(5) Chapter (5)
Software Reliability

Failure Modes:

Mode of failure:

General Reason for a failure are– poor quality of fabrication, design error, overload of component, weare-out. etc.

☐ **Hardware failure:** It refers to a system failure traceable to some component malfunction.

☐ **Software failure:** It refers to a system failure traceable to some error in the software.

5.3. Reliability Theory:

- Derive the equation: $R(t) = e^{-\lambda t}$ where $R(t)$ is software reliability and λ is a failure rate (constant)
- Define Reliability in terms of failure rate (Hazard) function i.e. derive the equation, $R(t) = e^{-\lambda t}$

Let, t be a random variable representing the time to failure and probability that the time to failure t is in some interval $(t_1, t_1+\Delta t)$ be $P(t_1 \leq t \leq t_1+\Delta t)$.

Now relating to the density and distribution function, we get

$$P(t_1 \leq t \leq t_1 + \Delta t) = f(t_1)\Delta t = F(t_1+\Delta t) - F(t) \quad -(1)$$

where,

$f(t_1)$ = value of probability distribution function at point t_1

$F(t_1)$ = value of cumulative probability distribution function at point t_1.

Dividing ① by Δt and letting $\Delta t \to 0$, we get,

$$f(t) = \frac{dF(t)}{dt} \quad \text{——②}$$

From ② we get

$$F(t) = \int_0^t f(x)\,dx \quad \text{——③}$$

Now, probability that the time to failure occurs in interval $0 \leq t \leq t_1$ is $P_f(t_1) = P(0 \leq t \leq t_1) = F(t_1) - F(0)$ ——④

where random variable t is only defined over interval 0 to ∞

As a consequence, $F(0)$ becomes zero. Then,

$$P_f(t) = F(t) = \int_0^t f(x)\,dx \quad \text{——⑤}$$

Let, $P_s(t_1)$ is is probability to failure where time of failure is langer than t_1, that is, $t > t_1$. Then, from fundamental laws of probability, we get,

$$P_s(t) + P_f(t) = 1$$

Therefore, probability of success or reliability is,

$$R(t) = 1 - P_f(t) = 1 - \int_0^t f(x)\,dx \quad \text{——⑥}$$

This reliability can be defined in terms of failure rate function where failure rate function or Hazard function $z(t)$ is defined in terms of probability $P_f(t)$ that a failure occurs in some interval t_1 to $t_1 + \Delta t$, given that the system survived up to time t. Then,

$$P(t_1 \leq t \leq t_1 + \Delta t \mid t > t_1) = z(t_1)\Delta t \quad \text{—— (7)}$$

This conditional probability can be shown as,

$$P(t_1 \leq t \leq t_1 + \Delta t \mid t > t_1) = \frac{f(t_1)\Delta t}{R(t_1)} = z(t_1)\Delta t \quad \text{—— (8)}$$

combining equation (2) and (8) we get,

$$\frac{dF(t)}{dt} \cdot \frac{\Delta t}{R(t_1)} = z\, z(t)\Delta t$$

$$\text{or, } \frac{1}{R(t)} \cdot \frac{dF(t)}{dt} = z(t) \quad \text{—— (9)}$$

Now, from equation (6) we get,

$$\frac{dF(t)}{dt} = -\frac{dR(t)}{dt} \quad \text{—— (10)}$$

From (9) and (10) we get,

$$\frac{1}{R(t)} \cdot \left(-\frac{dR(t)}{dt}\right) = z(t)$$

$$\text{or, } \frac{dR(t)}{R(t)} = -z(t)\,dt \quad \text{—— (11)}$$

Integrating both sides with respect to t, we get,

$$\ln R(t) = -\int_0^t z(x)\,dx + C \quad \text{—— (12)}$$

When $t = 0$, system is initially good and $R(0) = 1$ (initial condition). from (12) exponenting both sides we get,

$$R(t) = \exp\left[-\int_0^t z(x)\,dx + C\right] \quad \text{—— (13)}$$

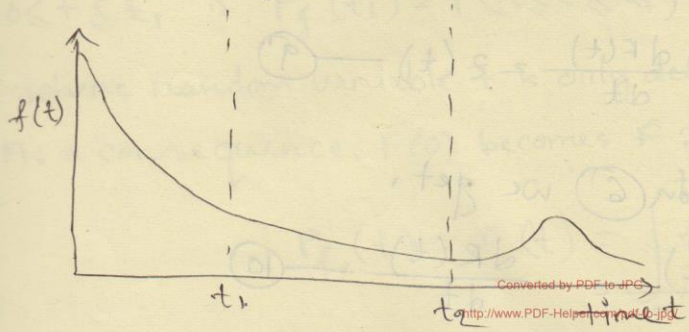
If the failure rate is constant, then, $z(t) = \lambda$, then,

$$R(t) = e^{-\lambda t}$$

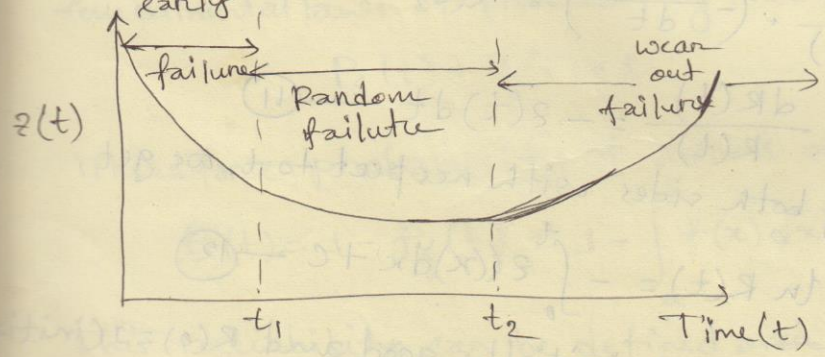
similarly if the failure rate is linearly increasing then, $z(t) = kt$. So,

$$R(t) = e^{-kt^2/2}$$

⊘ Draw the general form of failure curve for hazard rate.



(a) failure density



(b) hazard function.

⊘ What do you mean by software reliability and software availability?

☐ Software Reliability:

Software reliability is the probability that the program performs successfully ~~at~~ according to the specification for a given period of time.

☐ Software Availability:

Software availability is the probability that the program is performing successfully, according to the specification at a given point of time.

☐ Important difference:

reliability means no failure in the interval 0 to t, whereas availability means only that the system is up at time t:

⊘ What do you mean by old and new errror?

☐ Old errror: (Previously fixed)

Error which recurs in substantially the same form after the programmer has terminated the work on a code change believing that the error has been corrected.

∎ Generated (New) error:

One which does not exist until it is created as by produ
of a code change made to correct another error.

◎ Documentation:

It is an all-encompassing term which includes many
things in a computer program.

◎ Why documentation is necessary?

If a program is going to be used for only 1 or 2 years
without any changes, then if the code is relatively error
free, then the code is sufficient.

But if the program is to be changed several times
over 10-years period, then documentation is more important
than the code. We can always recode, and retest, especiall
if we have the complete test plan.

◎ For the maintenance cost models derive the equation
$D = (1-x)^k$ where D represents development fonce,
x represents the fraction of fonce left behind for
maintenance each time and k is the no. of projects.

Assume that,

P is programming work force,

M is maintenance force

D is the development force to handle new projects.

Then,

$$F = M(t) + D(t)$$

At $t=0$, $F = D = 1$ [$\because M = 0$]

Given, X be the fraction of development force left behind for maintenance at the end of a project. Then,

$$X = \frac{M}{F} \text{ and at } t=0, X \text{ is not defined}$$

After release of first project, $t = t_1$, then,

$$M(t_1) = X D(t_1) = X \times 1 = X \quad \text{---(1)}$$

$$D = 1 - M = 1 - X \quad \text{---(2)}$$

After release of second project, $t = t_2$, then,

$$M = X + XD = X + X(1-X) \quad \text{---(3)}$$

and $D = 1 - [X + X(1-X)]$

$$= 1 - X - X + X^2 = 1 - 2X + X^2 = (1-X)^2 \quad \text{---(4)}$$

Generalizing equation (1) to (4) yields, for k^{th} project release, $M = 1 - (1-X)^k$

$$D = (1-X)^k$$