray projection data acquired at many different angles around the patient. Image reconstruction has fundamental impacts on image quality and therefore on radiation dose. For a given radiation dose it is desirable to reconstruct images with the lowest possible noise without sacrificing image accuracy and spatial resolution. Reconstructions that improve image quality can be translated into a reduction of radiation dose because images of the same quality can be reconstructed at lower dose.

Two major categories of reconstruction methods exist, analytical reconstruction and iterative reconstruction (IR). Let's focus on the analytical reconstruction methods at first. There are many types of analytical reconstruction methods. The most commonly used analytical reconstruction methods on commercial CT scanners are all in the form of filtered backprojection (FBP), which uses a 1D filter on the projection data before backprojecting (2D or 3D) the data onto the image space. The popularity of FBP-type of method is mainly because of its computational efficiency and numerical stability. Various FBP-type of analytical reconstruction methods were developed for different generations of CT data-acquisition geometries, from 2D parallel- and fan-beam CT in the 1970s and 1980s to helical and multi-slice CT with narrow detector coverage in late 1990s and early 2000s, and to multi-slice CT with a wide detector coverage (up to 320 detector rows and 16 cm width). 3D weighted FBP methods are generally adopted on scanners with more than 16 detector rows [1]. For a general introduction of the fundamental principles of CT image reconstruction, please refer to Chapter 3 in Kak and Slaney's book [2]. An introduction to reconstruction methods in helical and multi-slice CT can be found in Hsieh's

- book [3]. A review of analytical CT image reconstruction methods used on clinical CT scanners can be found in the article by Flohr, et al [4].

Users of clinical CT scanners usually have very limited control over the inner workings of the reconstruction method and are confined principally to adjusting various parameters that potentially affect image quality. The reconstruction kernel, also referred to as "filter" or "algorithm" by some CT vendors, is one of the most

important parameters that affect the image quality. Generally speaking, there is a tradeoff between spatial resolution and noise for each kernel. A smoother kernel generates images with lower noise but with reduced spatial resolution. A sharper kernel generates images with higher spatial resolution, but increases the image noise.

The selection of reconstruction kernel should be based on specific clinical applications. For example, smooth kernels are usually used in brain exams or liver tumor assessment to reduce image noise and enhance low contrast detectability, whereas sharper kernels are usually used in exams to assess bony structures due to the clinical requirement of better spatial resolution.

Another important reconstruction parameter is slice thickness, which controls the spatial resolution in the longitudinal direction, influencing the tradeoffs among resolution, noise, and radiation dose. It is the responsibility of CT users to select the most appropriate reconstruction kernel and slice thickness for each clinical application so that the radiation dose can be minimized consistent with the image quality needed for the examination.

In addition to the conventional reconstruction kernels applied during image reconstruction, many noise reduction techniques, operating on image or projection data, are also available on commercial scanners or as third-party products. Many of these methods involve non-linear de-noising filters, some of which have been combined into the reconstruction kernels for the users' convenience. In some applications these methods

• perform quite well to reduce image noise while maintaining high-contrast resolution. If applied too aggressively, however, they tend to change the noise texture and sacrifice the low-contrast detectability in the image.

Therefore, careful evaluation of these filters should be performed for each diagnostic task before they are deployed into wide-scale clinical usage.

Scanning techniques and image reconstructions in ECG-gated cardiac CT have a unique impact on image quality and radiation dose. Half-scan (or short-scan) reconstruction is typically used to obtain better temporal resolution. For the widely employed retrospective ECG-gated helical scan mode, the helical pitch is very low (~0.2 to 0.3) in order to avoid anatomical discontinuities between contiguous heart cycles. A significant dose reduction technique in helical cardiac scanning is ECG tube-current pulsing, which involves modulating the tube current down to 4% to 20% of the full tube current for phases that are of minimal interest. Prospective ECG-triggered sequential (or step-and-shoot) scans are a more dose-efficient scanning mode for cardiac CT, especially for single-phase studies. An overview of scanning and reconstruction techniques in cardiac CT can be found in an

article by Flohr et al [5].
Different from analytical reconstruction methods, IR reconstructs images by iteratively optimizing an objective function, which typically consists of a data fidelity term and an edge-preserving regularization term [6]. The optimization process in IR involves iterations of forward projection and backprojection between image space and projection space. With the advances in computing technology, IR has become a very popular choice in routine CT practice because it has many advantages compared with conventional FBP techniques. Important physical factors including focal spot and detector geometry, photon statistics, X-ray beam spectrum, and scattering can be more accurately incorporated into IR, yielding lower image noise and higher spatial resolution compared with FBP. In addition, IR can reduce image artifacts such as beam hardening, windmill, and metal artifacts.
• Due to the intrinsic difference in data handling between FBP and iterative reconstruction, images from IR may have a different appearance (e.g., noise texture) from those using FBP reconstruction. More importantly, the spatial resolution in a local region of IR-reconstructed images is highly dependent on the contrast and noise of the surrounding structures due to the non-linear regularization term and other factors during the optimization process [7]. Measurements on different commercial IR methods have demonstrated this contrast- and noise-dependency of spatial resolution [8,9]. Because of this dependency, the amount of potential radiation dose reduction allowable by IR is dependent on the diagnostic task since the contrast of the subject and the noise of the exam vary substantially in clinical exams [10]. For low-contrast detection tasks, several phantom and human observer studies on multiple commercial IR methods demonstrated that only marginal or a small amount of radiation dose reduction can be allowed [11,12,13]. Careful clinical evaluation and reconstruction parameter optimization are required before IR can be used in routine practice [10,14,15]. Task-based image quality evaluation using model observers have been actively investigated so that image quality and dose reduction can be quantified objectively in an efficient manner [16,17,18].
• REFERENCES
1. Stierstorfer K, Rauscher A, Boese J, Bruder H, Schaller S, Flohr T. Weighted FBP - a simple approximate 3D FBP algorithm for multislice spiral CT with good dose usage for arbitrary pitch. Phys Med Biol 2004; 49:2209-18
2. Kak AC, Slaney M. Principles of Computed Tomographic Imaging. SIAM, Philadelphia, PA, 1987
3. Hsieh J. Computed Tomography: Principles, Design, Artifacts, and Recent Advances, 2nd ed: SPIE Press, Bellingham, Washington, 2009
4. Flohr TG, Schaller S, Stierstorfer K, Bruder H, Ohnesorge BM, Schoepf UJ. Multi-detector row CT

systems
and image-reconstruction techniques. Radiology 2005; 235(3): 756-73
5. Flohr TG, Schoepf UJ, Ohnesorge BM. Chasing the heart: new developments for cardiac CT. J Thorac
Imaging 2007; 22(1):4-16
6. Thibault JB, Sauer KD, Bouman CA, Hsieh J. A three-dimensional statistical approach to improved image
quality for multislice helical CT. Med Phys 2007; 34(11):4526-44
7. Fessler JA, Rogers WL. Spatial resolution properties of penalized-likelihood image reconstruction: Space-
invariant tomographs. IEEE Trans Image Process 1996; 5(9): 1346-1358
8. Li K, Garrett J, Ge Y, Chen GH. Statistical model based iterative reconstruction (MBIR) in clinical CT systems.
Part II. Experimental assessment of spatial resolution performance. Med Phys 2014; 41(7):071911
9. Yu L, Vrieze TJ, Leng S, Fletcher JG, McCollough CH. Technical Note: Measuring contrast- and noise-
dependent spatial resolution of an iterative reconstruction method in CT using ensemble averaging. Med
Phys 2015; 42(5):2261-67
10. Ehman EC, Yu L, Manduca A, et al. Methods for Clinical Evaluation of Noise Reduction Techniques in
Abdominopelvic CT. Radiographics 2014; 34(4):849-62
11. Schindera ST, Odedra D, Raza SA, et al. Iterative reconstruction algorithm for CT: can radiation dose be
decreased while low-contrast detectability is preserved? Radiology 2013(2); 269:511-18
12. Goenka AH, Herts BR, Obuchowski NA, et al. Effect of reduced radiation exposure and iterative reconstruction on detection of low-contrast low-attenuation lesions in an anthropomorphic liver phantom:
an 18-reader study. Radiology 2014; 272(1):154-63
13. McCollough CH, Yu L, Kofler JM, et al. Degradation of CT Low-Contrast Spatial Resolution Due to the Use of
Iterative Reconstruction and Reduced Dose Levels. Radiology 2015; 276(2):499-506
14. Winklehner A, Karlo C, Puippe G, et al. Raw data-based iterative reconstruction in body CTA: evaluation of
radiation dose saving potential. Eur Radiol 2011; 21(12):2521-26
15. Fletcher JG, Yu L, Li Z, et al. Observer Performance in the Detection and Classification of Malignant Hepatic
Nodules and Masses with CT Image-Space Denoising and Iterative Reconstruction. Radiology 2015; 276(2):465-78
• 16. Yu L, Leng S, Chen L, Kofler JM, Carter RE, McCollough CH. Prediction of human observer performance in a
2-alternative forced choice low-contrast detection task using channelized Hotelling observer: Impact of
radiation dose and reconstruction algorithms. Med Phys 2013; 40(4):041908-041901-041909
17. Leng S, Yu LF, Zhang Y, Carter R, Toledano AY, McCollough CH. Correlation between model observer and
human observer performance in CT imaging when lesion location is uncertain. Med Phys 2013; 40(8):
081908
18. Samei E, Richard S. Assessment of the dose reduction potential of a model-based iterative reconstruction
algorithm using a task-based performance metrology. Med Phys 2015; 42(1):314-23
Process Integration for VLSI•
(Top)

•
History

•
o
Background

o
VLSI

•
Structured design

•
Difficulties

•
See also

•
References

•
Further reading

•
External links

Very-large-scale integration

- **Article**
- **Talk**
- **Read**
- **Edit**
- **View history**

Tools
•
•
•
•
•
•
•
•
•
•
•
•
•
Appearance
Text
•
Small

Standard
Large
Width
•
Standard
Wide
Color (beta)
•
Automatic
Light
Dark
From Wikipedia, the free encyclopedia
"VLSI" redirects here. For the former company, see VLSI Technology.
Not to be confused with Very High Speed Integrated Circuit.

This article needs additional citations for verification. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.
Find sources: "Very-large-scale integration" – news • newspapers • books • scholar • JSTOR (September 2010) (Learn how and when to remove this message)

Very-large-scale integration (VLSI) is the process of creating an integrated circuit (IC) by combining millions or billions of MOS transistors onto a single chip. VLSI began in the 1970s when MOS integrated circuit (metal oxide semiconductor) chips were developed and then widely adopted, enabling complex semiconductor and telecommunications technologies. The microprocessor and memory chips are VLSI devices.
Before the introduction of VLSI technology, most ICs had a limited set of functions they could perform. An electronic circuit might consist of a CPU, ROM, RAM and other glue logic. VLSI enables IC designers to add all of these into one chip.
A VLSI integrated-circuit die
History
Background
The history of the transistor dates to the 1920s when several inventors attempted devices that were intended to control current in solid-state diodes and convert them into triodes. Success came after World War II, when the use of silicon and germanium crystals as radar detectors led to improvements in fabrication and theory. Scientists who had worked on radar returned to solid-state device development. With the invention of the first transistor at Bell Labs in 1947, the field of electronics shifted from vacuum tubes to solid-state devices.[1]
With the small transistor at their hands, electrical engineers of the 1950s saw the possibilities of constructing far more advanced circuits. However, as the complexity of circuits grew, problems arose.[2] One problem was the size of the circuit. A complex circuit like a computer was dependent on speed. If the components were large, the wires interconnecting them must be long. The electric signals took time to go through the circuit, thus slowing the computer.[2]
The invention of the integrated circuit by Jack Kilby and Robert Noyce solved this problem by making all the components and the chip out of the same block (monolith) of semiconductor material.[3] The circuits could be made smaller, and the manufacturing process could be automated. This led to the idea of integrating all components on a single-crystal silicon wafer, which led to small-scale integration (SSI) in the early 1960s, and then medium-scale integration (MSI) in the late 1960s.[4]
VLSI
See also: MOS integrated circuit
General Microelectronics introduced the first commercial MOS integrated circuit in 1964.[5] In the early 1970s, MOS integrated circuit technology allowed the integration of more than 10,000 transistors in a single chip.[6] This paved the way for VLSI in the 1970s and 1980s, with tens of thousands of MOS transistors on a single chip (later hundreds of thousands, then millions, and now

billions).
The first semiconductor chips held two transistors each. Subsequent advances added more transistors, and as a consequence, more individual functions or systems were integrated over time. The first integrated circuits held only a few devices, perhaps as many as ten diodes, transistors, resistors and capacitors, making it possible to fabricate one or more logic gates on a single device. Now known retrospectively as small-scale integration (SSI), improvements in technique led to devices with hundreds of logic gates, known as medium-scale integration (MSI). Further improvements led to large-scale integration (LSI), i.e. systems with at least a thousand logic gates. Current technology has moved far past this mark and today's microprocessors have many millions of gates and billions of individual transistors.
At one time, there was an effort to name and calibrate various levels of large-scale integration above VLSI. Terms like ultra-large-scale integration (ULSI) were used. But the huge number of gates and transistors available on common devices has rendered such fine distinctions moot. Terms suggesting greater than VLSI levels of integration are no longer in widespread use.
In 2008, billion-transistor processors became commercially available. This became more commonplace as semiconductor fabrication advanced from the then-current generation of 65 nm processors. Current designs, unlike the earliest devices, use extensive design automation and automated logic synthesis to lay out the transistors, enabling higher levels of complexity in the resulting logic functionality. Certa

- Donate
- Create account
- Log in

Contents

(Top)

Background

Disadvantages

Granularity

Hardware

• •
Software

•
•
Algorithmic methods

•
Fault tolerance

•
History

•
Biological brain as massively parallel computer

•
See also

•
References

•
Further reading

•
External links

Parallel computing

- **Article**
- **Talk**
- **Read**
- **Edit**
- **View history**

Tools
•
•
•
•
•
•
•
•
•
•
•
•
•
•
Appearance
Text
•
Small

Standard
Large
Width
•
Standard
Wide
Color (beta)
•
Automatic
Light
Dark

From Wikipedia, the free encyclopedia

"Parallelization" redirects here. For parallelization of manifolds, see Parallelization (mathematics).

Large supercomputers such as IBM's Blue Gene/P are designed to heavily exploit parallelism.

Parallel computing is a type of computation in which many calculations or processes are carried out simultaneously.[1] Large problems can often be divided into smaller ones, which can then be solved at the same time. There are several different forms of parallel computing: bit-level, instruction-level, data, and task parallelism. Parallelism has long been employed in high-performance computing, but has gained broader interest due to the physical constraints preventing frequency scaling.[2] As power consumption (and consequently heat generation) by computers has become a concern in recent years,[3] parallel computing has become the dominant paradigm in computer architecture, mainly in the form of multi-core processors.[4]

Parallelism vs concurrency

In computer science, parallelism and concurrency are two different things: a parallel program uses multiple CPU cores, each core performing a task independently. On the other hand, concurrency enables a program to deal with multiple tasks even on a single CPU core; the core switches between tasks (i.e. threads) without necessarily completing each one. A program can have both, neither or a combination of parallelism and concurrency characteristics.[5]

Parallel computers can be roughly classified according to the level at which the hardware supports parallelism, with multi-core and multi-processor computers having multiple processing elements within a single machine, while clusters, MPPs, and grids use multiple computers to work on the same task. Specialized parallel computer architectures are sometimes used alongside traditional processors, for accelerating specific tasks.

In some cases parallelism is transparent to the programmer, such as in bit-level or instruction-level parallelism, but explicitly parallel algorithms, particularly those that use concurrency, are more difficult to write than sequential ones,[6] because concurrency introduces several new classes of potential software bugs, of which race conditions are the most common. Communication and synchronization between the different subtasks are typically some of the greatest obstacles to getting optimal parallel program performance.

A theoretical upper bound on the speed-up of a single program as a result of parallelization is given by Amdahl's law, which states that it is limited by the fraction of time for which the parallelization can be utilised.

Background

Traditionally, computer software has been written for serial computation. To solve a problem, an algorithm is constructed and implemented as a serial stream of instructions. These instructions are executed on a central processing unit on one computer. Only one instruction may execute at a time —after that instruction is finished, the next one is executed.[7]

Parallel computing, on the other hand, uses multiple processing elements simultaneously to solve a problem. This is accomplished by breaking the problem into independent parts so that each processing element can execute its part of the algorithm simultaneously with the others. The processing elements can be diverse and include resources such as a single computer with multiple processors, several networked computers, specialized hardware, or any combination of the above.

[7] Historically parallel computing was used for scientific computing and the simulation of scientific problems, particularly in the natural and engineering sciences, such as meteorology. This led to the design of parallel hardware and software, as well as high performance computing.[8]

Frequency scaling was the dominant reason for improvements in computer performance from the mid-1980s until 2004. The runtime of a program is equal to the number of instructions multiplied by the average time per instruction. Maintaining everything else constant, increasing the clock frequency decreases the average time it takes to execute an instruction. An increase in frequency thus decreases runtime for all compute-bound programs.[9] However, power consumption P by a chip is given by the equation P = C × V 2 × F, where C is the capacitance being switched per clock cycle (proportional to the number of transistors whose inputs change), V is voltage, and F is the processor frequency (cycles per second).[10] Increases in frequency increase the amount of power used in a processor. Increasing processor power consumption led ultimately to Intel's May 8, 2004 cancellation of its Tejas and Jayhawk processors, which is generally cited as the end of frequency scaling as the dominant computer architecture paradigm.[11]

To deal with the problem of power consumption and overheating the major central processing unit (CPU or processor) manufacturers started to produce power efficient processors with multiple cores. The core is the computing unit of the processor and in multi-core processors each core is independent and can access the same memory concurrently. Multi-core processors have brought parallel computing to desktop computers. Thus parallelization of serial programs has become a mainstream programming task. In 2012 quad-core processors became standard for desktop computers, while servers have 10+ core processors. From Moore's law it can be predicted that the number of cores per processor will double every 18–24 months. This could mean that after 2020 a typical processor will have dozens or hundreds of cores, however in reality the standard is somewhere in the region of 4 to 16 cores, with some designs having a mix of performance and efficiency cores (such as ARM's big.LITTLE design) due to thermal and design constraints.[12] [citation needed]

An operating system can ensure that different tasks and user programs are run in parallel on the available cores. However, for a serial software program to take full advantage of the multi-core architecture the programmer needs to restructure and parallelize the code. A speed-up of application software runtime will no longer be achieved through frequency scaling, instead programmers will need to parallelize their software code to take advantage of the increasing computing power of multicore architectures.[13]

Relevant laws

A graphical representation of Amdahl's law. The law demonstrates the theoretical maximum speedup of an overall system and the concept of diminishing returns. If exactly 50% of the work can be parallelized, the best possible speedup is 2 times. If 95% of the work can be parallelized, the best possible speedup is 20 times. According to the law, even with an infinite number of processors, the speedup is constrained by the unparallelizable portion. Assume that a task has two independent parts, A and B. Part B takes roughly 25% of the time of the whole computation. By working very hard, one may be able to make this part 5 times faster, but this only reduces the time for the whole computation by a little. In contrast, one may need to perform less work to make part A twice as fast. This will make the computation much faster than by optimizing part B, even though part B's speedup is greater by ratio, (5 times versus 2 times).

Main article: Amdahl's law

Optimally, the speedup from parallelization would be linear—doubling the number of processing elements should halve the runtime, and doubling it a second time should again halve the runtime. However, very few parallel algorithms achieve optimal speedup. Most of them have a near-linear speedup for small numbers of processing elements, which flattens out into a constant value for large numbers of processing elements.

The maximum potential speedup of an overall system can be calculated by Amdahl's law. [14] Amdahl's Law indicates that optimal performance improvement is achieved by balancing enhancements to both parallelizable and non-parallelizable components of a task. Furthermore, it reveals that increasing the number of processors yields diminishing returns, with negligible speedup gains beyond a certain point. [15][16]

Amdahl's Law has limitations, including assumptions of fixed workload, neglecting inter-process communication and synchronization overheads, primarily focusing on computational aspect and ignoring extrinsic factors such as data persistence, I/O operations, and memory access overheads. [17][18][19]

Gustafson's law and Universal Scalability Law give a more realistic assessment of the parallel performance. [20][21]

A graphical representation of Gustafson's law

Dependencies

Understanding data dependencies is fundamental in implementing parallel algorithms. No program can run more quickly than the longest chain of dependent calculations (known as the critical path), since calculations that depend upon prior calculations in the chain must be executed in order. However, most algorithms do not consist of just a long chain of dependent calculations; there are usually opportunities to execute independent calculations in parallel.

Let Pi and Pj be two program segments. Bernstein's conditions[22] describe when the two are independent and can be executed in parallel. For Pi, let Ii be all of the input variables and Oi the output variables, and likewise for Pj. Pi and Pj are independent if they satisfy

Violation of the first condition introduces a flow dependency, corresponding to the first segment producing a result used by the second segment. The second condition represents an anti-dependency, when the second segment produces a variable needed by the first segment. The third and final condition represents an output dependency: when two segments write to the same location, the result comes from the logically last executed segment.[23]

Consider the following functions, which demonstrate several kinds of dependencies:

1: function Dep(a, b)
2: c := a * b
3: d := 3 * c
4: end function

In this example, instruction 3 cannot be executed before (or even in parallel with) instruction 2, because instruction 3 uses a result from instruction 2. It violates condition 1, and thus introduces a flow dependency.

1: function NoDep(a, b)
2: c := a * b
3: d := 3 * b
4: e := a + b
5: end function

In this example, there are no dependencies between the instructions, so they can all be run in parallel.

Bernstein's conditions do not allow memory to be shared between different processes. For that, some means of enforcing an ordering between accesses is necessary, such as semaphores, barriers or some other synchronization method.

Race conditions, mutual exclusion, synchronization, and parallel slowdown

Subtasks in a parallel program are often called threads. Some parallel computer architectures use smaller, lightweight versions of threads known as fibers, while others use bigger versions known as processes. However, "threads" is generally accepted as a generic term for subtasks.[24]

Threads will often need synchronized access to an object or other resource, for example when they must update a variable that is shared between them. Without synchronization, the instructions between the two threads may be interleaved in any order. For example, consider the following program:

Thread A Thread B
1A: Read variable V 1B: Read variable V
2A: Add 1 to variable V 2B: Add 1 to variable V
3A: Write back to variable V 3B: Write back to variable V

If instruction 1B is executed between 1A and 3A, or if instruction 1A is executed between 1B and 3B, the program will produce incorrect data. This is known as a race condition. The programmer must use a lock to provide mutual exclusion. A lock is a programming language construct that allows one thread to take control of a variable and prevent other threads from reading or writing it, until that variable is unlocked. The thread holding the lock is free to execute its critical section (the section of a program that requires exclusive access to some variable), and to unlock the data when it is finished. Therefore, to guarantee correct program execution, the above program can be rewritten to use locks:
Thread A Thread B
1A: Lock variable V 1B: Lock variable V
2A: Read variable V 2B: Read variable V
3A: Add 1 to variable V 3B: Add 1 to variable V
4A: Write back to variable V 4B: Write back to variable V
5A: Unlock variable V 5B: Unlock variable V
One thread will successfully lock variable V, while the other thread will be locked out—unable to proceed until V is unlocked again. This guarantees correct execution of the program. Locks may be necessary to ensure correct program execution when threads must serialize access to resources, but their use can greatly slow a program and may affect its reliability.[25]
Locking multiple variables using non-atomic locks introduces the possibility of program deadlock. An atomic lock locks multiple variables all at once. If it cannot lock all of them, it does not lock any of them. If two threads each need to lock the same two variables using non-atomic locks, it is possible that one thread will lock one of them and the second thread will lock the second variable. In such a case, neither thread can complete, and deadlock results.[26]
Many parallel programs require that their subtasks act in synchrony. This requires the use of a barrier. Barriers are typically implemented using a lock or a semaphore.[27] One class of algorithms, known as lock-free and wait-free algorithms, altogether avoids the use of locks and barriers. However, this approach is generally difficult to implement and requires correctly designed data structures.[28]
Not all parallelization results in speed-up. Generally, as a task is split up into more and more threads, those threads spend an ever-increasing portion of their time communicating with each other or waiting on each other for access to resources.[29][30] Once the overhead from resource contention or communication dominates the time spent on other computation, further parallelization (that is, splitting the workload over even more threads) increases rather than decreases the amount of time required to finish. This problem, known as parallel slowdown,[31] can be improved in some cases by software analysis and redesign.[32]
Fine-grained, coarse-grained, and embarrassing parallelism
Applications are often classified according to how often their subtasks need to synchronize or communicate with each other. An application exhibits fine-grained parallelism if its subtasks must communicate many times per second; it exhibits coarse-grained parallelism if they do not communicate many times per second, and it exhibits embarrassing parallelism if they rarely or never have to communicate. Embarrassingly parallel applications are considered the easiest to parallelize.
Flynn's taxonomy
Main article: Flynn's taxonomy
Michael J. Flynn created one of the earliest classification systems for parallel (and sequential) computers and programs, now known as Flynn's taxonomy. Flynn classified programs and computers by whether they were operating using a single set or multiple sets of instructions, and whether or not those instructions were using a single set or multiple sets of data.
Flynn's taxonomy

Single data stream
• SISD
• MISD

Multiple data streams
- **SIMD**
- **MIMD**

SIMD subcategories[33]

- **Array processing (SIMT)**
- **Pipelined processing (packed SIMD)**
- **Associative processing (predicated/masked SIMD)**

See also
- **SPMD**
- **MPMD**

The single-instruction-single-data (SISD) classification is equivalent to an entirely sequential program. The single-instruction-multiple-data (SIMD) classification is analogous to doing the same operation repeatedly over a large data set. This is commonly done in signal processing applications. Multiple-instruction-single-data (MISD) is a rarely used classification. While computer architectures to deal with this were devised (such as systolic arrays), few applications that fit this class materialized. Multiple-instruction-multiple-data (MIMD) programs are by far the most common type of parallel programs.
According to David A. Patterson and John L. Hennessy, "Some machines are hybrids of these categories, of course, but this classic model has survived because it is simple, easy to understand, and gives a good first approximation. It is also—perhaps because of its understandability—the most widely used scheme."[34]
Disadvantages
Parallel computing can incur significant overhead in practice, primarily due to the costs associated with merging data from multiple processes. Specifically, inter-process communication and synchronization can lead to overheads that are substantially higher—often by two or more orders of magnitude—compared to processing the same data on a single thread. [35][36][37] Therefore, the overall improvement should be carefully evaluated.
Granularity
Bit-level parallelism
Main article: Bit-level parallelism
Taiwania 3 of Taiwan, a parallel supercomputing device that joined COVID-19 research
From the advent of very-large-scale integration (VLSI) computer-chip fabrication technology in the 1970s until about 1986, speed-up in computer architecture was driven by doubling computer word size—the amount of information the processor can manipulate per cycle.[38] Increasing the word size reduces the number of instructions the processor must execute to perform an operation on variables whose sizes are greater than the length of the word. For example, where an 8-bit processor must add two 16-bit integers, the processor must first add the 8 lower-order bits from each integer using the standard addition instruction, then add the 8 higher-order bits using an add-with-carry instruction and the carry bit from the lower order addition; thus, an 8-bit processor requires two instructions to complete a single operation, where a 16-bit processor would be able to complete the operation with a single instruction.
Historically, 4-bit microprocessors were replaced with 8-bit, then 16-bit, then 32-bit microprocessors. This trend generally came to an end with the introduction of 32-bit processors, which has been a standard in general-purpose computing for two decades. Not until the early 2000s, with the advent of x86-64 architectures, did 64-bit processors become commonplace.
Instruction-level parallelism
Main article: Instruction-level parallelism
A canonical processor without pipeline. It takes five clock cycles to complete one instruction and thus the processor can issue subscalar performance (IPC = 0.2 < 1).
A computer program is, in essence, a stream of instructions executed by a processor. Without

instruction-level parallelism, a processor can only issue less than one instruction per clock cycle (IPC 1).

Most modern processors also have multiple execution units. They usually combine this feature with pipelining and thus can issue more than one instruction per clock cycle (IPC > 1). These processors are known as superscalar processors. Superscalar processors differ from multi-core processors in that the several execution units are not entire processors (i.e. processing units). Instructions can be grouped together only if there is no data dependency between them. Scoreboarding and the Tomasulo algorithm (which is similar to scoreboarding but makes use of register renaming) are two of the most common techniques for implementing out-of-order execution and instruction-level parallelism.

Task parallelism

Main article: Task parallelism

Task parallelisms is the characteristic of a parallel program that "entirely different calculations can be performed on either the same or different sets of data".[41] This contrasts with data parallelism, where the same calculation is performed on the same or different sets of data. Task parallelism involves the decomposition of a task into sub-tasks and then allocating each sub-task to a processor for execution. The processors would then execute these sub-tasks concurrently and often cooperatively. Task parallelism does not usually scale with the size of a problem.[42]

Superword level parallelism

Superword level parallelism is a vectorization technique based on loop unrolling and basic block vectorization. It is distinct from loop vectorization algorithms in that it can exploit parallelism of inline code, such as manipulating coordinates, color channels or in loops unrolled by hand.[43]

Hardware

Memory and communication

Main memory in a parallel computer is either shared memory (shared between all processing elements in a single address space), or distributed memory (in which each processing element has its own local address space).[44] Distributed memory refers to the fact that the memory is logically distributed, but often implies that it is physically distributed as well. Distributed shared memory and memory virtualization combine the two approaches, where the processing element has its own local memory and access to the memory on non-local processors. Accesses to local memory are typically faster than accesses to non-local memory. On the supercomputers, distributed shared memory space can be implemented using the programming model such as PGAS. This model allows processes on one compute node to transparently access the remote memory of another compute node. All compute nodes are also connected to an external shared memory system via high-speed interconnect, such as Infiniband, this external shared memory system is known as burst buffer, which is typically built from arrays of non-volatile memory physically distributed across multiple I/O nodes.

A logical view of a non-uniform memory access (NUMA) architecture. Processors in one directory can access that directory's memory with less latency than they can access memory in the other directory's memory.

Computer architectures in which each element of main memory can be accessed with equal latency and bandwidth are known as uniform memory access (UMA) systems. Typically, that can be achieved only by a shared memory system, in which the memory is not physically distributed. A system that does not have this property is known as a non-uniform memory access (NUMA) architecture. Distributed memory systems have non-uniform memory access.

Computer systems make use of caches—small and fast memories located close to the processor which store temporary copies of memory values (nearby in both the physical and logical sense). Parallel computer systems have difficulties with caches that may store the same value in more than one location, with the possibility of incorrect program execution. These computers require a cache coherency system, which keeps track of cached values and strategically purges them, thus ensuring correct program execution. Bus snooping is one of the most common methods for keeping track of which values are being accessed (and thus should be purged). Designing large, high-performance cache coherence systems is a very difficult problem in computer architecture. As a result, shared memory computer architectures do not scale as well as distributed memory

systems do.[44]
Processor–processor and processor–memory communication can be implemented in hardware in several ways, including via shared (either multiported or multiplexed) memory, a crossbar switch, a shared bus or an interconnect network of a myriad of topologies including star, ring, tree, hypercube, fat hypercube (a hypercube with more than one processor at a node), or n-dimensional mesh.
Parallel computers based on interconnected networks need to have some kind of routing to enable the passing of messages between nodes that are not directly connected. The medium used for communication between the processors is likely to be hierarchical in large multiprocessor machines.
Classes of parallel computers
Parallel computers can be roughly classified according to the level at which the hardware supports parallelism. This classification is broadly analogous to the distance between basic computing nodes. These are not mutually exclusive; for example, clusters of symmetric multiprocessors are relatively common.
Multi-core computing
Main article: Multi-core processor
A multi-core processor is a processor that includes multiple processing units (called "cores") on the same chip. This processor differs from a superscalar processor, which includes multiple execution units and can issue multiple instructions per clock cycle from one instruction stream (thread); in contrast, a multi-core processor can issue multiple instructions per clock cycle from multiple instruction streams. IBM's Cell microprocessor, designed for use in the Sony PlayStation 3, is a prominent multi-core processor. Each core in a multi-core processor can potentially be superscalar as well—that is, on every clock cycle, each core can issue multiple instructions from one thread.
Simultaneous multithreading (of which Intel's Hyper-Threading is the best known) was an early form of pseudo-multi-coreism. A processor capable of concurrent multithreading includes multiple execution units in the same processing unit—that is it has a superscalar architecture—and can issue multiple instructions per clock cycle from multiple threads. Temporal multithreading on the other hand includes a single execution unit in the same processing unit and can issue one instruction at a time from multiple threads.
Symmetric multiprocessing
Main article: Symmetric multiprocessing
A symmetric multiprocessor (SMP) is a computer system with multiple identical processors that share memory and connect via a bus.[45] Bus contention prevents bus architectures from scaling. As a result, SMPs generally do not comprise more than 32 processors.[46] Because of the small size of the processors and the significant reduction in the requirements for bus bandwidth achieved by large caches, such symmetric multiprocessors are extremely cost-effective, provided that a sufficient amount of memory bandwidth exists.[45]
Distributed computing
Main article: Distributed computing
A distributed computer (also known as a distributed memory multiprocessor) is a distributed memory computer system in which the processing elements are connected by a network. Distributed computers are highly scalable. The terms "concurrent computing", "parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them.[47] The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel.[48]
Cluster computing
Main article: Computer cluster
A Beowulf cluster
A cluster is a group of loosely coupled computers that work together closely, so that in some respects they can be regarded as a single computer.[49] Clusters are composed of multiple standalone machines connected by a network. While machines in a cluster do not have to be symmetric, load balancing is more difficult if they are not. The most common type of cluster is the

Beowulf cluster, which is a cluster implemented on multiple identical commercial off-the-shelf computers connected with a TCP/IP Ethernet local area network.[50] Beowulf technology was originally developed by Thomas Sterling and Donald Becker. 87% of all Top500 supercomputers are clusters.[51] The remaining are Massively Parallel Processors, explained below.
Because grid computing systems (described below) can easily handle embarrassingly parallel problems, modern clusters are typically designed to handle more difficult problems—problems that require nodes to share intermediate results with each other more often. This requires a high bandwidth and, more importantly, a low-latency interconnection network. Many historic and current supercomputers use customized high-performance network hardware specifically designed for cluster computing, such as the Cray Gemini network.[52] As of 2014, most current supercomputers use some off-the-shelf standard network hardware, often Myrinet, InfiniBand, or Gigabit Ethernet.
Massively parallel computing
Main article: Massively parallel (computing)
A cabinet from IBM's Blue Gene/L massively parallel supercomputer
A massively parallel processor (MPP) is a single computer with many networked processors. MPPs have many of the same characteristics as clusters, but MPPs have specialized interconnect networks (whereas clusters use commodity hardware for networking). MPPs also tend to be larger than clusters, typically having "far more" than 100 processors.[53] In an MPP, "each CPU contains its own memory and copy of the operating system and application. Each subsystem communicates with the others via a high-speed interconnect."[54]
IBM's Blue Gene/L, the fifth fastest supercomputer in the world according to the June 2009 TOP500 ranking, is an MPP.
Grid computing
Main article: Grid computing
Grid computing is the most distributed form of parallel computing. It makes use of computers communicating over the Internet to work on a given problem. Because of the low bandwidth and extremely high latency available on the Internet, distributed computing typically deals only with embarrassingly parallel problems.
Most grid computing applications use middleware (software that sits between the operating system and the application to manage network resources and standardize the software interface). The most common grid computing middleware is the Berkeley Open Infrastructure for Network Computing (BOINC). Often volunteer computing software makes use of "spare cycles", performing computations at times when a computer is idling.[55]
Cloud computing
Main article: Cloud computing
The ubiquity of Internet brought the possibility of large-scale cloud computing.
Specialized parallel computers
Within parallel computing, there are specialized parallel devices that remain niche areas of interest. While not domain-specific, they tend to be applicable to only a few classes of parallel problems.
Reconfigurable computing with field-programmable gate arrays
Reconfigurable computing is the use of a field-programmable gate array (FPGA) as a co-processor to a general-purpose computer. An FPGA is, in essence, a computer chip that can rewire itself for a given task.
FPGAs can be programmed with hardware description languages such as VHDL[56] or Verilog.[57] Several vendors have created C to HDL languages that attempt to emulate the syntax and semantics of the C programming language, with which most programmers are familiar. The best known C to HDL languages are Mitrion-C, Impulse C, and Handel-C. Specific subsets of SystemC based on C++ can also be used for this purpose.
AMD's decision to open its HyperTransport technology to third-party vendors has become the enabling technology for high-performance reconfigurable computing.[58] According to Michael R. D'Amour, Chief Operating Officer of DRC Computer Corporation, "when we first walked into AMD, they called us 'the socket stealers.' Now they call us their partners."[58]
General-purpose computing on graphics processing units (GPGPU)
Main article: GPGPU

Nvidia's Tesla GPGPU card

General-purpose computing on graphics processing units (GPGPU) is a fairly recent trend in computer engineering research. GPUs are co-processors that have been heavily optimized for computer graphics processing.[59] Computer graphics processing is a field dominated by data parallel operations—particularly linear algebra matrix operations.

In the early days, GPGPU programs used the normal graphics APIs for executing programs. However, several new programming languages and platforms have been built to do general purpose computation on GPUs with both Nvidia and AMD releasing programming environments with CUDA and Stream SDK respectively. Other GPU programming languages include BrookGPU, PeakStream, and RapidMind. Nvidia has also released specific products for computation in their Tesla series. The technology consortium Khronos Group has released the OpenCL specification, which is a framework for writing programs that execute across platforms consisting of CPUs and GPUs. AMD, Apple, Intel, Nvidia and others are supporting OpenCL.

Application-specific integrated circuits

Main article: Application-specific integrated circuit

Several application-specific integrated circuit (ASIC) approaches have been devised for dealing with parallel applications.[60][61][62]

Because an ASIC is (by definition) specific to a given application, it can be fully optimized for that application. As a result, for a given application, an ASIC tends to outperform a general-purpose computer. However, ASICs are created by UV photolithography. This process requires a mask set, which can be extremely expensive. A mask set can cost over a million US dollars.[63] (The smaller the transistors required for the chip, the more expensive the mask will be.) Meanwhile, performance increases in general-purpose computing over time (as described by Moore's law) tend to wipe out these gains in only one or two chip generations.[58] High initial cost, and the tendency to be overtaken by Moore's-law-driven general-purpose computing, has rendered ASICs unfeasible for most parallel computing applications. However, some have been built. One example is the PFLOPS RIKEN MDGRAPE-3 machine which uses custom ASICs for molecular dynamics simulation.

Vector processors

Main article: Vector processor

The Cray-1 is a vector processor.

A vector processor is a CPU or computer system that can execute the same instruction on large sets of data. Vector processors have high-level operations that work on linear arrays of numbers or vectors. An example vector operation is A = B × C, where A, B, and C are each 64-element vectors of 64-bit floating-point numbers.[64] They are closely related to Flynn's SIMD classification.[64]

Cray computers became famous for their vector-processing computers in the 1970s and 1980s. However, vector processors—both as CPUs and as full computer systems—have generally disappeared. Modern processor instruction sets do include some vector processing instructions, such as with Freescale Semiconductor's AltiVec and Intel's Streaming SIMD Extensions (SSE).

Software

Parallel programming languages

Main article: List of concurrent and parallel programming languages

Concurrent programming languages, libraries, APIs, and parallel programming models (such as algorithmic skeletons) have been created for programming parallel computers. These can generally be divided into classes based on the assumptions they make about the underlying memory architecture—shared memory, distributed memory, or shared distributed memory. Shared memory programming languages communicate by manipulating shared memory variables. Distributed memory uses message passing. POSIX Threads and OpenMP are two of the most widely used shared memory APIs, whereas Message Passing Interface (MPI) is the most widely used message-passing system API.[65] One concept used in programming parallel programs is the future concept, where one part of a program promises to deliver a required datum to another part of a program at some future time.

Efforts to standardize parallel programming include an open standard called OpenHMPP for hybrid multi-core parallel programming. The OpenHMPP directive-based programming model offers a syntax to efficiently offload computations on hardware accelerators and to optimize data movement

to/from the hardware memory using remote procedure calls.
The rise of consumer GPUs has led to support for compute kernels, either in graphics APIs (referred to as compute shaders), in dedicated APIs (such as OpenCL), or in other language extensions.

Automatic parallelization

Main article: Automatic parallelization

Automatic parallelization of a sequential program by a compiler is the "holy grail" of parallel computing, especially with the aforementioned limit of processor frequency. Despite decades of work by compiler researchers, automatic parallelization has had only limited success.[66]
Mainstream parallel programming languages remain either explicitly parallel or (at best) partially implicit, in which a programmer gives the compiler directives for parallelization. A few fully implicit parallel programming languages exist—SISAL, Parallel Haskell, SequenceL, System C (for FPGAs), Mitrion-C, VHDL, and Verilog.

Application checkpointing

Main article: Application checkpointing

As a computer system grows in complexity, the mean time between failures usually decreases. Application checkpointing is a technique whereby the computer system takes a "snapshot" of the application—a record of all current resource allocations and variable states, akin to a core dump—; this information can be used to restore the program if the computer should fail. Application checkpointing means that the program has to restart from only its last checkpoint rather than the beginning. While checkpointing provides benefits in a variety of situations, it is especially useful in highly parallel systems with a large number of processors used in high performance computing.[67]

Algorithmic methods

As parallel computers become larger and faster, we are now able to solve problems that had previously taken too long to run. Fields as varied as bioinformatics (for protein folding and sequence analysis) and economics have taken advantage of parallel computing. Common types of problems in parallel computing applications include:[68]

- Dense linear algebra
- Sparse linear algebra
- Spectral methods (such as Cooley–Tukey fast Fourier transform)
- N-body problems (such as Barnes–Hut simulation)
- Structured grid problems (such as Lattice Boltzmann methods)
- Unstructured grid problems (such as found in finite element analysis)
- Monte Carlo method
- Combinational logic (such as brute-force cryptographic techniques)
- Graph traversal (such as sorting algorithms)
- Dynamic programming
- Branch and bound methods
- Graphical models (such as detecting hidden Markov models and constructing Bayesian networks)
- HBJ model, a concise message-passing model[69]
- Finite-state machine simulation

Fault tolerance

Further information: Fault-tolerant computer system

Parallel computing can also be applied to the design of fault-tolerant computer systems, particularly via lockstep systems performing the same operation in parallel. This provides redundancy in case one component fails, and also allows automatic error detection and error correction if the results differ. These methods can be used to help prevent single-event upsets caused by transient errors.[70] Although additional measures may be required in embedded or specialized systems, this method can provide a cost-effective approach to achieve n-modular redundancy in commercial off-the-shelf systems.

History

For broader coverage of this topic, see History of computing.

ILLIAC IV, "the most infamous of supercomputers"[71]

The origins of true (MIMD) parallelism go back to Luigi Federico Menabrea and his Sketch of the Analytic Engine Invented by Charles Babbage.[72][73][74]

In 1957, Compagnie des Machines Bull announced the first computer architecture specifically designed for parallelism, the Gamma 60.[75] It utilized a fork-join model and a "Program Distributor" to dispatch and collect data to and from independent processing units connected to a central memory.[76][77]

In April 1958, Stanley Gill (Ferranti) discussed parallel programming and the need for branching and waiting.[78] Also in 1958, IBM researchers John Cocke and Daniel Slotnick discussed the use of parallelism in numerical calculations for the first time.[79] Burroughs Corporation introduced the D825 in 1962, a four-processor computer that accessed up to 16 memory modules through a crossbar switch.[80] In 1967, Amdahl and Slotnick published a debate about the feasibility of parallel processing at American Federation of Information Processing Societies Conference.[79] It was during this debate that Amdahl's law was coined to define the limit of speed-up due to parallelism.

In 1969, Honeywell introduced its first Multics system, a symmetric multiprocessor system capable of running up to eight processors in parallel.[79] C.mmp, a multi-processor project at Carnegie Mellon University in the 1970s, was among the first multiprocessors with more than a few processors. The first bus-connected multiprocessor with snooping caches was the Synapse N+1 in 1984.[73]

SIMD parallel computers can be traced back to the 1970s. The motivation behind early SIMD computers was to amortize the gate delay of the processor's control unit over multiple instructions.[81] In 1964, Slotnick had proposed building a massively parallel computer for the Lawrence Livermore National Laboratory.[79] His design was funded by the US Air Force, which was the earliest SIMD parallel-computing effort, ILLIAC IV.[79] The key to its design was a fairly high parallelism, with up to 256 processors, which allowed the machine to work on large datasets in what would later be known as vector processing. However, ILLIAC IV was called "the most infamous of supercomputers", because the project was only one-fourth completed, but took 11 years and cost almost four times the original estimate.[71] When it was finally ready to run its first real application in 1976, it was outperformed by existing commercial supercomputers such as the Cray-1.

Biological brain as massively parallel computer

In the early 1970s, at the MIT Computer Science and Artificial Intelligence Laboratory, Marvin Minsky and Seymour Papert started developing the Society of Mind theory, which views the biological brain as massively parallel computer. In 1986, Minsky published The Society of Mind, which claims that "mind is formed from many little agents, each mindless by itself".[82] The theory attempts to explain how what we call intelligence could be a product of the interaction of non-intelligent parts. Minsky says that the biggest source of ideas about the theory came from his work in trying to create a machine that uses a robotic arm, a video camera, and a computer to build with children's blocks.[83]

Similar models (which also view the biological brain as a massively parallel computer, i.e., the brain is made up of a constellation of independent or semi-independent agents) were also described by:

- Thomas R. Blakeslee,[84]
- Michael S. Gazzaniga,[85][86]
- Robert E. Ornstein,[87]
- Ernest Hilgard,[88][89]
- Michio Kaku,[90]
- George Ivanovich Gurdjieff,[91]
- Neurocluster Brain Model.[92]

See also

- Computer multitasking
- Concurrency (computer science)
- Content Addressable Parallel Processor
- List of distributed computing conferences
- Loop-level parallelism

• Manchester dataflow machine
• Manycore
• Parallel programming model
• Parallelization contract
• Serializability
• Synchronous programming
• Transputer
• Vector processing

References

1.
• Gottlieb, Allan; Almasi, George S. (1989). Highly parallel computing. Redwood City, Calif.: Benjamin/Cummings. ISBN 978-0-8053-0177-9.
• • S.V. Adve et al. (November 2008). "Parallel Computing Research at Illinois: The UPCRC Agenda" Archived 2018-01-11 at the Wayback Machine (PDF). Parallel@Illinois, University of Illinois at Urbana-Champaign. "The main techniques for these performance benefits—increased clock frequency and smarter but increasingly complex architectures—are now hitting the so-called power wall. The computer industry has accepted that future performance increases must largely come from increasing the number of processors (or cores) on a die, rather than making a single core go faster."
• • Asanovic et al. Old [conventional wisdom]: Power is free, but transistors are expensive. New [conventional wisdom] is [that] power is expensive, but transistors are "free".
• • Asanovic, Krste et al. (December 18, 2006). "The Landscape of Parallel Computing Research: A View from Berkeley" (PDF). University of California, Berkeley. Technical Report No. UCB/EECS-2006-183. "Old [conventional wisdom]: Increasing clock frequency is the primary method of improving processor performance. New [conventional wisdom]: Increasing parallelism is the primary method of improving processor performance… Even representatives from Intel, a company generally associated with the 'higher clock-speed is better' position, warned that traditional approaches to maximizing performance through maximizing clock speed have been pushed to their limits."
• • Parallel and Concurrent Programming in Haskell. O'Reilly Media. 2013. ISBN 9781449335922.
• • Hennessy, John L.; Patterson, David A.; Larus, James R. (1999). Computer organization and design: the hardware/software interface (2. ed., 3rd print. ed.). San Francisco: Kaufmann. ISBN 978-1-55860-428-5.
• • Barney, Blaise. "Introduction to Parallel Computing". Lawrence Livermore National Laboratory. Retrieved 2007-11-09.
• • Thomas Rauber; Gudula Rünger (2013). Parallel Programming: for Multicore and Cluster Systems. Springer Science & Business Media. p. 1. ISBN 9783642378010.
• • Hennessy, John L.; Patterson, David A. (2002). Computer architecture / a quantitative approach (3rd ed.). San Francisco, Calif.: International Thomson. p. 43. ISBN 978-1-55860-724-8.
• • Rabaey, Jan M. (1996). Digital integrated circuits : a design perspective. Upper Saddle River, N.J.: Prentice-Hall. p. 235. ISBN 978-0-13-178609-7.
• • Flynn, Laurie J. (8 May 2004). "Intel Halts Development Of 2 New Microprocessors". New York Times. Retrieved 5 June 2012.
• • Thomas Rauber; Gudula Rünger (2013). Parallel Programming: for Multicore and Cluster Systems. Springer Science & Business Media. p. 2. ISBN 9783642378010.
• • Thomas Rauber; Gudula Rünger (2013). Parallel Programming: for Multicore and Cluster Systems. Springer Science & Business Media. p. 3. ISBN 9783642378010.
• • Bakos, Jason D. (2016-01-01), Bakos, Jason D. (ed.), "Chapter 2 - Multicore and data-level optimization: OpenMP and SIMD", Embedded Systems, Boston: Morgan Kaufmann, pp. 49–103, doi:10.1016/b978-0-12-800342-8.00002-x, ISBN 978-0-12-800342-8, retrieved 2024-11-18
• • The Art of Multiprocessor Programming, Revised Reprint. Morgan Kaufmann. 22 May 2012. ISBN 9780123973375.
• • Vajda, András (10 June 2011). Programming Many-Core Chips. Springer. ISBN 9781441997395.
• • Amdahl, Gene M. (1967-04-18). "Validity of the single processor approach to achieving large

scale computing capabilities". Proceedings of the April 18-20, 1967, spring joint computer conference on - AFIPS '67 (Spring). New York, NY, USA: Association for Computing Machinery. pp. 483–485. doi:10.1145/1465482.1465560. ISBN 978-1-4503-7895-6.
• • Computer Architecture: A Quantitative Approach. Morgan Kaufmann. 2003. ISBN 978-8178672663.
• • Parallel Computer Architecture A Hardware/Software Approach. Elsevier Science. 1999. ISBN 9781558603431.
• • McCool, Michael; Reinders, James; Robison, Arch (2013). Structured Parallel Programming: Patterns for Efficient Computation. Elsevier. p. 61. ISBN 978-0-12-415993-8.
• • Gunther, Neil (2007). Guerrilla Capacity Planning: A Tactical Approach to Planning for Highly Scalable Applications and Services. ISBN 978-3540261384.
• • Bernstein, Arthur J. (1 October 1966). "Analysis of Programs for Parallel Processing". IEEE Transactions on Electronic Computers. EC-15 (5): 757–763. doi:10.1109/PGEC.1966.264565.
• • Roosta, Seyed H. (2000). Parallel processing and parallel algorithms : theory and computation. New York, NY [u.a.]: Springer. p. 114. ISBN 978-0-387-98716-3.
• • "Processes and Threads". Microsoft Developer Network. Microsoft Corp. 2018. Retrieved 2018-05-10.
• • Krauss, Kirk J (2018). "Thread Safety for Performance". Develop for Performance. Archived from the original on 2018-05-13. Retrieved 2018-05-10.
• • Tanenbaum, Andrew S. (2002-02-01). Introduction to Operating System Deadlocks. Pearson Education, Informit. Retrieved 2018-05-10. {{cite book}}: |website= ignored (help)
• • Cecil, David (2015-11-03). "Synchronization internals – the semaphore". Embedded. AspenCore. Retrieved 2018-05-10.
• • Preshing, Jeff (2012-06-08). "An Introduction to Lock-Free Programming". Preshing on Programming. Retrieved 2018-05-10.
• • "What's the opposite of "embarrassingly parallel"?". StackOverflow. Re
tshingombe fiston

---

techniques ingenieur defense material

---

tshingombe fiston Wed, Dec 4, 2024 at 2:56 PM
To: tshingombe fiston , tshigombekb@gmail.com
tshingombe fiston

JUST PUBLISHED by the IEC - Edition 2024-11

IEC JUST PUBLISHED
Tue, Dec 3, 2024 at 11:00 AM
Reply-To: IEC JUST PUBLISHED
To: tshingombefiston@gmail.com

New International Standards released by the IEC
Is this email not displaying correctly?
View it in your browser.

Just published

156 publications since 2024-11-01

Picks of the month
Commented version of IEC 60335-2-15:2024 CMV, Household and similar electrical appliances - Safety - Part 2-15: Particular requirements for appliances for heating liquids
Redline version of IEC TS 62600-100:2024 RLV, Marine energy - Wave, tidal and other water current converters - Part 100: Electricity producing wave energy converters - Power performance assessment
Extended version of IEC TS 62271-316:2024 EXV, High-voltage switchgear and controlgear - Part 316: Direct current by-pass switches and paralleling switches
Pre-release version of IEC 61109:2024 PRV, Insulators for overhead lines composite suspension and tension insulators with AC voltage greater than 1 000 V and DC voltage greater than 1 500 V - Definitions, test methods and acceptance criteria
Consolidated version of IEC 62288:2024 CSV, Maritime navigation and radiocommunication equipment and systems - Presentation of navigation-related information on shipborne navigational displays - General requirements, methods of testing and required test results
Technical report IEC TR 61328:2024, Live working - Guidelines for the installation of transmission and distribution line conductors and earth wires - Stringing equipment and accessory items
ISO/IEC 5259-2:2024, Artificial intelligence - Data quality for analytics and machine learning (ML) - Part 2: Data quality measures

New IEC Academy Masterclass - Introduction to the IEC 60335 series
On January 21st discover the essentials of the IEC 60335 series in this exclusive 4-hour online masterclass, delivered by IEC experts and designed to enhance your understanding of safety requirements for household and similar electrical appliances.
This course is ideal for technical experts, product designers, and professionals from manufacturing sites or testing labs who are new to the IEC 60335 series or seeking a refresher on its basics.
Register here.

IEC Products & Services Portal
Access freely all previews and order a subscription tailored to your needs:
- **Thematic collection**
- **Personalized selection of technical committees**
- **Personalized selection of IEC publications**

Graphical symbols (IEC 60417, IEC 60617, ISO 7000) are available in additional languages (CN, DK, FI, FR and JP).

IEC Glossary is also available. Consult more than 47 000 electrotechnical terms and definitions.

With a subscription to the IEC Products & Services Portal you will always have access to up to date content.

This IEC Just published email notification contains all publications issued last month. For a daily update of the list of released publications, please go to IEC Just published.

This is an automatic email notification. Please contact us if you have any questions and/or comments.

Company organization
IEC 63310:2024 PRV

Functional performance criteria for AAL robots used in connected home environment
ICS code 03.080, 11.180 | SyC AAL CHF 173 .-
ISO/IEC 19788-1:2024

Information technology for learning, education and training - Metadata for learning resources - Part 1: Framework
ICS code 03.100.30, 35.240.90 | ISO/IEC JTC 1/SC 36 CHF 194 .-

Electrical engineering
IEC 60034:2024 SER

Rotating electrical machines - ALL PARTS
ICS code 29.160.01 | TC 2 CHF 9247 .-
IEC 60071:2024 SER

Insulation co-ordination - ALL PARTS
ICS code 29.080.30 | TC 99 CHF 1542 .-
IEC 60079:2024 SER

Explosive atmospheres - ALL PARTS
ICS code 29.260.20 | TC 31 CHF 9450 .-
IEC 60086:2024 SER

Primary batteries - ALL PARTS
ICS code 29.220.10, 39.040.10 | TC 35 CHF 1516 .-
IEC 60092-378:2024

Electrical installations in ships - Part 378: Optical fiber cables
ICS code 29.060.20, 47.020.60 | SC 18A CHF 115 .-
IEC 60287:2024 SER

Electric cables - ALL PARTS
ICS code 29.060.20 | TC 20 CHF 1711 .-
IEC 60364-4-42:2024

Low-voltage electrical installations - Part 4-42: Protection for safety - Protection against thermal effects
ICS code 29.120.50, 91.140.50 | TC 64 CHF 235 .-
IEC 60502:2024 SER

Power cables with extruded insulation and their accessories for rated voltages from 1 kV (Um = 1,2 kV) up to 30 kV (Um = 36 kV) - ALL PARTS
ICS code 29.060.20 | TC 20 CHF 1211 .-
IEC 60598-1:2024 RLV

Luminaires - Part 1: General requirements and tests
ICS code 29.140.40 | SC 34D CHF 850 .-
IEC 60598-1:2024

Luminaires - Part 1: General requirements and tests
ICS code 29.140.40 | SC 34D CHF 425 .-
IEC 60664:2024 SER

Insulation coordination for equipment within low-voltage systems - ALL PARTS
ICS code 29.080.30 | TC 109 CHF 1114 .-
IEC 60664-1/AMD1:2024 PRV

Amendment 1 - Insulation coordination for equipment within low-voltage supply systems - Part 1: Principles, requirements and tests
ICS code 29.080.30 | TC 109 CHF 120 .-
IEC 60884-2-1:2024 PRV

Plugs and socket-outlets for household and similar purposes - Part 2-1: Particular requirements for fused plugs
ICS code 29.120.30 | SC 23B CHF 88 .-
IEC 60884-2-2:2024 PRV

Plugs and socket-outlets for household and similar purposes - Part 2-2: Particular requirements for socket-outlets for appliances
ICS code 29.120.30 | SC 23B CHF 176 .-
IEC 60884-2-3:2024 PRV

Plugs and socket-outlets for household and similar purposes - Part 2-3: Particular requirements for switched socket-outlets without interlock for fixed installations
ICS code 29.120.30 | SC 23B CHF 254 .-
IEC 60884-2-6:2024 PRV

Plugs and socket-outlets for household and similar purposes - Part 2-6: Particular requirements for switched socket-outlets with interlock for fixed electrical installations
ICS code 29.120.30 | SC 23B CHF 254 .-
IEC 60884-2-7:2024 PRV

Plugs and socket-outlets for household and similar purposes - Part 2-7: Particular requirements for cord extension sets
ICS code 29.120.30 | SC 23B CHF 254 .-

IEC 60947:2024 SER

Low-voltage switchgear and controlgear - ALL PARTS
ICS code 29.130.20 | SC 121A CHF 6727 .-
IEC 60947-4-2:2020+AMD1:2024 CSV

Low-voltage switchgear and controlgear - Part 4-2: Contactors and motor-starters - Semiconductor motor controllers, starters and soft-starters
ICS code 29.130.20 | SC 121A CHF 800 .-
IEC 60947-4-2:2020/AMD1:2024

Amendment 1 - Low-voltage switchgear and controlgear - Part 4-2: Contactors and motor-starters - Semiconductor motor controllers, starters and soft-starters
ICS code 29.130.20 | SC 121A CHF 80 .-
IEC 61008-1:2024

Residual current operated circuit-breakers without integral overcurrent protection for household and similar uses (RCCBs) - Part 1: General rules
ICS code 29.120.50 | SC 23E CHF 375 .-
IEC 61008-2-1:2024

Residual current operated circuit-breakers without integral overcurrent protection for household and similar uses (RCCBs) - Part 2-1: RCCBs according to classification 4.1.1
ICS code 29.120.50 | SC 23E CHF 150 .-
IEC 61008-2-2:2024

Residual current operated circuit-breakers without integral overcurrent protection for household and similar uses (RCCBs) - Part 2-2: RCCBs according to classification 4.1.2, 4.1.3, 4.1.4, 4.1.5 and 4.1.6
ICS code 29.120.50 | SC 23E CHF 270 .-
IEC 61009-1:2024

Residual current operated circuit-breakers with integral overcurrent protection for household and similar uses (RCBOs) - Part 1: General rules
ICS code 29.120.50 | SC 23E CHF 400 .-
IEC 61009-2-1:2024

Residual current operated circuit-breakers with integral overcurrent protection for household and similar uses (RCBOs) - Part 2-1: RCBOs according to classification 4.1.1
ICS code 29.120.50 | SC 23E CHF 150 .-
IEC 61009-2-2:2024

Residual current operated circuit-breakers with integral overcurrent protection for household and similar uses (RCBOs) - Part 2-2: RCBOs according to classification 4.1.2, 4.1.3, 4.1.4, 4.1.5 and 4.1.6
ICS code 29.120.50 | SC 23E CHF 235 .-
IEC 61109:2024 PRV

Insulators for overhead lines composite suspension and tension insulators with AC voltage greater than 1 000 V and DC voltage greater than 1 500 V - Definitions, test methods and acceptance criteria
ICS code 29.080.10 | TC 36 CHF 750 .-
IEC 62271:2024 SER

High-voltage switchgear and controlgear - ALL PARTS
ICS code 29.130.10, 29.130.99 | SC 17A CHF 14515 .-

IEC TS 62271-316:2024 EXV

High-voltage switchgear and controlgear - Part 316: Direct current by-pass switches and paralleling switches
ICS code 29.130.10 | SC 17A CHF 723 .-
IEC TS 62271-316:2024

High-voltage switchgear and controlgear - Part 316: Direct current by-pass switches and paralleling switches
ICS code 29.130.10 | SC 17A CHF 345 .-
IEC 62305:2024 SER

Protection against lightning - ALL PARTS
ICS code 29.020, 91.120.40 | TC 81 CHF 1313 .-
IEC 62683-2-3:2024

Low-voltage switchgear and controlgear – Product data and properties for information exchange – Engineering data – Part 2-3: Functional safety and reliability
ICS code 29.130.20 | TC 121 CHF 115 .-
IEC 62868-1/AMD1:2024 PRV

Amendment 1 - Organic light emitting diode (OLED) light sources for general lighting - Safety - Part 1: General requirements and tests
ICS code 29.140.99 | SC 34A CHF 60 .-
IEC 62868-2-1/AMD1:2024 PRV

Amendment 1 - Organic light emitting diode (OLED) light sources for general lighting - Safety - Part 2-1: Particular requirements - Semi-integrated OLED modules
ICS code 29.140.99 | SC 34A CHF 30 .-
IEC 62868-2-2/AMD1:2024 PRV

Amendment 1 - Organic light emitting diode (OLED) light sources for general lighting - Safety - Part 2-2: Particular requirements - Integrated OLED modules
ICS code 29.140.99 | SC 34A CHF 30 .-
IEC 62868-2-3/AMD1:2024 PRV

Amendment 1 - Organic light emitting diode (OLED) light sources for general lighting - Safety - Part 2-3: Particular requirements - Flexible OLED tiles and panels
ICS code 29.140.99 | SC 34A CHF 30 .-
IEC 62868-2-4:2024 PRV

Organic light emitting diode (OLED) light sources for general lighting - Safety - Part 2-4: Particular requirements - Rigid OLED tiles and panels
ICS code 29.140.99 | SC 34A CHF 60 .-
IEC TS 63290:2024

Supplementary requirements for intelligent assemblies
ICS code 29.130.20 | SC 121B CHF 300 .-
IEC TR 63482:2024/COR1:2024

Corrigendum 1 - Maintenance of low voltage switchgear and controlgear and their assemblies
ICS code 29.130.20 | TC 121 CHF 0 .-
IEC 63522-8:2024

Electrical relays - Tests and measurements - Part 8: Timing
ICS code 29.120.70 | TC 94 CHF 115 .-
IEC 63522-35:2024 PRV

Electrical relays - Tests and measurements - Part 35: Resistance to cleaning solvents
ICS code 29.120.70 | TC 94 CHF 60 .-
IEC/IEEE 62271-37-013:2021/COR1:2024

Corrigendum 1 - High-voltage switchgear and controlgear - Part 37-013: Alternating current generator circuit-breakers
ICS code 29.130.10 | SC 17A CHF 0 .-

Health
IEC 60204:2024 SER

Safety of machinery - Electrical equipment of machines - ALL PARTS
ICS code 13.110, 29.020, 33.100.10 | TC 44 CHF 2090 .-
IEC 60332:2024 SER

Tests on electric and optical fibre cables under fire conditions - ALL PARTS
ICS code 13.220.40, 29.020, 29.060.20 | TC 20 CHF 880 .-
IEC 60335-2-14:2024 PRV

Household and similar electrical appliances - Safety - Part 2-14: Particular requirements for kitchen machines
ICS code 13.120, 97.040.50 | TC 61 CHF 750 .-
IEC 60335-2-15:2024 CMV

Household and similar electrical appliances - Safety - Part 2-15: Particular requirements for appliances for heating liquids
ICS code 13.120, 97.040.50 | TC 61 CHF 540 .-
IEC 60335-2-15:2024 EXV-CMV

Household and similar electrical appliances - Safety - Part 2-15: Particular requirements for appliances for heating liquids
ICS code 13.120, 97.040.50 | TC 61 CHF 1167 .-
IEC 60335-2-15:2024 EXV

Household and similar electrical appliances - Safety - Part 2-15: Particular requirements for appliances for heating liquids
ICS code 13.120, 97.040.50 | TC 61 CHF 757 .-
IEC 60335-2-15:2024

Household and similar electrical appliances - Safety - Part 2-15: Particular requirements for appliances for heating liquids
ICS code 13.120, 97.040.50 | TC 61 CHF 270 .-
IEC 60335-2-45:2024 CMV

Household and similar electrical appliances - Safety - Part 2-45: Particular requirements for portable heating tools and similar appliances
ICS code 13.120, 25.140.20 | TC 61 CHF 380 .-
IEC 60335-2-45:2024 EXV-CMV

Household and similar electrical appliances - Safety - Part 2-45: Particular requirements for portable heating tools and similar appliances
ICS code 13.120, 25.140.20 | TC 61 CHF 1023 .-
IEC 60335-2-45:2024 EXV

Household and similar electrical appliances - Safety - Part 2-45: Particular requirements for portable heating tools and similar appliances
ICS code 13.120, 25.140.20 | TC 61 CHF 757 .-
IEC 60335-2-45:2024

Household and similar electrical appliances - Safety - Part 2-45: Particular requirements for portable heating tools and similar appliances
ICS code 13.120, 25.140.20 | TC 61 CHF 190 .-
IEC 60335-2-74:2021+AMD1:2024 CSV

Household and similar electrical appliances - Safety - Part 2-74: Particular requirements for portable immersion heaters
ICS code 13.120, 97.040.50 | TC 61 CHF 165 .-
IEC 60335-2-74:2021/AMD1:2024

Amendment 1 - Household and similar electrical appliances - Safety - Part 2-74: Particular requirements for portable immersion heaters
ICS code 13.120, 97.040.50 | TC 61 CHF 20 .-
IEC 60364-5-52:2009+AMD1:2024 CSV

Low-voltage electrical installations - Part 5-52: Selection and erection of electrical equipment - Wiring systems
ICS code 13.260, 91.140.50 | TC 64 CHF 600 .-
IEC 60364-5-52:2009/AMD1:2024

Amendment 1 - Low-voltage electrical installations - Part 5-52: Selection and erection of electrical equipment - Wiring systems
ICS code 13.260, 91.140.50 | TC 64 CHF 20 .-
IEC TS 60695-2-20:2024 RLV

Fire hazard testing - Part 2-20: Glowing/hot-wire based test methods - Hot-wire ignition test (HWI) method - Apparatus, verification, test method and guidance
ICS code 13.220.40, 29.020 | TC 89 CHF 68 .-
IEC TS 60695-2-20:2024

Fire hazard testing - Part 2-20: Glowing/hot-wire based test methods - Hot-wire ignition test (HWI) method - Apparatus, verification, test method and guidance
ICS code 13.220.40, 29.020 | TC 89 CHF 40 .-
IEC 60825:2024 SER

Safety of laser products - ALL PARTS
ICS code 13.110, 31.260 | TC 76 CHF 2647 .-
IEC TR 61328:2024

Live working - Guidelines for the installation of transmission and distribution line conductors and earth wires - Stringing equipment and accessory items
ICS code 13.260, 29.240.20, 29.260.99 | TC 78 CHF 300 .-
IEC 61511:2024 SER

Functional safety - Safety instrumented systems for the process industry sector - ALL PARTS
ICS code 13.110, 25.040.01 | SC 65A CHF 1514 .-
IEC 63240-1:2024 RLV

Active assisted living (AAL) reference architecture and architecture model - Part 1: Reference architecture
ICS code 11.020.99, 11.180 | SyC AAL CHF 255 .-
IEC 63240-1:2024

Active assisted living (AAL) reference architecture and architecture model - Part 1: Reference architecture
ICS code 11.020.99, 11.180 | SyC AAL CHF 150 .-
IEC SRD 63408:2024

Safety aspects – Guidelines for adult AAL care recipients in standards and other specifications
ICS code 11.020.99, 11.180 | SyC AAL CHF 80 .-
ISO 80369-20:2024

Small-bore connectors for liquids and gases in healthcare applications - Part 20: Common test methods
ICS code 11.040.25 | SC 62D CHF 151 .-

Household appliances
IEC 60335-2-34:2024 CMV

Household and similar electrical appliances - Safety - Part 2-34: Particular requirements for motor-compressors
ICS code 97.040.30 | SC 61C CHF 540 .-
IEC 60335-2-34:2024 EXV-CMV

Household and similar electrical appliances - Safety - Part 2-34: Particular requirements for motor-compressors
ICS code 97.040.30 | SC 61C CHF 1167 .-
IEC 60335-2-34:2024 EXV

Household and similar electrical appliances - Safety - Part 2-34: Particular requirements for motor-compressors
ICS code 97.040.30 | SC 61C CHF 757 .-
IEC 60335-2-34:2024

Household and similar electrical appliances - Safety - Part 2-34: Particular requirements for motor-compressors
ICS code 97.040.30 | SC 61C CHF 270 .-
IEC 60456:2024

Washing machines for household use - Methods for measuring the performance
ICS code 97.060 | SC 59D CHF 425 .-
IEC 60705:2024

Household microwave ovens - Methods for measuring performance
ICS code 97.040.20 | SC 59K CHF 235 .-
IEC/ASTM 62885-6:2023/COR1:2024

Corrigendum 1 - Surface cleaning appliances - Part 6: Wet hard floor cleaning appliances for household or similar use - Methods for measuring the performance
ICS code 97.080 | SC 59F CHF 0 .-

IT equipment
ISO/IEC TR 11801-9911:2024

Information technology - Generic cabling for customer premises - Part 9911: Guidelines for the use of balanced single pair applications within a balanced 4-pair cabling system
ICS code 35.200 | ISO/IEC JTC 1/SC 25 CHF 129 .-
ISO/IEC/IEEE 8802-1Q:2024/AMD35:2024

Telecommunications and exchange between information technology systems - Requirements for local and metropolitan area networks - Part 1Q: Bridges and bridged networks - Amendment 35: Congestion isolation
ICS code 35.110 | ISO/IEC JTC 1/SC 6 CHF 216 .-
ISO/IEC/IEEE 8802-1AE:2020/AMD4:2024

Telecommunications and exchange between information technology systems - Requirements for local and metropolitan area networks - Part 1AE: Media access control (MAC) security - Amendment 4: MAC Privacy Protection
ICS code 35.110 | ISO/IEC JTC 1/SC 6 CHF 216 .-

IT security
ISO/IEC 9797-2:2021/COR1:2024

Information security - Message authentication codes (MACs) - Part 2: Mechanisms using a dedicated hash-function - Technical Corrigendum 1
ICS code 35.030 | ISO/IEC JTC 1/SC 27 CHF 0 .-
ISO/IEC 18014-2:2021/COR1:2024

Information security - Time-stamping services - Part 2: Mechanisms producing independent tokens - Technical Corrigendum 1
ICS code 35.030 | ISO/IEC JTC 1/SC 27 CHF 0 .-

IT systems
IEC 61131:2024 SER

Programmable controllers - ALL PARTS
ICS code 35.040.40, 35.240.50 | SC 65B CHF 3052 .-
ISO/IEC 5259-2:2024

Artificial intelligence - Data quality for analytics and machine learning (ML) - Part 2: Data quality measures
ICS code 35.020 | ISO/IEC JTC 1/SC 42 CHF 173 .-
ISO/IEC TS 9922:2024

Programming Languages - Technical specification for C++ extensions for concurrency 2
ICS code 35.060 | ISO/IEC JTC 1/SC 22 CHF 129 .-
ISO/IEC TS 10866:2024

Information technology - Cloud computing and distributed platforms - Framework and concepts for organizational autonomy and digital sovereignty
ICS code 35.210 | ISO/IEC JTC 1/SC 38 CHF 96 .-
ISO/IEC 14496-26:2024

Information technology - Coding of audio-visual objects - Part 26: Audio conformance
ICS code 35.040.40 | ISO/IEC JTC 1/SC 29 CHF 216 .-
ISO/IEC 15444-1:2024

Information technology - JPEG 2000 image coding system - Part 1: Core coding system
ICS code 35.040.30 | ISO/IEC JTC 1/SC 29 CHF 216 .-
ISO/IEC TS 18013-6:2024

Personal identification - ISO-compliant driving licence - Part 6: mDL test methods
ICS code 35.240.15 | ISO/IEC JTC 1/SC 17 CHF 194 .-
ISO/IEC 18975:2024

Information technology - Automatic identification and data capture techniques - Encoding and resolving identifiers over HTTP
ICS code 35.040.50 | ISO/IEC JTC 1/SC 31 CHF 96 .-
ISO/IEC TS 21419:2024

Information technology - Cross-jurisdictional and societal aspects of implementation of biometric technologies - Use of biometrics for identity management in healthcare
ICS code 35.240.80, 35.240.15 | ISO/IEC JTC 1/SC 37 CHF 129 .-
ISO/IEC 21794-5:2024

Information technology - Plenoptic image coding system (JPEG Pleno) - Part 5: Holography
ICS code 35.040.30 | ISO/IEC JTC 1/SC 29 CHF 194 .-
ISO/IEC TS 23220-2:2024

Cards and security devices for personal identification - Building blocks for identity management via mobile devices - Part 2: Data objects and encoding rules for generic eID systems
ICS code 35.240.15 | ISO/IEC JTC 1/SC 17 CHF 129 .-
ISO/IEC 30181:2024

Internet of Things (IoT) - Functional architecture for resource identifier interoperability
ICS code 35.020 | ISO/IEC JTC 1/SC 41 CHF 151 .-

Manufacturing
IEC TS 62453-43:2024

Field device tool (FDT) interface specification – Part 43: Object model integration profile – CLI and HTML
ICS code 25.040.40, 35.100.05, 35.110 | SC 65E CHF 425 .-
IEC 62541-15:2024 PRV

OPC Unified Architecture - Part 15: Safety
ICS code 25.040.40 | SC 65C CHF 563 .-
IEC 63261:2024

Representation of electrical and instrument objects in digital 3D plant models during engineering
ICS code 25.040.40 | SC 65E CHF 80 .-

IEC 63270-1:2024 PRV

Predictive maintenance of industrial automation equipment and systems - Part 1: General requirements
ICS code 25.040.01 | SC 65E CHF 405 .-
IEC TR 63283-5:2024

Industrial-process measurement, control and automation – Smart manufacturing – Part 5: Market and innovation trends analysis
ICS code 25.040.40 | TC 65 CHF 345 .-

Ships
IEC 61892:2024 SER

Mobile and fixed offshore units - Electrical installations - ALL PARTS
ICS code 47.020.60, 29.260.99 | TC 18 CHF 1607 .-
IEC 62288:2021+AMD1:2024 CSV

Maritime navigation and radiocommunication equipment and systems - Presentation of navigation-related information on shipborne navigational displays - General requirements, methods of testing and required test results
ICS code 47.020.70 | TC 80 CHF 800 .-
IEC 62288:2021/AMD1:2024

Amendment 1 - Maritime navigation and radiocommunication equipment and systems - Presentation of navigation-related information on shipborne navigational displays - General requirements, methods of testing and required test results
ICS code 47.020.70 | TC 80 CHF 10 .-

Telecommunications
CISPR TR 31:2024

Description of the radio services database
ICS code 33.100.10 | CIS/H CHF 115 .-
IEC 60793-1-40:2024 RLV

Optical fibres - Part 1-40: Attenuation measurement methods
ICS code 33.180.10 | SC 86A CHF 400 .-
IEC 60793-1-40:2024

Optical fibres - Part 1-40: Attenuation measurement methods
ICS code 33.180.10 | SC 86A CHF 235 .-
IEC 60870-5:2024 SER

Telecontrol equipment and systems - Part 5: Transmission protocols - ALL PARTS
ICS code 33.200 | TC 57 CHF 3500 .-
IEC 60958:2024 SER

Digital audio interface - ALL PARTS
ICS code 33.160.01, 33.160.30 | TC 100 CHF 978 .-
IEC 61000-3:2024 SER

Electromagnetic compatibility (EMC) - Part 3: Limit - ALL PARTS
ICS code 33.100.10 | SC 77A CHF 3200 .-
IEC 61000-4-41:2024

Electromagnetic compatibility (EMC) - Part 4-41: Testing and measurement techniques - Broadband radiated immunity tests
ICS code 33.100.20 | SC 77B CHF 300 .-
IEC 61196-1-108:2024 PRV

Coaxial communication cables - Part 1-108: Electrical test methods - Test for phase, phase constant, phase and group delay, propagation velocity, electrical length, and mean characteristic impedance
ICS code 33.120.10 | SC 46A CHF 173 .-
IEC 61850:2024 SER

Communication networks and systems for power utility automation - ALL PARTS
ICS code 33.200 | TC 57 CHF 18743 .-
IEC 61850-6:2009+AMD1:2018+AMD2:2024 CSV

Communication networks and systems for power utility automation - Part 6: Configuration description language for communication in electrical substations related to IEDs
ICS code 33.200 | TC 57 CHF 2082 .-
IEC 61850-6:2009/AMD2:2024

Amendment 2 - Communication networks and systems for power utility automation - Part 6: Configuration description language for communication in electrical substations related to IEDs
ICS code 33.200 | TC 57 CHF 375 .-
IEC 61937:2024 SER

Digital audio - Interface for non-linear PCM encoded audio bitstreams applying IEC 60958 - ALL PARTS
ICS code 33.160.30, 33.160.60, 35.040.40 | TC 100 CHF 1712 .-
IEC 61937-16:2024

Digital audio – Interface for non-linear PCM encoded audio bitstreams applying IEC 60958 – Part 16: AVSA
ICS code 33.160.30, 35.040.40 | TA 20 CHF 40 .-
IEC 61970:2024 SER

Energy management system application program interface (EMS-API) - ALL PARTS
ICS code 33.200 | TC 57 CHF 3885 .-
IEC 62148-2:2010+AMD1:2024 CSV

Fibre optic active components and devices - Package and interface standards - Part 2: SFF 10-pin transceivers
ICS code 33.180.01 | SC 86C CHF 165 .-
IEC 62148-2:2010/AMD1:2024

Amendment 1 - Fibre optic active components and devices - Package and interface standards - Part 2: SFF 10-pin transceivers
ICS code 33.180.01 | SC 86C CHF 10 .-
IEC 62351:2024 SER

Power systems management and associated information exchange - Data and communications

security - ALL PARTS
ICS code 33.200 | TC 57 CHF 5359 .-

Terminology and metrology
IEC 60060:2024 SER

High-voltage test techniques - ALL PARTS
ICS code 17.220.20, 19.080 | TC 42 CHF 786 .-
IEC 60092:2024 SER

Electrical installations in ships - ALL PARTS
ICS code 01.040.47, 47.020.60 | TC 18 CHF 4413 .-
IEC 61340-4-11:2024 PRV

Electrostatics - Part 4-11: Standard test methods for specific applications - Testing of electrostatic properties of composite IBC
ICS code 17.220.99, 29.020, 55.080 | TC 101 CHF 225 .-
IEC 61869-20:2024 PRV

Instrument transformers – Part 20: Safety requirements of instrument transformers for high voltage applications
ICS code 17.220.20 | TC 38 CHF 120 .-
IEC/IEEE 63253-5713-8:2024

Station Service Voltage Transformers (SSVT)
ICS code 17.220.20 | TC 38 CHF 345 .-

Testing
IEC 60068-2:2024 SER

Environmental testing - Part 2: Tests - ALL PARTS
ICS code 19.040 | TC 104 CHF 6956 .-

Textile
IEC 63203-204-2:2024 PRV

Wearable electronic devices and technologies - Part 204-2: Electronic textile - Test method to characterize electrical resistance change in knee and elbow bending test of e-textiles
ICS code 59.080.80, 59.080.01 | TC 124 CHF 285 .-

IECEE Test Report Forms
IECEE TRF 60227-5J:2024

This Test Report Form applies to: IEC 60227-5:2024 in conjunction with IEC 60227-1:2024, IEC 63294:2021
CHF 550 .-
IECEE TRF 60227-5I:2024

This Test Report Form applies to: IEC 60227-5:2024 in conjunction with IEC 60227-1:2024, IEC 63294:2021

CHF 550 .-
IECEE TRF 60335-2-6S:2024

This Test Report Form applies to: IEC 60335-2-6:2024 in conjunction with IEC 60335-1:2020
CHF 1600 .-
IECEE TRF 60335-2-24U:2024

This Test Report Form applies to: IEC 60335-2-24:2020 in conjunction with IEC 60335-1:2010, IEC 60335-1:2010/AMD1:2013, IEC 60335-1:2010/AMD2:2016
CHF 1600 .-
IECEE TRF 60601-2-37H:2024

This Test Report Form applies to: IEC 60601-2-37:2024 for use in conjunction with IEC 60601-1:2005, IEC 60601-1:2005/AMD1:2012, IEC 60601-1/AMD2:2020
CHF 550 .-
IECEE TRF 60947-2L:2024

This Test Report Form applies to: IEC 60947-2:2024 for use in conjunction IEC 60947-1:2020
CHF 1600 .-
IECEE TRF 61000-6-2,4B:2024

This Test Report applies to: IEC 61000-6-2:2016 & IEC 61000-6-4:2018
CHF 1100 .-
IECEE TRF 61010-2-032G:2024

This Test Report Form applies to: IEC 61010-2-032:2019 used in conjunction with IEC 61010-1:2010, IEC 61010-1:2010/AMD1:2016
CHF 1100 .-
IECEE TRF 62477-1F:2024

This Test Report Form applies to: IEC 62477-1:2022
CHF 1100 .-
IECEE TRF 80601-2-55E:2024

This Test Report Form applies to: ISO 80601-2-55:2018, ISO 80601-2-55:2018/AMD1:2023 for use with IEC 60601-1:2005, IEC 60601-1:2005/AMD1:2012, IEC 60601-1:2005/AMD2:2020
CHF 1100 .-
IECEx TRF 60079-0v7j_ds:2024

IECEx Test Report for IEC 60079-0:2017 edition 7.0, Explosive atmospheres - Part 0: Equipment - General requirements
CHF 235 .-
IECEx TRF 60079-0v6h_ds:2024

IECEx Test Report for IEC 60079-0:2011 edition 6.0, Explosive atmospheres - Part 0: Equipment - General requirements
CHF 190 .-
IECEx TRF 60079-46v1d_ds:2024

IECEx Test Report for IEC TS 60079-46:2017 edition 1.0, Explosive atmospheres - Part 46: Equipment assemblies
CHF 40 .-
IECEx TRF 80079-36v1c_ds:2024

IECEx Test Report for ISO 80079-36:2016 edition 1.0, Explosive atmospheres - Part 36: Non-electrical equipment for explosive atmospheres - Basic method and requirements
CHF 115 .-
IECEx TRF 80079-37v1b_ds:2024

IECEx Test Report for ISO 80079-37:2016 edition 1.0, Explosive atmospheres - Part 37: Non-electrical equipment for explosive atmospheres - Non electrical type of protection constructional safety "c", control of ignition source "b", liquid immersion "k"
CHF 115 .-

Withdrawn/Replaced publications
Publication TC/SC Replaced by
IEC 60169-10:1983
SC 46F IEC 61169-10:2024

IEC 61144:1992
TC 10 Withdrawn
IEC 61753-121-3:2010
SC 86B Withdrawn
IEC TR 62271-302:2010
SC 17A Withdrawn
IEC 61755-3-32:2015
SC 86B Withdrawn
IEC 61753-121-2:2017
SC 86B Withdrawn

Join our social media conversations

On Wed, Dec 4, 2024 at 1:30 PM tshingombe fiston wrote:

On Wed, Dec 4, 2024 at 1:10 PM tshingombe fiston wrote:

On Wed, Dec 4, 2024 at 12:47 PM tshingombe fiston wrote:
001d9cff-41b2-53a5-a9fc-cd064dee3084_2024_4e2c7ed2-46c7-5568-8711-94d763ff19a8_pdf_version-241030...
001d9cff-41b2-53a5-a9fc-cd064dee3084_2024_854e29bc-6187-5e5b-aef8-da2169b9bb2e_pdf_version-241030...
CoinbaseCommerce-product_checkouts-2024-12-03-2024-12-03-report.csv
download (1).jpg
download (3).png
livre_blanc_apprentissage-statistique-non-supervise.pdf
livre_blanc_comment-integrer-les-materiaux-plastiques-dans-une-economie-circulaire.pdf
livre_blanc_courbes-et-surfaces-pour-la-cfao-par-m-bezier-le-createur-des-courbes-de-bezier.pdf
livre_blanc_ecoconception-etat-de-lart-des-outils-disponibles.pdf
livre_blanc_ecoconduite-et-automobile-du-reve-a-la-realite.pdf
livre_blanc_fabrication-additive-impression-3d-dans-le-domaine-de-la-sante.pdf
livre_blanc_introduction-a-lintelligence-artificielle.pdf
livre_blanc_lecoconception-un-outil-dinnovation-pour-une-chimie-durable.pdf
livre_blanc_les-metavers-quelles-applications-pour-les-industries.pdf
livre_blanc_lubrification-en-mise-en-forme-frottement-et-usure-caracterisation.pdf
livre_blanc_mesures-estimer-les-incertitudes-et-fixer-les-tolerances.pdf
livre_blanc_performance-energetique-des-batiments-existants-etudes-de-cas.pdf
livre_blanc_vers-une-industrialisation-du-recyclage-des-emissions-de-co2.pdf

one note assessment tshingombe career saqa.pdf
tshinaiu invoice portofolio.pdf

technique ingenieure.docx
1824K View as HTML Download

To fully appreciate this email, please enable images
View this email in your browser

Dear All,

On behalf of Mr Wolfram Zeitz, the IECEE Secretariat is pleased to inform you that the following new documents have been added to the IECEE Web site:

REMINDER:
NCBs are requested to forward all relevant information to their associated Customers' Testing Facilities.

General

IECEE - Working groups
https://www.iecee.org/committees/cmc-working-groups-task-forces

The composition of the following WGs has been updated:
- **CTL Expert Task Force 04 "INST, CONT, CAP, MISC" change in membership: Mr Xiao Jarod (Lin)**
- **CTL Expert Task Force 05 "LITE, SAFE" change in membership: Mr Kilb Holger**
- **CTL Expert Task Force 06 "CABL" change in membership: Mr Park Gibeom**
- **CTL Expert Task Force 10 "EMC" change in membership: Mr Kim Taehoon**
- **CTL Expert Task Force 14 "EMobility" change in membership: Dr Kim Seung Joo**
- **CTL Expert Task Force 16 "Cyber Security" change in membership: Mr Ytzhaik Moshe**

CB Scheme

IECEE/IECEE CB Scheme/IEC Standards operated by the IECEE
https://www.iecee.org/certification/iec-standards

The following product category has been updated:
- **CABL ISO 19642-3:2019**
- **CABL ISO 19642-4:2019**
- **CABL ISO 19642-5:2019**

- CABL ISO 19642-6:2019
- CABL ISO 19642-7:2019
- CABL ISO 19642-8:2019
- CABL ISO 19642-9:2019
- CABL ISO 19642-10:2019

National and Group difference information is available under each particular standard.

Authorized Certificate Signatories
https://www.iecee.org/certification/authorized-cbtc-signatories

This page is updated on an ongoing basis.

IECEE/IECEE CB Scheme/IECEE Members
https://www.iecee.org/members

The following NCB scopes have been updated:
- NCB CVC Certification & Testing Co., Ltd. (https://www.iecee.org/members/national-certification-bodies/cvc-certification-testing-co-ltd)
- NCB Eurofins Electric & Electronic Product Testing AG (https://www.iecee.org/members/national-certification-bodies/eurofins-electric-electronic-product-testing-ag)
- NCB Hermon Laboratories Ltd (https://www.iecee.org/members/national-certification-bodies/hermon-laboratories-ltd)
- NCB Intertek Semko AB (https://www.iecee.org/members/national-certification-bodies/intertek-semko-ab)
- NCB Intertek Testing Services (Singapore) Pte Ltd (https://www.iecee.org/members/national-certification-bodies/intertek-testing-services-singapore-pte-ltd)
- NCB Korea Testing Laboratory (KTL) (https://www.iecee.org/members/national-certification-bodies/korea-testing-laboratory-ktl)
- NCB LABORATOIRE CENTRAL DES INDUSTRIES ELECTRIQUES – LCIE (https://www.iecee.org/members/national-certification-bodies/laboratoire-central-des-industries-electriques-lcie)
- NCB SIQ Ljubljana (https://www.iecee.org/members/national-certification-bodies/siq-ljubljana)
- NCB TÜV Rheinland InterCert Kft., MEEI Division (https://www.iecee.org/members/national-certification-bodies/tuv-rheinland-intercert-kft-meei-division)
- NCB TÜV Rheinland Japan Ltd. (https://www.iecee.org/members/national-certification-bodies/tuv-rheinland-japan-ltd)
- NCB TÜV Rheinland LGA Products GmbH (https://www.iecee.org/members/national-certification-bodies/tuv-rheinland-lga-products-gmbh)
- NCB VDE Prüf- und Zertifizierungsinstitut GmbH (https://www.iecee.org/members/national-certification-bodies/vde-pruf-und-zertifizierungsinstitut-gmbh)

The following TL scopes have been updated:
- CBTL CVC Testing Technology (Jiaxing) Co., Ltd. (https://www.iecee.org/members/cbtls/cvc-testing-technology-jiaxing-co-ltd)
- CBTL Centro de Ensayos Innovación y Servicios (CEIS) S.L. (https://www.iecee.org/members/cbtls/centro-de-ensayos-innovacion-y-servicios-ceis-sl)
- CBTL Eurofins Electrical Testing Service (Shenzhen) Co., Ltd. (https://www.iecee.org/members/cbtls/eurofins-electrical-testing-service-shenzhen-co-ltd)
- CBTL HCT Co., Ltd. (https://www.iecee.org/members/cbtls/hct-co-ltd-29791)
- CBTL HERMON LABORATORIES Ltd. (https://www.iecee.org/members/cbtls/hermon-laboratories-

ltd)
• CBTL Hunan Electric Research Institute Testing Group Co., Ltd. (https://www.iecee.org/members/cbtls/hunan-electric-research-institute-testing-group-co-ltd)
• CBTL Intertek Testing Services Shenzhen Ltd. Guangzhou Branch (https://www.iecee.org/members/cbtls/intertek-testing-services-shenzhen-ltd-guangzhou-branch-14445)
• CBTL Intertek Testing Services Zhejiang Ltd. (https://www.iecee.org/members/cbtls/intertek-testing-services-zhejiang-ltd)
• CBTL LCIE CHINA Company limited (https://www.iecee.org/members/cbtls/lcie-china-company-limited)
• CBTL Mettler-Toledo GmbH (https://www.iecee.org/members/cbtls/mettler-toledo-gmbh)
• CBTL Prof. Ir. Damstra Laboratory (https://www.iecee.org/members/cbtls/prof-ir-damstra-laboratory)
• CBTL SGS Brightsight Barcelona S.L., Madrid office (https://www.iecee.org/members/cbtls/sgs-brightsight-barcelona-sl-madrid-office)
• CBTL SGS-CSTC Standards Technical Services Co., Ltd. Shenzhen Branch (https://www.iecee.org/members/cbtls/sgs-cstc-standards-technical-services-co-ltd-shenzhen-branch)
• CBTL Shandong Institute for Product Quality Inspection (SDQI) (https://www.iecee.org/members/cbtls/shandong-institute-product-quality-inspection-sdqi)
• CBTL TÜV Rheinland (Shanghai) Co., Ltd. (https://www.iecee.org/members/cbtls/tuv-rheinland-shanghai-co-ltd-13319)
• CBTL TÜV Rheinland (Shenzhen) Co., Ltd. (https://www.iecee.org/members/cbtls/tuv-rheinland-shenzhen-co-ltd)
• CBTL TÜV Rheinland InterCert Kft., MEEI Division (https://www.iecee.org/members/cbtls/tuv-rheinland-intercert-kft-meei-division)
• CBTL TÜV Rheinland Japan, Ltd. Yokohama Laboratory (https://www.iecee.org/members/cbtls/tuv-rheinland-japan-ltd-yokohama-laboratory)
• CBTL TÜV Rheinland LGA Products GmbH Nürnberg (https://www.iecee.org/members/cbtls/tuv-rheinland-lga-products-gmbh-nurnberg)
• CBTL TÜV SÜD America, Inc., Alpharetta GA (https://www.iecee.org/members/cbtls/tuv-sud-america-inc-alpharetta-ga)
• CBTL TÜV SÜD Product Service GmbH - Straubing (https://www.iecee.org/members/cbtls/tuv-sud-product-service-gmbh-straubing)
• CBTL UL Solutions Northbrook (https://www.iecee.org/members/cbtls/ul-solutions-northbrook-14422)
• CBTL Underwriters Laboratories Taiwan Co., Ltd (https://www.iecee.org/members/cbtls/underwriters-laboratories-taiwan-co-ltd)
• CBTL VDE Prüf- und Zertifizierungsinstitut GmbH (https://www.iecee.org/members/cbtls/vde-pruf-und-zertifizierungsinstitut-gmbh)
• CBTL Zhejiang Academy of Science and Technology for Inspection & Quarantine (https://www.iecee.org/members/cbtls/comprehensive-technical-service-center-ruian-branch-wenzhou-customs)
• CBTL Zhejiang Testing & Inspection Institute for Mechanical and Electrical Products Quality Co.,Ltd (ZTME) (https://www.iecee.org/members/cbtls/zhejiang-testing-inspection-institute-mechanical-and-electrical-products-quality-coltd-ztme)

IECEE - Acceptance Certificates
https://www.iecee.org/members

The following NCB and CBTL acceptance certificates have been updated:

• CBTL Centro de Ensayos Innovación y Servicios (CEIS) S.L. (https://www.iecee.org/members/cbtls/centro-de-ensayos-innovacion-y-servicios-ceis-sl)
• CBTL Intertek Testing & Certification Ltd. - Milton Keynes (https://www.iecee.org/members/cbtls/intertek-testing-certification-ltd-milton-keynes)
• CBTL Mettler-Toledo GmbH (https://www.iecee.org/members/cbtls/mettler-toledo-gmbh)
• CBTL SGS North America, Inc. (https://www.iecee.org/members/cbtls/sgs-north-america-inc)
• CBTL SGS-CSTC Standards Technical Services Co., Ltd. Shenzhen Branch (https://www.iecee.org/members/cbtls/sgs-cstc-standards-technical-services-co-ltd-shenzhen-branch)
• CBTL SGS-CSTC Standards Technical Services Co., Ltd. Shunde Branch (https://www.iecee.org/members/cbtls/sgs-cstc-standards-technical-services-co-ltd-shunde-branch)
• CBTL TÜV SÜD America, Inc., Alpharetta GA (https://www.iecee.org/members/cbtls/tuv-sud-america-inc-alpharetta-ga)
• SPTL CCIC-CSA International Certification Co., Ltd Shanghai Branch (https://www.iecee.org/members/sptls/ccic-csa-international-certification-co-ltd-shanghai-branch)

Peer assessment

IECEE/Peer assessment/Assessment reports PUBLISHED
https://www.iecee.org/peerassessment

Report number Type of assessment Organization Download
IECEE-PAC/24-124/RAR Re-Assessment CBTL Association of Polish Electricians – Quality Testing Office (SEP – BBJ) Testing Laboratory LUBLIN Download

IECEE-PAC/24-123/RAR Re-Assessment CBTL Association of Polish Electricians – Quality Testing Office (SEP – BBJ) Testing Laboratory WARSAW Download

IECEE-PAC/24-125/RAR Re-Assessment CBTL Cosmos Corporation Matsusaka lab. Download

IECEE-PAC/24-236/RAR Re-Assessment + Extension Assessment CBTL Cosmos Corporation Matsusaka lab. Download

IECEE-PAC/24A-153/IAR Initial Assessment CBTL Guangdong U.K Testing & Certification Co., Ltd. Download

IECEE-PAC/24-173/RAR Re-Assessment CBTL Kiwa Dare B.V. Download

IECEE-PAC/24-149/RAR Re-Assessment CBTL QIMA Testing (Shanghai) Limited Download

IECEE-PAC/24-166/RAR Re-Assessment + Extension Assessment CBTL STC (Guangdong) Company Limited (GDSTC) Download

IECEE-PAC/24-139/RAR Re-Assessment CBTL TUV Rheinland (India) Pvt. Ltd. Download

IECEE-PAC/24-245/RAR Re-Assessment CBTL TUV Rheinland (India) Pvt. Ltd., Gurugram Download

IECEE-PAC/24-134/RAR Re-Assessment CBTL TÜV Rheinland Taiwan Ltd., Taoyuan Testing Laboratories Download

IECEE-PAC/24-016/RAR Re-Assessment + Extension Assessment CBTL TÜV SÜD Product Service GmbH Munich Download

IECEE-PAC/24-074/RAR Re-Assessment NCB Association of Polish Electricians – Quality Testing Office (SEP – BBJ) Download

IECEE-PAC/24-235/RAR Re-Assessment + Extension Assessment NCB Cosmos Corporation Download

IECEE-PAC/24A-109/IAR Initial Assessment NCB EMC-Testcenter AG Download

IECEE-PAC/24-017/RAR Re-Assessment NCB TÜV SÜD Product Service GmbH Download

IECEE-PAC/23-157/RAR Re-Assessment SPTL Element Materials Technology (Skelmersdale) Download

IECEE-PAC/24-260/RAR Re-Assessment SPTL TÜV SÜD Product Service GmbH, Munich Download

NOTE: due to the large file size of the reports provided, it may take a few minutes to open/save them

IMPORTANT:
In order to timely process the evaluation of the above Re-Assessment Reports, Members are invited to send their comments, if any, within one month of this notification.

NOTE:
- **IAR stands for Initial Assessment Report**
- **EAR stands for Scope Extension Assessment Report**
- **RAR stands for Re-assessment Report**
- **RLAR stands for Re-location Assessment Report**
- **FAR stands for Follow-up Assessment Report**
- **RWC stands for Reviewer Comments**
- **MBC stands for Member Comments**
- **LAR stands for Lead Assessor Reply**

IECEE/Peer assessment/Assessment reports CLEARED
https://www.iecee.org/peerassessment

Report number Type of assessment Organization Download
IECEE-PAC/24-019/RAR Re-Assessment CBTL CQCTS (CQC Testing Technical Services Co. , Ltd.) Download

IECEE-PAC/23-003/RAR Re-Assessment + Extension Assessment CBTL Centro de Ensayos Innovación y Servicios (CEIS) S.L. Download

IECEE-PAC/23-114/RAR Re-Assessment CBTL Intertek Testing & Certification Ltd. - Milton Keynes Download

IECEE-PAC/24-005/RAR Re-Assessment + Extension Assessment CBTL SGS-CSTC Standards Technical Services Co., Ltd. Shenzhen Branch Download

IECEE-PAC/24-007/RAR Re-Assessment CBTL SGS-CSTC Standards Technical Services Co., Ltd. Shunde Branch Download

IECEE-PAC/24-133/RAR Re-Assessment + Extension Assessment CBTL TÜV SÜD America, Inc., Alpharetta GA Download

IECEE-PAC/24-233/RAR Re-Assessment + Relocation Assessment SPTL CCIC-CSA International Certification Co., Ltd Shanghai Branch Download

NOTE: due to the large file size of the reports provided, it may take a few minutes to open/save them

NOTE:
- **IAR stands for Initial Assessment Report**
- **EAR stands for Scope Extension Assessment Report**
- **RAR stands for Re-assessment Report**
- **RLAR stands for Re-location Assessment Report**
- **FAR stands for Follow-up Assessment Report**
- **RWC stands for Reviewer Comments**
- **MBC stands for Member Comments**
- **LAR stands for Lead Assessor Reply**

Test Report Forms (TRFs)

IECEE/Restricted area/IEC & IECEN TRFs
https://www.iecee.org/certification/iec-test-report-forms

The following TRFs (including where applicable Group Differences and National Difference) have been published:

Product Category Standard reference TRF No. TRF Master Date TRF originator
HOUS IEC 60335-2-96:2024 in conjunction with IEC 60335-1:2020
IEC60335_2_96J 2024-12-06 DEKRA Certification B.V.
HOUS IEC 61770:2008, IEC 61770:2008/AMD1:2015 EU_GD_IEC61770D_II
European Group Differences Addendum to IEC61770D 2024-12-06 IMQ S.p.A.

The following lists of TRFs available lists has been updated:

TRFs in preparation (https://www.iecee.org/certification/iec-test-report-forms)

Committee of Testing Laboratories (CTL)

IECEE/CTL/ETFs & working groups

https://www.iecee.org/committees/ctl-working-groups-task-forces

The composition of the following ETFs have been updated:

ETF 06, ETF 05, ETF 03, ETF 16, ETF 04, ETF 10, ETF 14

Have a nice day and should you have any questions or should you experience any problems, please do not hesitate to contact the IECEE Secretariat (secretariat@iecee.org).

With best regards,
IECEE Secretariat

Installment payment for $ 153.30 / $3679

All fees need to be paid in USD. The applicable bank commission or charges to be borne by the candidate and the net amount
payable to be paid into the receiver's account.
DESCRIPTION UNIT PRICE QTY AMOUNT
1st Installment payment for Master Of MEP
Engineering and Project Management(472 hrs)
1st Installment payment for Master Of MEP Engineering an d Project Management(472 hrs)

$ 153.30 1 $ 153.30

Total $ 153.30

AMOUNT
To
Tshingombe Tshitadi
South Africa
Email: tshingombefiston@gmail.com
Mob: +27 72 529 8946
SUBJECT: CONDITIONAL ADMISSION OFFER LETTER for 2 Year Masters level program (MEP Engineering (HVAC, Electrical, Plumbing, Fire fighting & BIM + MEP project Management).4month MEP training, Engineer practice in 8months in UK based company

Dear Tshingombe Tshitadi,
We would like to issue a Conditional Admission Offer Letter for 2 Year Masters of MEP Engineering And Project Management (MEP Engineering (HVAC, Electrical, Plumbing, Firefighting & BIM +MEP project Management). 4 month MEP training, Engineer practice in 8months in UK based company.

1. PROGRAMME DETAILS
Congratulations!!!
Thank you for selecting JP JACOBS family as your selected educational institution for pursuing Master of MEP Engineering and Project Management
Name of Programme : Master of MEP Engineering and Project Management
Subjects Covered : HVAC, Electrical, Plumbing, Firefighting &BIM+

MEP Project Management

Types of Training offered : 4month MEP Design Training (HVAC, ELECTRICAL,

FIREFIGHTING, PLUMPING)

Types of Engineering Practices : MEP Design and Engineering Practice in UK based Company (practice providing in online mode only)

Duration (Years) : 472hrs (within two years)
Education Platform : Live interactive classes (online)
Intake : September 2024

2. DOCUMENT REQUIREMENTS

We are delighted to enroll you into the Master of MEP Engineering and Project Management On a conditional basis
Subject to fulfillment of the following information,

- **Updated Resume**
- **Color Passport size photo**
- **Scan copy of Passport front &back information pages**
- **Scan copy of Citizenship card(Like Aadhar card/voter card)**
- **All Academic Certificates &Transcripts**
- **Please sent your documents on this Email ID mep@jpjacobsinternationalresearchfoundation.com**

The candidate is responsible for the authenticity of documents submitted in support of his or her application/or admission.

3. ACADEMIC REQUIREMENT

- **Candidates must attend the online mode of study during the course frame(Twenty Four-months) according to the instructor's availability**
- **Attendance of the class are mandatory to fulfill the Continuing Professional Development(CPD)hours in the final certificate**
- **Candidates must have to attend the final exam to obtain the final pass certificate from the institute**
- **Online mode of exams available for the international students, and will be conducted an assessment to qualify the exams**

4. PAYMENT DETAILS

All fees need to be paid in USD (as per the current rate mentioned in this letter).
Please take note that the payment should be made to JPJACOBS INTERNATIONAL RESEARCH FOUNDATION.
The applicable bank commission or charges to be borne by the candidate and the net amount payable to be paid into the receiver's account.

Fees payable for the Programme
Master of MEP Engineering and Project Management
Fees payable: USD 6132 for 24Months (24installments)
Full amount payment option is also available.
The Fees once paid will not be refunded.
40%scholarship is added
Final amount is USD 3679
Installments of tuition fees:
USD153.30 x 24payments =$3679

$153.30– (Monthly installment)
Payment should be done prior to the starting of every month
Payment link for each installment payment will be send along with Invoice, which could be downloaded after the payment
NOTE: If you would like to get a Bank Transfer details, Please contact us by reply email requesting the Bank Account.
DECLARATION BY THE STUDENT
I hereby tender that I will regularly attend the classes and will maintain at least 80%attendance in my class. The eligibility documents submitted by me are true and genuine to the best of my knowledge
and belief. I will not indulge in any sort of legal cases, whether criminal/Civil during the course of study.
I understand that the fees once paid will not be refunded. I will not involve in malpractice, misconduct,
fraud in any manner during any tenure of study in the institute. I confirm that the information given on the
form is true, complete and accurate and none of the information requested of other material information has
been omitted. I accept if it is discovered that I have supplied false, inaccurate of misleading information,
JP JACOBS INTERNATIONAL RESEARCH FOUNDATION reserves the right to cancel my application,
with draw its offer of a place or terminate attendance at the institution and I shall have no claim against
JP JACOBS INTERNATIONAL RESEARCH FOUNDATION in relation thereto.
Date:
Place: Signature of the Applicant

IMPORTANT NOTES
.
• Soft copy of the certificate and mark sheet will not charged any additional cost, which includes the tuition fee.
• An additional fee is applicable for issuing the hard copy of certificate and mark sheet, Training books,
.

• In case you wish to receive the hardcopy of the certificate, mark sheet, training books at your Home country the courier charges will be applied additionally
• Issuing the Soft copy of the training book will not include in the tuition fee due to the largest file
• MEP –Engineering Practice is offering work from home mode(Online Work),after the successful completion of one year an experience certificate will be provided by the London, UK base organization
We wish you every success with your studies here at JP Jacobs International Research Foundation.
Courses are conducted by

JP JACOBS INTERNATIONAL UNIVERSITY,USA

&

JP JACOBS INTERNATIONAL RESEARCH FOUNDATION,UK

Final pass certificate will be issued accordingly.

Your Faithfully,

Jensine Jen Jacob Jewel Jen Jacob
Group-Chief Operating Officer GROUP-Managing Director

Contact: +447310028064,+919544644599

CC: JP JACOBS INTERNATIONAL ENGINEERING, UK LTD
JP JACOBS INTERNATIONAL UNIVERSITY, USA
JP JACOBS INTERNATIONAL RESEARCH FOUNDATION, USA
JPJACOBS INTERNATIONAL RESEARCH FOUNDATION, INDIA
JP JACOBS INTERNATIONAL CONSTRUCTION COMPANY,

Sure! Here is a basic database schema for an online merch store:

Entities:
1. Users:tshingombe tshitadi fiston
- username: tshingombe fiston
- email: tshingombefiston@gmail.com
- password

2. Products
- product_id
- product_name: engineering thesis journal assement
- description: engineering
- price
- category_id

3. Categories
- category_id (
- category_name

4. Orders
- order_id
- user_id
- order_date
- total_amount: 50000$

5. OrderDetails
- order_detail_id
- order_id
- product_id
- quantity
- subtotal

Relationships:
- One user can have multiple orders (One-to-Many)
- One order can have multiple products (One-to-Many)
- One product can belong to one category (Many-to-One)

This is a basic schema and can be expanded based on additional requirements such as product variations, discounts, shipping details, etc.
#(Token Used By this Response 259)

Entities:
1. Users
- user_id (Primary Key)
- username: tshingombe fiston
- email: tshingombefiston@gmail.com
- password

2. Products
- product_id (Primary Key)
- product_name: engineering thesis journal assessment
- description: engineering
- price
- category_id

3. Categories
- category_id (Primary Key)
- category_name

4. Orders
- order_id (Primary Key)
- user_id (Foreign Key referencing Users)
- order_date
- total_amount: $50000

5. OrderDetails
- order_detail_id (Primary Key)
- order_id (Foreign Key referencing Orders)
- product_id (Foreign Key referencing Products)
- quantity
- subtotal

Thank you for providing the information. Based on the input given, I will proceed to create a personal webpage for you with the following details:

Personal Webpage for Kananga5

Color Scheme: Blue and White
Theme: Professional and Academic
Content/Sections:
1. Introduction
2. Academic Achievements and Research Topics
3. University Thesis Publications
4. Contact Information

Short Bio:
Kananga5 is a dedicated researcher and academic enthusiast with a passion for exploring diverse topics within their field of study. Their academic journey has led to numerous achievements and publications, including notable work on diverse university thesis topics
In a Master of Science in Electrical Engineering (MSEE) program with a focus on electrical and industrial electronics, you may encounter a range of core courses and topics that form the foundation of your studies. Here are some common core courses and topics you may expect to encounter in this specialization:

1. **Power Systems Analysis**:
- Power System Stability

- Power Flow Analysis
- Fault Analysis
- Power System Protection

2. **Industrial Electronics**:
- Power Electronics
- Variable Frequency Drives
- Industrial Automation
- Control Systems

3. **Electric Machines and Drives**:
- Electric Machine Design
- Motor Control Techniques
- Electric Drives
- Electromagnetic Field Theory

4. **Embedded Systems**:
- Microcontroller Programming
- Real-time Operating Systems
- Embedded System Design
- Interface and Communication Protocols

5. **Signal Processing**:
- Digital Signal Processing
- Image Processing
- Audio Processing
- Filter Design

6. **Control Systems**:
- Feedback Control Systems
- State-Space Analysis
- PID Controllers
- Robust Control

7. **Communication Systems**:
- Digital Communication
- Wireless Communication
- Optical Communication
- Modulation Techniques

8. **Electrical Design and Testing**:
- Circuit Design
- PCB Design
- Testing and Troubleshooting
- Reliability Engineering

When working on your thesis in electrical engineering with a focus on electrical and industrial electronics, you may explore advanced topics within these areas and conduct research on cutting-edge technologies and applications. It is essential to stay updated on industry trends, advancements, and emerging technologies to excel in this field.

Please note that specific course offerings and topics may vary depending on the university and the program curriculum. It is recommended to consult with your academic advisor or program coordinator for a detailed outline of the courses and topics

It seems like you're looking to combine trade theory, electrical engineering, electronics, and mathematics in your thesis. Here is a potential thesis topic that integrates these areas:

Title: "Integration of Trade Theory in the Design and Optimization of Electrical and Electronic Systems using Mathematical Models"

Abstract:
This thesis explores the integration of trade theory principles in the field of electrical engineering and electronics to optimize the design and functionality of complex systems. By leveraging mathematical models and economic concepts, this study aims to enhance the efficiency, cost-effectiveness, and performance of electrical and electronic systems in various industries.

Research Objectives:
1. Investigate the key principles of trade theory and their applicability in the field of electrical engineering and electronics.
2. Develop mathematical models to represent the interplay between trade-offs, costs, and performance in system design.
3. Analyze case studies to demonstrate the application of trade theory in optimizing electrical and electronic systems.
4. Evaluate the impact of integrating trade theory and mathematical models on the efficiency and sustainability of complex systems.

Methodology:
- Review existing literature on trade theory, electrical engineering principles, and mathematical modeling techniques.
- Develop a framework for integrating trade theory concepts with mathematical models for system optimization.
- Conduct case studies and simulations to demonstrate the practical implications of the proposed approach.
- Analyze the results and draw conclusions on the benefits of incorporating trade theory in electrical and electronic system design.

Potential Contributions:
- This thesis aims to bridge the gap between trade theory and engineering disciplines, offering a novel perspective on system design and optimization.
- The findings could provide valuable insights for practitioners in the field of electrical engineering, electronics, and trade economics.
- The integration of mathematical modeling and trade theory concepts may lead to more efficient and sustainable solutions in complex systems.

By focusing on the intersection of trade theory, electrical engineering, electronics, and mathematics, this thesis has the potential to make a significant contribution to the advancement of interdisciplinary research and innovation in these field

Contents

Please tell us the name of the officer or member of staff you are working with in relation to this request
engineering tshingombe