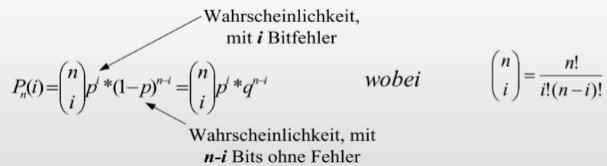
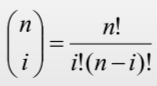
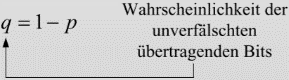
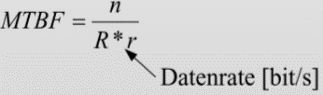
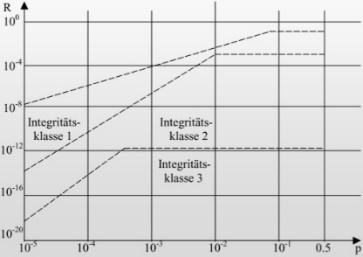
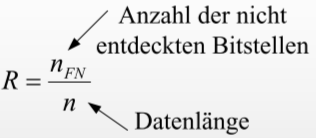
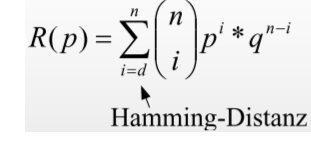
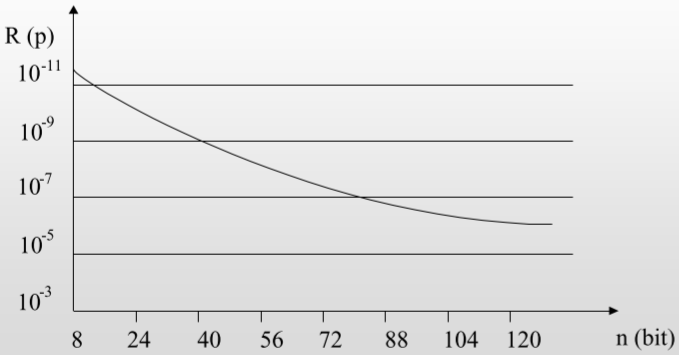
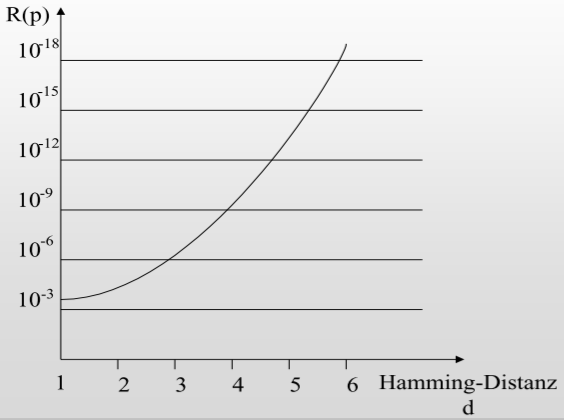
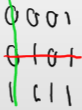
**Fehler in Betriebssysteme  
Bitfehlerrate**: Bitfehlerrate **pF** ist ein Mass für die Störempfindlichkeit des Übertragungskanals.

**Bitfehlerwahrscheinlichkeit p:**- ist die WSK in der ein Bit verfälscht wird.  
- Datenlänge **n** mit genau ***i*** Fehler  
- Fehlerwahrscheinlichkeit **Pn(i)**

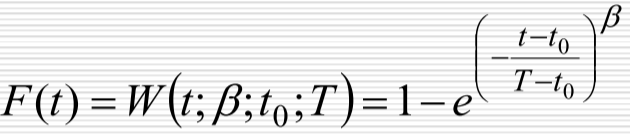
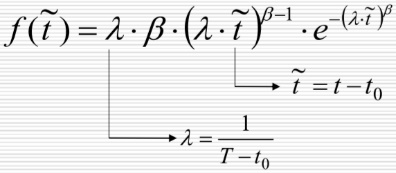
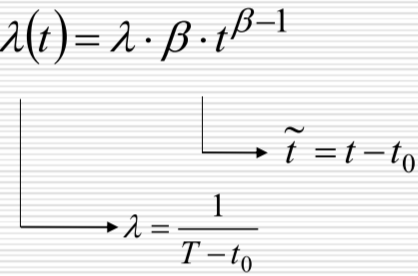
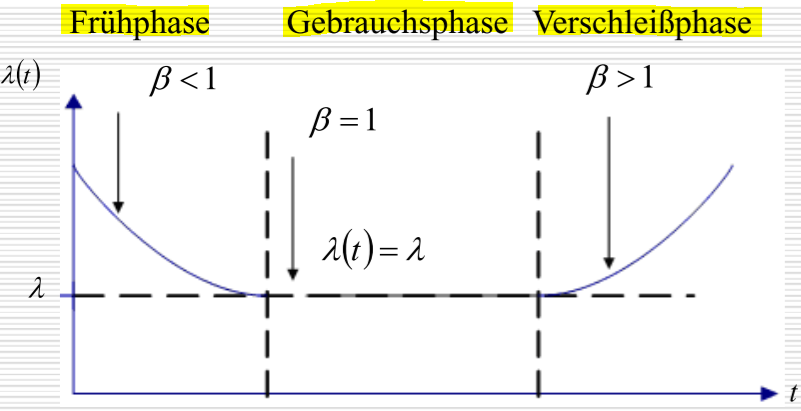
****

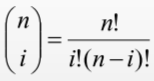
- Ist abhängig vom Übertragungsmedium  
  
**Restfehlerrate R:** bezieht sich auf die Anz nicht erkannt bleibenden Fehler.  
   
damit ist **MTBF** die mittlere Ausfallzeit der unbekannt bleibenden Fehler.  
**Datenintegrität**: Ist die Unveränderbarkeit von Infos & deren Informationsgehalt zwischen Sender. & Empfänger.  
**Datenintegritätsklassen(nach DIN19244, Gafik)   
Restfehlerwsk:** Die Summe der einzelnen Bitfehlerwsk. Für den Fall, dass keine Datenschutzmethoden implementiert sind wird **d** auf **1** gesetzt.  
   
**Restfehlerwsk R(p) in Abhängigkeit von: 1. Datenlänge n:** mit **p = 10-3** & **d**= 4  
  
**2.Hamming-Distanz d:** **n**=8bit & **p**=10-3  


**Fehlerarten:** **1.** Erkennbar & korrigierbar **2.** Erkennbar, nicht korrigierbar **3.** Nicht erkennbare Fehler 🡨 schlimm  
Einzelbit-, Bündel-, Synchronisat.fehler =Lambda=Fehler, **s**=secure Fehler,  
**D**=dangerous Fehler(<darunter folg.)  
**DD**=dang.detect. **DU**=dang.undetectable

**Fehlererkennung**: **1**. redundante Infos werden mitübertragen **2**. Übertragungsf. Werden beim Empfänger erkannt. **3**. Methode zu Fehlererkennung sind durch: Paritätsbit, Prüfsum., CRC, Haming-Code  
**1. Paritätsbit**: P. ist ein zeichendisk. Prüfbit, P. wird zu den Datenbit hinzugef. & macht die Bitsumme gerade oder unger., dient bei asynchroner Übertragung, einzelne Fehler können erkannt werden. **Vorteil:** Braucht nur ein Bit um Parität zu setzen.  
**Aber es merkt nicht, wenn 2bits ändern.**  
Arten:1. gerade/unger. Parität 2. Vertikale Parität(**VRC=Blockübertr.**) 3. Längsparität(**LRC**) 4. Matrixparität(BBC)  
**>** Das Paritätsbit wird so gewählt, dass die **Anz.1** gerade/unger.(LRC/VRC)sind.  
**>** **BBC**: LRC+VRC=BBC & Schnit.Pkt. LRC & VRC mit bit, dass VRC ger/ung. setzt.  
**2. Prüfsumme:**Datenblock von 8,16,32Bit Inhalte der Daten-Bytes addiert. Prüfsum wird an Datenblock angehängt & mitges.  
**+**: geht schnell, **-** nicht erk. von manipul. Daten. (vertau/wegla/veränd. von Bytes)   
**3. CRC (Cyclic Redundancy Check)**: Daten als Polynom dargest.: Grad Poly. = Anz.St. Daten – 1. Polynomst. Wird nur angegeben, wenn Datenbit an dieser Stelle = 1 ist, sonst nicht.  
Bsp. (8-Stellen)10011011=x7+ x4 +x3 +x1 +1  
**G(x)**=Generatorpolyn, **rG**=Az.Stellen G(x), **M(x)=U(x)**= Nutzdaten, **Anz.Nullst.**=**rG**-1 oder **Potenz** von **G(x)**, Ergebnis=**T(x)**= **M(x)+Rest von Polynomdivision**, **XOR**: 0+0=0;0+1/1+0=1,**1+1=0**Senden:1. **rG**-1 an **M(x)** anhängen, 2. Teile **M(x)+Anz.0** durch **G(x)** mit XOR. 3. Nimm **Rest**(Polynomdiv.) füge vorne soviele 0 an, wie nötig um **rG**-1 Stellen für CRC zu bekommen **M(x) &** send. D.h. Rest = 1111, dann CRC=00001111 oder in hex. 0x0f (für rG=9)  
Empfangen: 1.**T(x)** nehmen & durch **G(x)** teilen. If Rest = 0, dann ok, sonst Fehler.  
**CRC erkennt folgende Fehler:** 1. sämtliche Einzelbitf. 2. sämt. Doppelbitf., wenn (xk+1) nicht durch das Prüfpol. teilbar ist, k<= Rahmenlänge 3. Sämt. Fehl. Bei ung.Anz. wenn (x+1) ein Faktor des Polynoms ist 4. sämt. Fehlerbursts d. Länge <= Grad des Polynoms.  
**Internat. Genormte Prüfpolynome:** **CRC-8**=x**8**+1, **CRC-16**= x**16**+ x15+ x2+1, **CRC-32**= x**32**+^25^23^22^16^12^11^10^8^7^5^4^2+x+1, HEC= x8+ x2+x+1, CCITT= x16+ x12+ x5+1

Drei Phasen: Früh, Gebrauchs- & Verschleissphase  
**Badewannenkurve/ nicht Weibull**

**Verteilungen für Hard-& Software-Zuverlässigkeitsprognosen**Diskrete Verteilung: Binomial-V.&Poisson-V.  
Kontinuierl. V.: Weibull-V & Exponential-V  
**Weibull-Verteilung:  
Daten**: 1. Anzahl der Komponenten 2. Ausfallzeitpunkt der Komponenten  
Ausfallwskfkt(zur berechnung von MTTF)**:**   
**t=**Zeit, **β**=Formparam., **t0**=Korrekturpar. der Zeit, **T**=Lageparam. der charakteristis. Lebensdauer  
Ausfalldichtefunktion f(~t)/f(x):  
Ausfallrate:  
   
Zeitabhängige Ausfallrate  **(t)** **Praxis:** **Früh**: Produktionsfehler (Lötstellen die kaputt gehen), nimmt zufällig ab. **Gebrauchs-=Zufallsausfälle**: kein Verschleiss, Ausfallrate konstant, Bsp. in der Regel Halbleiterbauelemente, Totalschaden Auto nur aufgrund Unfall(Zufall). **Spät**: Anstieg bei zunehmender Lebensdauer, zuhneh. Alterungsverscheiss, Ausfallrate steigt wieder, Bsp. verderben Lebensmittel, verschl. Metallteile, **Exponential-Verteilung:** Theoretische Bedeutung da kein konstant. Ausfallrate,  
Früh-,&Ermüdungsausf. Nicht durch exp. Verteilung abbildbar.

**Begriffe:  
Zuverlässigkeit=** Unter Zuverlässigkeit wird der Teil der Qualität verstanden, der das Verhalten der Einheiten während einer vorgegebenen Zeitspanne, bei vorgegebenen Anwendungsbed., betrifft. **Lebensdauer(TTF, T(Lageparameter))**= Die Lebensd. eines Produkts t ist die Zeit bis zum Ausfall (**failure**). Mit Zeit können konkrete Zeiteinheiten (h, month, years) gemeint sein, die Zeit kann aber auch in Lastwechseln, Schaltvorgänge, Laufleistung usw. angegeben werden. **Ausfall** kann in zwei Arten auftreten, dem: 1. **Sprungausfall**= plötzlicher Übergang in den funktionsunfähigen Zustand (Reifen platzt ...) 2. **Driftausfall**= definiertes Ende z.B. Reifen erreicht 1,6mm Profiltiefe. **Mittlere Reparaturdauer(MTTR)**=mean time to repair **Ausfallsteilheit**= **b** (Formparameter)   
**Zuverlässigkeitsfunktion=MTTF** (Mean time to failure) **Ausfallrate=MTBF**(Mean time between Failures) gilt nur bei **reparablen** Systemen. **MDT**(Mean time down)

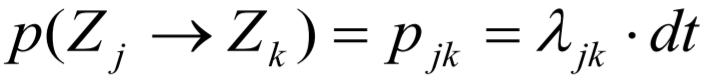
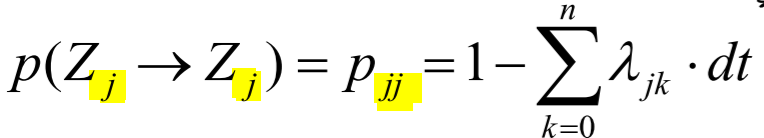
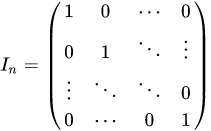
**Gaussverteilung**

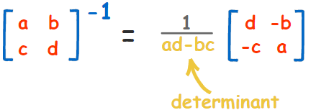
**Ausfallwahrscheinlichkeit**: Die WSK f(t) einer Komp. des Bestandes zum Zeitpkt. T<ti auszufallen.  
**Ausfallrate:**ist eine Kenngröße für die Zuverlässigkeit eines Objektes. Gibt an, wie viele Obj. in einer Zeitspanne durchschn. ausfallen. Angegeben in **1/Zeit**, also Ausfall/Zeitspanne. Ist die Ausfallrate zeitlich konstant, wird sie üblicherweise mit der Konstante **λ** bezeichnet, ist der Kehrwert die mittlere Lebensdauer **MTTF**, bei reparablen Obj. die mittlere Zeit zw. zwei Ausfällen **MTBF**. Eine spezielle Einheit für die Ausfallrate ist **FIT** Failure In Time mit der Einheit „**Ausfälle pro 109 Stunden**“, also pro Milliarden Stunden. Die Ausfallrate kann nicht an einem einzelnen Objekt gemessen werden. Sie wird aus Beobachtungen an einer größeren Anzahl gleicher Obj. geschätzt. Die Ausfallrate zu einer bestimmten Zeit ist dann gegeben durch die **Anzahl** **Obj.**, die in einem **bestimmten** **Zeitintervall** (z. B. einen Tag) ausfallen, **dividiert** **durch** die **Anzahl** **guter** **Obj.** **am** **Anfang** **des** **Zeitintervalls**(also vor Beginn d. Tags der Messung(Bsp.Tag 10). In der Praxis wird die Ausfallrate als eine Abschätzung definiert. **Summary:** Die Ausfallrate sagt uns wie lange ein Produkt im Durchschnitt lebt & die Ausfallwsk sagt uns wie hoch die WSK dieses Produktes nach einer best. Lebensdauer ist, um auszufallen. (Je länger, je höher)

**Ausfallswskdichte:** Die WSK eines Ausfall im Zeitintervall Δt. **Ausfallhäufigkeit**: Die Differenz der relativen Bestände am Anfang & Ende des betrachteten Zeitintervalls. **Ausfallhäufigkeitsdichte**: Das Ergebnis von Ausfallhäufigkeit durch betrachteten Zeitintervall.

**SIL(Zert):** Sicherheitsintegrations-level oder Sicherheitsstufe oder Sicherheits-Integritätslevel. Sie dient zur Beurteilung elektr. Systeme in Bezug auf die Zuverlässigkeit von Sicherheitsfunktionen. Aus dem angestrebten Level ergeben sich die sicherheitsgerichteten Konstruktionsprinzipien, die eingehalten werden müssen, damit das Risiko einer Fehlfunktion minimiert werden kann. SIL1/2=Verfügbarkeit, SIL ¾=Safety&Security **Gravity(downtime)** SIL1-4: Minor, Severe, Major, Catastropic(nuclear)

M=I-Q 🡪 Wert von jeder Pos. von I-Matrix – Wert von derselben Pos. Q-Matr.  
dt weglassen **🡪 M**

**Markov-Modell:**Ein Mrkv-Modell ist Gedächtnislos, d.h. Übergangswsk hängt nur vom aktuellen Zustand ab & nicht von d. gesam. Vergangenheit. **Zustand**=Kreis, **Übergangs-(µ) /Ausfallsrate**(**)**=Pfeil. Das Mkv-Modell kann zwei Sys. analysieren: 1. **Nicht-reparable** Sys. (Sys. Muss ausgetauscht werden) 2. **Reparable** **Sys**.(kann wieder in fehlerfreien Zust. Gehen)  
**Übergangswsk:** **jk** =Übergangsrate (>=0, sonst Übergang nicht möglich),**Z**=Zustand, **dt**=Zeitintervall, Zj=Z0  
  
**Recoverywsk**(zurück z. gleichen Zustand, auch **Beharrungszustand**)  
  
d.h. Recov.wsk.=1-Ausfall-/Übergangwsk.  
daraus folgt:  
**Berechnung MTTF:**(Durchschnittszeit zwischen zwei Ausfällen).  
**Schritte: 1. P-Matrix bilden 2. M-Matrix bilden 3. N-Matrix bilden 4. MTTF-Wert bilden  
P-Matrix=Übergangsmatrix**:  
**1.** Zustandsübergänge beliebiger Architekturen des Mrkv-Modells berechnen. **2.** Falls keine direkte Verb. zwi. 2 Zustände: Wert=0 **3.** Überprüfen: Alle Parameter pro Zeile aufsummieren=1 **4.** P-Matrix stellt Anfangszustand des Systems dar. **5.** Anz. Zeilen/Spalten=Anz. Zustände **6.** 1. Zeile erstes Feld=P00, 2 Feld=P01, 3.Feld=P02, … 2.Zeile 1.Feld=P10, 2.Feld=P11, 3.Feld=P12.. **7.** Beharr.Zustand=P00= 1- alle anderen Parameter dieser Zeile. **8. Multiplizieren mit *dt* nicht vergessen bei jedem Übergang, egal ob !!**  
**Q-Matrix=Zuverlässigkeitsmatrix**:  
**1.** Sys. Muss in Betrieb sein. (Z0=ok)   
**2.** Keine absorbierenden Zustände(Keine Fehlerraten-Übergänge gehen von diesem Zust. weg, d.h. keine weggehende , sondern nur wegg. **µ (Übergange zu Sicherem Zustand oder OK Zustand, beide grün)**.  
**3.** Ausschliessen von Zust. mit sicheren Fehlern (oftmals Sys. S) & Zustände mit gefährlichen unentdeckte Fehlern(Sys. DU, rot), d.h. Kreuz machen über alle Zeilen & Reihen. Alles was mit diesem Zust. zu tun hat muss weg 🡪Sicher Zust. weg, da DU bestimmt werden muss & Zustand mit DU weg, da schon zu spät & keine Reaktion möglich.  
**M-Matrix:  
1.** I-Matrix(Einheitsmatrix) – Q-Matrix rechnen. 🡪 Einheitsmatrix= quadratische Matrix, deren Hauptdiagonale nur aus **1er** besteht. Alle anderen Elemente bestehen aus Nullen. **Beispiel:  
**

**N-Matrix:  
Inverse von M:**Im Fall von einer 1x1 Matrix wäre es 1/M.  
2x2 ist komplizierter:

**1.** Position a&d müssen getauscht werden **2.** b&c negieren **3.** Alle Werte einzeln durch die Determinante(ab-bc) teilen.  
**MTTF:**MTTF-Wert = Elemente der ersten Zeile der N-Matrix addiert.  
Im Fall einer 2x2 N-Matrix wäre das:  
Im Fall einer 1x1-M-&N-Matrix, ist der MTTF gleich der N-Matrix, also 1/M  
🡪 **MTTF=Resultat in Stunden**

**Echtzeitsysteme(EZS):**Echtzeitbetrieb ist ein Betrieb eines Rechensys., bei dem Programme zur Verarbeitung anfallender Daten **ständig betriebsbereit** sind & zwar so, dass sie **innerhalb einer vorgegebenen Zeitspanne verfügbar** sind.

**ESZ** sind immer **abhängig** von der vorgegebenen **Zeitbedingungen**. Von **Echtzeitsystemen** spricht man, wenn ein Sys. ein Ergebnis innerhalb eines vorher fest definierten Zeitintervalles garantiert berechnet. **Echtzeit** bedeutet nicht unbedingt schnell. (Ist oftmals, muss aber nicht). Man ist nicht mehr nur für sein "**Item**" verantwortlich, sondern es muss gewährleistet werden, dass dieses Item mit einem anderen/mehreren Items fehlerfrei funktioniert. D.h. Jeder Item-Owner muss seine eigenen Funktionstests ihres Items durchführen vom Endprodukt.D.h. sowohl die Korrektheit, als auch die Dauer der Ermittlung der Ergebnisse wird garantiert. **Echtzeitsysteme haben/sind:** **1.** Hohe Arbeitsgeschw. **2.** Fehlertoleranz **3.** Kosteneffizient **4.** Niedriger Energiebedarf **5.** Hohe Ressourcenauslastung(CPU,RAM) **6.** Im Einsatzfall können auch Kompromisse nötig sein.  
Arten von EZS: **1.** **Hartes EZS**: Verletzung einer Deadline hat katastrophale Konsequenzen für das Sys. Bzw. die Umwelt **2. Weiches EZS:** Verl. der Deadline führt zum Absinken des Wertes des Ergebnisses, Einfluss auf die «Güte» des Dienstes, Sys.leistung sinkt mit Anz. Verl. Deadlines. **3.** **Mittelhartes ESZ**: Nach Überschreiten der Deadline ist das Ergebnis wertlos, jedoch keine Katastrophe (firm deadline) **Emmbedded Systems:** Ein in sich unabhäng. Sys., welches in ein grösseres Sys. Eingebettet wird. **Operation System Säulen:** **1.** Aktuator(Input) **2.**Verarbeitung(Logik) **3.**Ausgang(Output). Begriffe: **Job**: Programmeinheit, die geplant & ausgef. wird. **Task**: Mehrere zusammengeh. Jobs. **Deadline:** Zeitpkt, bis zu dem der Job/Task ausgef. sein muss. **Jitter:** (unerwünschte) Schwanken der Ausführungszeiten des Jobs/Task. (Bsp. falsch Interpretation von Sensoren, welche den Vibrator am Steuerrad ausführen & deshalb den Fahrer erschrecken oder andere Reaktion hervorrufen. **Schedule**: Plan, der die Abarbeitungsreihenfolge von Aktivitäten festlegt.