Chapter 1

Teorija zanesljivosti

1.1 Uvod

V pričujočem poglavju bomo skušali razložiti in formalizirati pomembne osnovne pojme področa zanesljivosti. Začnimo kar z dvema najpogostejšima definicijama zanesljivosti.

Definition 1 *Po* [*Con*1] *je zanesljiv tisti sistem, ki počne natanko tisto kar hoče uporabnik (kupec) in to natanko takrat, ko se to od sistema zahteva.*

Definition 2 *Po* [*Els*1], [*Xie*1] *je zanesljivost verjetnost, da bo sistem vršil predvideno funkcijo v vnaprej podanem časovnem intervalu in vnaprej podanih delovnih pogojih brez odpovedi.*

Prva definicija je *uporabniško orientirana* in deloma sovpada s kvaliteto sistema, saj uporabnik pogosto enači zanesljivost delovanja s pojmom kvalitete. Kakršnakoli odpoved sistema uporabniku znižuje zaupanje do sistema in s tem posredno tudi zaupanje do proizvajalca/prodajalca sistema.

Druga definicija je *snovalsko orientirana* in meri zanesljivost v obliki verjetnosti nepojavitve odpovedi v opazovanem časovnem intervalu. Pogojena je z *intenzivnostjo odpovedovanja*, ki jo označujemo z $\lambda(t)$, merimo pa s številom odpovedi v končnem časovnem intervalu. Tipični rang velikosti za $\lambda(t)$ je na primer 1 fatalna odpoved na 10^7 prevoženih ur v civilnem letalskem prometu ali 1 odpoved na 10^9 delovnih ur v primeru posameznega integriranega vezja. Zanesljivost kot verjetnost delovanja v intervalu $[0, t]$ po [*Xie*1] lahko zapišemo z izrazom

$$R(t) = P(T > t), t \geq 0, \tag{1.1}$$

pri čemer je T točka odpovedi. Verjetnost odpovedi v omenjenem intervalu ali nezanesljivost zapišemo kot

$$F(t) = 1 - R(t) = P(T \leq t). \tag{1.2}$$

Zanesljivost $R(t)$ se izraža kot integral gostote verjetnosti časa odpovedi po izrazu

$$R(t) = \int_t^\infty f(x)dx. \tag{1.3}$$

Ena od osnovnih postavk, ki nas zanima v sistemih je pričakovani čas do odpovedi ali *pričakovana življenska doba*. Slednji pojem imenujemo za MTTF (angl. *mean time to failure*) v nepopravljivih in MTBF (angl. *mean time between failures*) v popravljivih sistemih. Obe postavki sta izračunljivi po izrazu

$$MTTF = MTBF = \int_0^\infty R(t)dt = \int_0^\infty t * f(t)dt.$$

Tretja kratica, ki jo pogosto uporabljamo, je MTTR (angl. *mean time to repair*), predstavlja pa potrebni čas za servisiranje ali menjavo komponente.

Pomemben pojem soroden zanesljivosti je tudi *dosegljivost (*angl. *availability),* ki nam za definirani računalniški sistem in opazovani časovni interval opazovanja podaja predviden delež časa, v katerem bo sistem na razpolago za normalno servisiranje zahtev.

1.2 Redundanca kot temeljna metoda izboljševanja zanesljivosti

Osnovna metoda za doseganje željene zanesljivosti je vpeljava *redundantnih* (odvečnih) komponent. Ob odpovedi posamezne komponente, bodisi zaradi napake ali izteka življenske dobe, tako funkcijo strežne enote prevzame redundantna komponenta. Z logičnega vidika je vezava komponent paralelna. Z vidika delovanja sistema kot celote ni nujno delovanje vseh redundantnih komponent. Ponavadi je dovolj, da sistemsko breme prevzema le ena od njih. Tako ločujemo več možnih stanj posamezne komponente v redundančni vezavi:

- *aktivno delujoče stanje*, pri čemer komponenta z vidika sistema opravlja svojo funkcijo,
- *aktivno nedelujoče stanje*, pri čemer z vidika sistema komponenta ne opravlja svoje funkcije, jo je pa hipno sposobna prevzeti v kateremkoli trenutku ob odpovedi predhodno aktivne delujoče komponente, ki je do tedaj opravljala svojo funkcijo (angl. *hot standby*),
- *pasivno nedelujoče stanje*, pri čemer je potrebno komponento pred opravljanjem svoje funkcije najprej spraviti v delujoče stanje (angl. *cold standby*), za kar je potreben zagonski čas in
- *stanje v odpovedi*, pri čemer komponente brez servisiranja ali zamenjave ne moremo spraviti v delujoče stanje.

Sisteme, v katerih imamo vgrajenih več entitet, kot bi jih potrebovali za normalno obratovanje tako imenujemo za *redundantne sisteme*. Ob kakršnikoli uporabi redundantnih komponent je potrebno zagotoviti tudi mehanizem vklapljanja komponent v "cold stanby" stanju in tako v primeru "cold stanby", kot tudi v primeru "hot standby" konfiguracije izvedbo preklopa poti, po kateri naj bi potovalo breme (preusmeritev bremena na delujoče komponente). V obeh primerih za opravilo poskrbi hipotetična naprava, ki jo bomo poimenovali *stikalo* (angl. *switch*). Tudi slednje ni idealno, tako da moramo računati tudi z njegovo ne/zanesljivostjo, možnostjo odpovedi in življensko dobo.

Slaba plat "cold standby" konfiguracije je počasnost zagona novo vpeljane komponente in s tem tudi potencialen začasni izpad procesnih zmožnosti. Dobra plat "cold standby" konfiguracije je v tem, da se neaktivne komponente (komponente v latenci) ne "obrabljajo". Povedano drugače se njihova življenska doba zaradi pasivnosti ne dekrementira vse dotlej, dokler niso aktivirane. Dobra plat "hot standby" konfiguracije je tako v hitrosti preklopa, slaba pa v hitremu iztekanju življenske dobe več komponent hkrati. Seveda slednje velja predvsem za materialne relaizacije (npr. strojno opremo), ne pa za programsko opremo.

1.3 Napaka, vgrajena hiba, odpoved

Po [*Pet*1] večina zanesljivostnih problemov na področju programske in strojne opreme izhaja iz trojice pojmov *napake* (angl. *error*), *vgrajene hibe* (angl. *fault*) in *odpovedi* (angl. *failure*). Združenje IEEE pomene omenjene trojice klasificira na naslednji način:

- za *napako* se smatra predvsem napako v razmišljanju ali specifikaciji, snovalsko napačno razumevanje problema ali uporabljene metodologije,
- za *vgrajeno hibo* se smatra napaka, ki jo implementiramo v programsko ali strojno opremo,
- za *odpoved* smatramo kakršnokoli nenačrtovano delovanje sistema kot celote, ki je rezultat vgrajene hibe.

Zaradi lažjega razumevanja in širšega pogleda na delovanje sistemov, bomo v nadaljevanju govorili le o pojmu napake, ki bo pokrival prva dva termina in o pojmu odpovedi.

Poljuben računalniški sistem je sestavljen iz dveh osnovnih sklopov in sicer iz *aparaturne* (strojne) in *programske* opreme. Pogostnost ali *intenzivnost odpovedanja* računalniškega sistema, ki neposredno vpliva na sistemsko zanesljivost, je pogojena z naslednjimi dejavniki:

- z intenzivnostjo odpovedovanja programske opreme,
- z intenzivnostjo odpovedovanja aparaturne opreme,
- z intenzivnostjo nepravilnega načina rokovanja uporabnika s sistemom in

- z intenzivnostjo porajanja ostalih zunanjih vplivnih dejavnikov (temperatura, vlaga, strela, itd.).

Nezanesljivost programske opreme izhaja iz napak, ki so bile vnešene v fazi razvoja. Odkrijemo jih lahko v *fazi testiranja*, ali pa šele kasneje v *fazi eksploatacije*. V slednjem primeru je odpravljanje napak neprimerno dražje.

Nezanesljivost aparaturne opreme izhaja iz odpovedi posameznih elektronskih komponent, kar vodi do odpovedi, nepravilnega ali pa degradiranega delovanja aparaturne opreme kot celote. Z razliko od programske opreme pri aparaturni v idealnem primeru predpostavljamo, da vsi sistemi zapuščajo fazo testiranja brez napak ali okvar. Do okvare in s tem odpovedi pride šele v *fazi eksploatacije* in sicer bodisi pod vplivom zunanjih dejavnikov, dotrajanosti materialov ali slabše realizacije posameznih komponent.

1.4 Možna stanja opazovanega sistema z vidika zanesljivosti

Z vidika zanesljivosti opazovanega sistema lahko njegova stanja razdelimo na naslednje skupine:

- sistem kot celota deluje normalno,
- zaradi napake sistem kot celota ne deluje (sistem ne deluje in je *v odpovedi*),
- zaradi napake sistem kot celota deluje nepravilno (sistem deluje nepravilno in je *v odpovedi*) in
- zaradi napake odpove le del sistema, katerega funkcije prevzame preostali še delujoči del sistema (sistem z *degradiranim delovanjem*).

Z vidika uporabnika je v drugem in tretjem primeru sistem neuporaben, v zadnjem primeru pa je sistem še uporaben, zmanjšajo pa se njegove performanse.

1.5 Faze življenjske dobe računalniškega sistema

Življensko dobo posameznega računalniškega sistema in njegovih posameznih sestavnih delov v splošnem razdelimo na tri faze:

- *faza otroštva*: v tej fazi intenzivnost odpovedovanja $\lambda(t)$ v času hitro upada zaradi testiranja, ki iz populacije eliminira okvarjene entitete (če niso popravljive), ali pa se omenjene entitete popravljajo (podsklopi in programska oprema),
- *faza eskploatacije*: v tej fazi imamo opravka s konstantno ali večinoma počasi (npr. linearno) rastočo intenzivnostjo odpovedovanja $\lambda(t)$ (tipičen primer konstantne $\lambda(t)$ najdemo pri elektronskih komponentah, kot so integrirana vezja, tipičen primer linearne rasti pa pri obrabljivih produktih, kot so npr. avtomobilske gume),

- *faza starosti*: v tej fazi začne intenzivnost odpovedovanja $\lambda(t)$ zelo hitro rasti predvsem pri strojnih komponentah, kjer prihaja do dotrajanosti materialov.

Vse tri faze so prikazane na sliki 1.1, pri čemer je v fazi eksploatacije linearna rast $\lambda(t)$ narisana s prekinjeno črto.

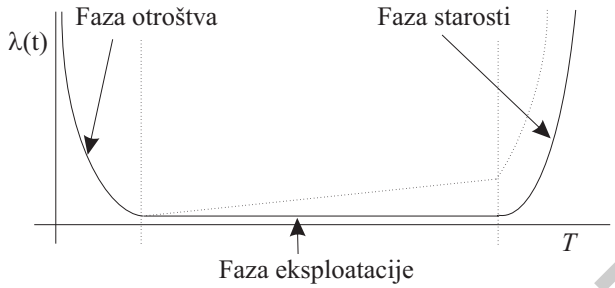

Figure 1.1: Različne življenske dobe sestavnih delov računalniških sistemov.

1.6 Modeli intenzivnosti odpovedovanja in pričakovane življenske dobe

Že v uvodu pričujočega poglavja smo se seznanili s funkcijo intenzivnosti odpovedovanja $\lambda(t)$. Z vidika uporabnika je najzanimivejša intenzivnost odpovedovanja v eksploatacijski dobi. Relacijo med $\lambda(t)$, $f(t)$ in $R(t)$ zapišemo z izrazom

$$\lambda(t) = \frac{f(t)}{R(t)}.$$

Različne vrste $\lambda(t)$ v omenjeni dobi si bomo ogledali v naslednjih razdelkih.

1.6.1 Konstantna intenzivnost odpovedovanja v fazi eksploatacije

Konstantna $\lambda(t)$ je v zrelostni ali eksploatacijski dobi tipična predvsem za elektronske komponente kot so tranzistorji, upori, kondenzatorji in integrirana vezja. Za slednje po [*Els*1] velja, da traja faza otroštva približno 10^4 delovnih ur (približno eno delovno leto). V fazi testiranja posameznih komponent se ta doba umetno zmanjšuje s slabšanjem delovnih pogojev (angl. *burn-in procedures, accelerated testing*), ki nadomeščajo relativno dolgotrajno izpostavljanje običajnim pogojem. Funkcija intenzivnosti odpovedovanja (v literaturi se ponekod imenuje tudi za funkcijo hazarda) se tako v eksploatacijski dobi izraža z izrazom

$$\lambda(t) = \lambda, \tag{1.4}$$

pri čemer je λ konstanta. Odtod sledi, da je funkcija gostote verjetnosti časa odpovedi

$$f(t) = \lambda e^{-\lambda t}, \tag{1.5}$$

funkciji zanesljivosti in odpovedovanja pa po vrsti

$$R(t) = e^{-\lambda t}, \tag{1.6}$$

$$F(t) = 1 - e^{-\lambda t}. \tag{1.7}$$

Po izrazu

$$MTBF = MTTF = \int_0^\infty R(t)dt = \int_0^\infty e^{-\lambda t}dt = \frac{1}{\lambda}, \tag{1.8}$$

izpeljemo pričakovano vrednost funkcije $f(t)$. Interpretiramo jo kot pričakovani čas za popravilo (seveda pri popravljivih sistemih), izraža pa se kot

$$MTTR = \frac{1}{\mu}, \tag{1.9}$$

pri čemer μ predstavlja intenzivnost servisiranja ali inverzno vrednost časa servisiranja.

Case 3 *Proizvajalec vrši OLT (angl. operational life test) teste na keramičnih kondenzatorjih in pri testiranju ugotovi, da je intenzivnost odpovedovanja $\lambda(t)$ konstantna in sicer $3 * 10^{-8}$ odpovedi na delovno uro. Kakšna je zanesljivost kondenzatorja po 10^4 delovnih urah in kolikšno je pričakovano število odpovedi po 5.000 delovnih urah na seriji velikosti 2.000 kosov?*

$$\lambda(t) = 3 * 10^{-8} \rightarrow R(10^4) = e^{-3*10^{-8}*10^4} = 0.99970. \tag{1.10}$$

V primeru izračuna števila odpovedi pri testiranju vpeljemo spremenljivke n (število komponent v testiranju), n_s (pričakovano število komponent, ki preživijo testiranje), n_f (pričakovano število komponent v odpovedi). Velja, da je $n = n_s + n_f$. Tako lahko izračunamo, da je

$$n_s = e^{-\lambda t} * n = e^{-3*10^{-8}*5000} * 2000 = 1999, n_f = 2000 - 1999 = 1. \tag{1.11}$$

1.6.2 Linearno rastoča intenzivnost odpovedovanja v fazi eksploatacije

Omenjena značilnost linearne rasti intenzivnosti odpovedovanja $\lambda(t)$ se pojavlja predvsem pri obrabljivih komponentah mehanske narave. Od elektronskih komponent so to predvsem releji. Funkcija intenzivnosti odpovedovanja se izraža kot

$$\lambda(t) = \lambda t, \tag{1.12}$$

kjer je λ konstanta. Odtod sledi, da je funkcija gostote verjetnosti časa odpovedi

$$f(t) = \lambda t e^{-\frac{\lambda t^2}{2}}, \tag{1.13}$$

in funkciji zanesljivosti in odpovedovanja

$$R(t) = e^{-\frac{\lambda t^2}{2}},\ F(t) = 1 - e^{-\frac{\lambda t^2}{2}}. \tag{1.14}$$

Potrebno je poudariti, da $f(t)$ v tem primeru sovpada z Rayleighovo distribucijo. Po njej se pričakovana vrednost življenske dobe in njena varianca po vrsti izražata kot

$$MTBF = MTTF = \int_0^\infty R(t)dt = \sqrt{\frac{\pi}{2\lambda}},\ \rho^2 = \frac{2}{\lambda}(1 - \frac{\pi}{4}). \tag{1.15}$$

Case 4 *Proizvajalec gum ob testiranju 150 gum iz nove serije ugotovi, da je intenzivnost odpovedovanja linearno rastoča in sicer jo na omenjenem vzorcu numerično oceni na* $\lambda(t) = 0.5 * 10^{-8}t$. *Določi zanesljivost takšne gume po enem letu uporabe in kakšen je pričakovani čas za menjavo gume (MTTF), ter njegova standardna deviacija?*

$$R(10^4) = e^{-\frac{0.5}{2} * 10^{-8} * 10^8} = 0.7788, \tag{1.16}$$

$$MTTF = MTBF = \sqrt{\frac{\pi}{2\lambda}} = \sqrt{\frac{\pi}{2 * 0.5 * 10^{-8}}} = 17.724 \text{ delovnih ur}, \tag{1.17}$$

$$\rho = \sqrt{\frac{2}{\lambda}(1 - \frac{\pi}{4})} = 9.265 \text{ delovnih ur}. \tag{1.18}$$

1.6.3 Linearno padajoča intenzivnost odpovedovanja v otroški fazi

Omenjena intenzivnost je tipična za pozno otroško dobo tako za mehanske, kot tudi elektronske komponente. Funkcija hazarda se izraža kot

$$\lambda(t) = a - bt,\ a \geq bt, \tag{1.19}$$

kjer sta a in b konstanti.

1.6.4 Weibullov model intenzivnosti odpovedovanja

V primerih ko intenzivnosti odpovedovanja ne moremo ponazoriti s konstanto ali linearno funkcijo uporabimo Weibullov model intenzivnosti odpovedovanja. Pri tem se $\lambda(t)$ izraža kot

$$\lambda(t) = \frac{\alpha}{\beta}t^{\alpha-1}, \tag{1.20}$$

funkcija gostote verjetnosti časa odpovedi pa kot

$$f(t) = \frac{\alpha}{\beta}t^{\alpha-1}e^{-\frac{t^\alpha}{\beta}}, t > 0. \tag{1.21}$$

Pri tem sta β in α pozitivni, po vrsti pa predstavljata karakteristični življenski in oblikovni parameter porazdelitve. Omenjeni model lahko pokriva več različnih

intenzivnosti odpovedovanj. Pri $\alpha = 1$ tako dobimo konstantno odpovedovanje, pri $\alpha > 1$ monotono rastočo intenzivnost odpovedovanja, pri $\alpha < 1$ monotono padajočo, pri $\alpha = 2$ pa linearno rastočo. Obe konstanti se določita glede na izkustvene podatke iz testiranj (časov odpovedovanj); povedano drugače se skuša najti ustrezno prileganje funkcije $\lambda(t)$ k podatkom o odpovedovanju. $MTBF$ ali $MTTF$ se izračuna kot

$$MTBF = MTTF = \int_0^\infty R(t)dt = \int_0^\infty e^{-\frac{t\alpha}{\beta}} = \beta^{\frac{1}{\alpha}}\Gamma(1+\frac{1}{\alpha}). \tag{1.22}$$

Pri tem Γ predstavlja *gama* funkcijo. Definirana je kot

$$\Gamma(n) = \int_0^\infty x^{n-1}e^{-x}dx. \tag{1.23}$$

Case 5 *Naročnik rešitve, ki jo servisira računalniški sistem, zahteva pričakovani medodpovedni čas 20.000 delovnih ur (MTBF). Intenzivnost odpovedovanja v intervalu meritev* 10^3 *delovnih ur sovpada z Weibullovim modelom s konstantami* $\alpha = 1,5$, $\beta = 100$. *Ali sistem zagotavlja zahtevani MTBF? Če ga ne, kakšen bi moral biti karakteristični življenski faktor* β *za zahtevani MTBF?*

$$MTBF' = 100^{\frac{1}{1,5}}\Gamma(1+\frac{1}{1,5}) = 19,383, \tag{1.24}$$

Ker je čas meritev trajal 10^3 *delovnih ur je tako MTTF* $19,383*10^3 = 19.383$ *delovnih ur, s čimer pričakovanja naročnika niso dosežena. Če hočemo doseči zahteve naročnika moramo zadostiti izrazu*

$$20.000 = \beta^{\frac{1}{1,5}}\Gamma(1+\frac{1}{1,5}), \tag{1.25}$$

kar vodi do izbrane vrednosti $\beta = 104,46$ [$Els1$].

1.7 Določanje zanesljivosti glede na arhitekturo sistema

Do sedaj smo se seznanili z osnovnimi pojmi kot so zanesljivost, funkcija intenzivnosti odpovedovanja in povprečni čas do odpovedi. Omenjeni pojmi glasijo tako na posamezne komponente, kot tudi na sistem kot celoto. Na sistemsko zanesljivost vplivajo tako zanesljivosti posameznih komponent, kot tudi njihova razporeditev v sistemu. Tako v splošnem velja, da z konfiguracijo zelo zanesljivih komponent lahko dosežemo enako zanesljivost, kot z večjim številom manj zanesljivih (cenejših) komponent v neki drugi konfiguraciji. Konfiguracijo sistema lahko obravnavamo kot [$Els1$]:

- serijsko vezavo entitet,

- paralelno vezavo entitet,
- serijsko paralelno vezavo entitet,
- paralelno serijsko vezavo entitet in
- mešano vezavo entitet.

Ko je sistem postavljen je potrebno določiti njegovo zanesljivost v odvisnosti od zanesljivosti posameznih entitet in vezave. Le to zanesljivost nato primerjamo z željeno zanesljivostjo, ki jo naj bi jo sistem imel. Če slednja ni dosežena, je potrebno konfiguracijo sistema ustrezno popraviti. Pri popravkih moramo seveda paziti, da sistem kot celota še zmeraj izpolnjuje funkcionalne zahteve in performančne zmožnosti. V pričujočem razdelku bomo zaradi preglednosti predpostavljali, da imajo entitete skozi čas enako verjetnost pravilnega delovanja, s čimer bomo z notacije $R(t)$ prešli na notacijo P. Seveda pa se moramo ves as zavedati, da P ni konstanta, temveč časovno pogojena padajoča funkcija.

1.7.1 Serijski sistem

Serijski sistem je sestavljen iz n serijsko vezanih entitet (komponent, funkcionalnih enot, podsistemov). Odpoved posamezne entitete rezultira v odpoved sistema kot celote. Vsaka entiteta ima seveda lahko na nekem nižjem abstraktnem nivoju zopet svojo konfiguracijo, ki je lahko poljubna (zgled serijske vezave je npr. avto: motor, 4 kolesa, zaviralni sistem, itd.). Vpeljimo naslednje termine:

x_i : i-ta entiteta deluje normalno,

$\overline{x}_i$: i-ta entiteta je v odpovedi,

$P(x_i)$: verjetnost, da i-ta entiteta deluje normalno,

$P(\overline{x}_i)$: verjetnost, da je i-ta entiteta v odpovedi,

P_{sys} : verjetnost delovanja sistema kot celote,

F_{sys}: verjetnost nedelovanja sistema kot celote.

Ob predpostavki, da odpovedi posameznih entitet ne vplivajo na odpovedi ostalih, bi tako izraza za sistemsko zanesljivost in nezanesljivost zapisali kot

$$P_{sys} = P(x_1) * P(x_2)...P(x_n) = \Pi_{i=1}^{n} P(x_i), \tag{1.26}$$

$$F_{sys} = 1 - P_{sys}. \tag{1.27}$$

Omeniti je potrebno, da je zanesljivost serijskega sistema venomer manjša ali enaka zanesljivosti komponente z najmanjšo zanesljivostjo.

Case 6 *Imamo serijsko vezan sistem iz 3 entitet, pri čemer so verjetnosti normalnega delovanja entitet po vrsti 0.9, 0.8 in 0.75. Zanesljivost in nezanesljivost tovrstnega sistema bi tako izračunali kot*

$$P_{sys} = P(x_1) * P(x_2) * P(x_3) = 0.54, F_{sys} = 1 - P_{sys} = 0.46. \tag{1.28}$$

1.7.2 Paralelni redundančni sistem

V paralelnih sistemih je vezanih n entitet paralelno in v splošnem odpoved ene od njih ne vodi do odpovedi sistema kot celote. Posamezne od paralelnih vej tvorijo *bremenske poti*. Tovrstni sistemi delujejo normalno vse dotlej, dokler deluje vsaj ena od paralelno vezanih entitet (bremenskih poti). Ob predpostavki, da odpoved ene od entitet ne vpliva na odpoved druge velja izraz za izračun nezanesljivosti

$$F_{sys} = \Pi_{i=1}^{n} P(\overline{x}_i), \tag{1.29}$$

preko njega pa pridemo do sistemske zanesljivosti

$$P_{sys} = 1 - F_{sys} = 1 - \Pi_{i=1}^{n} P(\overline{x}_i). \tag{1.30}$$

Če predpostavimo, da so vse komponente identične v smislu verjetnosti delovanja se gornji izraz poenostavi v

$$P_{sys} = 1 - P(\overline{x})^n = 1 - (1-p)^n, \tag{1.31}$$

kjer p predstavlja verjetnost za normalno delovanje posamezne komponente. Praviloma velja, da je zanesljivost paralelnega sistema večja ali enaka zanesljivosti najbolj zanesljive entitete v konfiguraciji. To potrjuje tudi naslednji zgled.

Case 7 *Predpostavimo, da imamo opravka s tremi paralelno vezanimi entitetami z verjetnostmi normalnega delovanja kot v prejšnjem zgledu. Zanesljivost sistema bi tako izračunali kot*

$$P_{sys} = 1 - \Pi_{i=1}^{3} P(\overline{x}_i) = 1 - (1-0.9)(1-0.8)(1-0.75) = 0.995. \tag{1.32}$$

1.7.3 Paralelno serijski sistemi

Paralelno serijski sistem je sestavljen iz m paralelnih bremenskih poti, vsaka od njih pa iz serijsko spojenih n entitet. Tako je sistem sestavljen v splošnem iz $n * m$ entitet. Predpostavimo, da je $P(x_{ij})$ verjetnost pravilnega delovanja j-te komponente na i-ti poti ($j = 1, 2, ...n, i = 1, 2, ...m$). Zanesljivost i-te poti se tako izraža kot

$$P_i = \Pi_{j=1}^{n} P(x_{ij}),\ i = 1, ...m,\ j = 1, 2, ...n. \tag{1.33}$$

Z izrazom $\overline{P}_i$ bomo označili nezanesljivost i-te poti in odtod izračunali sistemsko zanesljivost kot

$$P_{sys} = 1 - \Pi_{i=1}^{m} \overline{P}_i = 1 - \Pi_{i=1}^{m}(1 - \Pi_{j=1}^{n} P(x_{ij})). \tag{1.34}$$

Če predpostavimo, da imajo vse sistemske entitete enako verjetnost delovanja dobimo za sistemsko zanelsjivost izraz

$$P_{sys} = 1 - (1 - p^n)^m. \tag{1.35}$$

1.7.4 Serijsko paralelni sistem

Serijsko paralelni sistem je praviloma sestavljen iz serije n podsistemov, pri čemer v vsakem od slednjih najdemo paralelno vezanih m entitet. Zanesljivost takšnega sistema se izraža kot

$$P_{sys} = \Pi_{i=1}^{n} \left[1 - \Pi_{j=1}^{m}(1 - P(x_{ij}))\right], \ i = 1, 2...n, \ j = 1, 2, ...m. \tag{1.36}$$

Pri tem $P(x_{ij})$ predstavlja verjetnost, da j-ta kompontenta v i-tem podsistemu normalno deluje. Ob predpostavki, da vse entitete delujejo pravilno z verjetnostjo p, se sistemska zanesljivost izraža kot P

$$P_{sys} = [1 - (1 - p)^m]^n . \tag{1.37}$$

V splošnem velja, da ima serijsko paralelno konfiguriran sistem višjo zanesljivost, kot paralelno serijsko vezan sistem ob predpostavki, da imamo enako število entitet v konfiguraciji in imajo vse entitete enako verjetnost pravilnega delovanja.

1.7.5 Mešane vezave

Mešane vezave vsebujejo tako serijske kot paralelne vezave, pri čemer so entitete razporejene neenakomerno. Oglejmo si nekaj zgledov povzetih po [*Els*1], sestavljenih iz 6 komponent, za vsako od katerih je podana verjetnost delovanja $p = 0,85$.

Case 8 *Imamo tri različne konfiguracije šestih komponent z verjetnostjo pravilnega delovanja p=0,85 zvezane v paralelno serijsko vezavo (a), serijsko paralelno vezavo (b) in mešano vezavo (c), prikazane na sliki 1.2. Izračunaj verjetnosti delovanja vseh treh sistemov.*

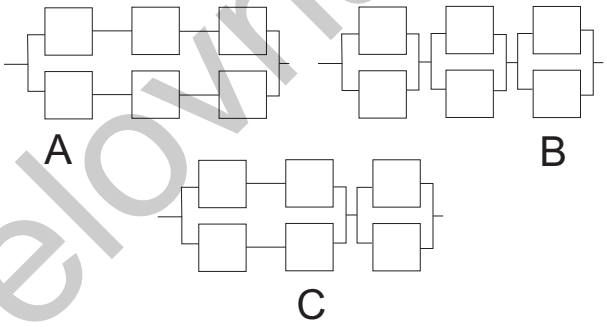

Figure 1.2: Zgledi treh različnih razporeditev 6 ekvivalentnih komponent.

$$P_{sysA} = 1 - (1 - 0,85^3)^2 = 0,8511,$$
$$P_{sysB} = \left(1 - (1 - 0,85)^2\right)^3 = 0,934007,$$
$$P_{sysC} = \left(1 - (1 - 0,85^2\right)^2) * (1 - (1 - 0,85)^2) = 0,924726.$$

1.7.6 Optimalna razporeditev entitet v konfiguraciji

Zanesljivost sistema kot celote je ob razpoložljivi kvoti entitet odvisna predvsem od njihovega načina vezave. Problem iskanja optimalne vezave je kombinatorično še kompleksnejši, če imamo opravka z entitetami katerih verjetnosti odpovedovanja so različne. Oglejmo si enega od enostavnejših algoritmov, ki nas glede na svojo naravo ne pripelje vedno do optimalne rešitve. Njegov cilj bo postavitev serijsko paralelne vezave, za katero smo že povedali, da je načeloma ugodnejša od paralelno serijske.

Predpostavimo, da je v serijsko paralelni vezavi n podsistemov $K_1, K_2, ...K_n$. V vsakem od podsistemov je m paralelno vezanih entitet. Celotno število entitet v sistemu je tako $u = n * m$. Entitete razvrstimo po padajočem vrstnem redu verjetnosti delovanja. Naš cilj je predstaviti algoritem, ki bo omenjene entitete razvrstil v sistem tako, da maksimiziramo zanesljivost sistema kot celote. Glede na serijsko paralelno vezavo je naš cilj sestaviti podsisteme tako, da bodo njihove zanesljivosti čimbolj podobne. Če ne bi bile, bi tako dobili izredno zanesljive podsisteme na eni strani, na drugi strani pa izredno nezanesljive podsisteme. Predstavljeni algoritem avtorjev Baxter and Harche (1992) je top-down hevristične (TDH) narave in vsebuje naslednje korake:

1. Razvrsti entitete po verjetnostih pravilnega delovanja v vektor u: $p_1 \geq p_2 \geq ... \geq p_u$.
2. Uvrsti entitete C_j v podsistem $K_j : j = 1, 2, ..., n$.
3. Uvrsti entitete C_j v podsistem $K_{2n+1-j} : j = n + 1, ..., 2n$.
4. $v = 2$.
5. Izračunaj $R_i^v = 1 - \Pi_{j \in K_i} q_j$ za $i = 1, 2, ...n$. Uvrsti C_{vn+i} v podsistem K_i, za katerega je R_i^v j-ti najmanjši ($j = 1, 2, ...n$).
6. Če je $v < m$ potem $v = v + 1$ in ponovi korak 5. Če je $v = m$ se ustavi.

Poglejmo si uporabo algoritma na konkretnem zgledu [$Els1$].

Case 9 *Predpostavimo, da gradimo sistem iz šestih komponent z verjetnostmi delovanja 0,95, 0,75, 0,85, 0,65, 0,4 in 0,55, pri čemer jih imamo namen razvrstiti v konfiguracijo n=2, m=3.*

1. u=(0,95, 0,85, 0,75, 0,65, 0,55, 0,4)

2. C_1 uvrstimo v podsistem K_1, C_2 v K_2.

3.) C_3 uvrstimo v podsistem K_2, C_4 v K_1.

4.) $v = 2$.

5.) $R_1^{(2)} = 1-(1-0,95)(1-0,65) = 0,9825, R_2^{(2)} = 1-(1-0,85)(1-0,75) = 0,9625$. Ker je $R_2^{(2)} \leq R_1^{(2)}$, C_5 uvrstimo v K_2.

6.) $v = 3$, C_6 uvrstimo v K_1. S tem je postopk končan. Končna zanesljivost sistema je 0,972802.

1.7.7 "k out of n" konfiguracija sistema

Sistem "k out of n" sistem deluje, če v njem deluje najmanja k od n razpoložljivih entitet. Tipičen primer "k out of n" sistemov so redundantne konfiguracije letalskih motorjev (Boeing 737,777, Airbus 320 "1 out of 2" konfiguraciji, Boeing 747 "3 out of 4" konfiguracija, DC-10 "2 out of 3" konfiguracija).

V splošnem se verjetnost delovanja natanko k od n enot izraža kot

$$p(k,n,p) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k}, \tag{1.38}$$

verjetnost zanesljivega delovanja, kjer lahko deluje tudi več kot k entitet pa kot

$$R(k,n,p) = \sum_{r=k}^{n} \left(\begin{array}{c} n \\ r \end{array} \right) p^r (1-p)^{n-r}. \tag{1.39}$$

Pri tem smo predpostavili, da imajo vse entitete enako verjetnost pravilnega delovanja. Oglejmo si še zgled uporabe povzet po [$Els1$].

Case 10 *Imamo komunikacijski sistem s 4 identičnimi paralelnimi komunikacijskimi kanali; sistem kot celota deluje, če delujejo vsaj 3 kanali. Če je verjetnost delovanja posameznega kanala p=0,9, se verjetnost delovanja sistema kot celote izraža po izrazu*

$$R(3,4,p) = \sum_{r=3}^{4} \left(\begin{array}{c} 4 \\ r \end{array} \right) p^r (1-p)^{4-r} = 4p^3(1-p) + p^4 = 4p^3 - 3p^4 = 0,9477.$$

1.7.8 "Consecutive *k* out of *n*:F" konfiguracija sistema

V pričujočem poglavju nas bodo zanimale odpovedi, tako da zapis "k out of n:F" usmerimo na opazovanje eventuelne odpovedi k ali več komponent, ki vodi do odpovedi sistema kot celote. Termin "Consecutive" usmerja naše opazovanje na odpovedovanje sklopa fizično sosednjih entitet. Tako za "Consecutive k out of n:F" sistem velja, da preide iz delovanja v stanje odpovedi, ko odpove najmanj k sosednjih od n serijsko vezanih entitet. Tipičen primer tovrstnih sistemov so konfiguracije satelitskih sistemov v sekvence, preko katerih se emitirajo signali. Običajno so sateliti postavljeni tako, da izpad enega satelita ne pomeni prekinitve komunikacije, če le ne pride istočasno tudi do odpovedi kakšnega od njegovih sosedov. Gre torej za "Consecutive 2 out of n:F" konfiguracijo sistema. Oglejmo si način izračuna zanesljivosti tovrstnih sistemov, pri čemer predpostavljamo, da so verjetnosti pravilnega delovanja posameznih entitet v serijski konfiguraciji enake.

$$R(p,2,n) = \sum_{j=0}^{\lfloor (n+1)/2 \rfloor} P\left[\text{sistem deluje in } j \text{ entitet je v odpovedi}\right]. \tag{1.40}$$

Z desnim delom gornjega izraza smo predpostavili, da je odpovedala največ polovica entitet v seriji. Če bi jih odpovedalo več, bi se v sekvenci pojavili vsaj dve sosedni entiteti v odpovedi, kar bi vodilo do odpovedi sistema kot celote. Če je torej odpovedalo največ pol entitet moramo zagotoviti še to, da je med dvema pokvarjenima vsaj ena delujoča entiteta. Število teh kombinacij je

$$\left(\begin{array}{c} (j+1)+(n-2j+1)-1 \\ n-2j+1 \end{array} \right) = \left(\begin{array}{c} n-j+1 \\ j \end{array} \right). \tag{1.41}$$

Tako lahko za zanesljivost zapišemo naslednji izraz

$$R(p,2,n) = \sum_{j=0}^{\lfloor (n+1)/2 \rfloor} \left(\begin{array}{c} n-j+1 \\ j \end{array} \right) (1-p)^j p^{n-j}. \tag{1.42}$$

Omenjeni izraz velja le za primer ko so verjetnosti delovanja entitet enake, algoritem za primer ko pa so verjentosti različne pa je dosti kompleksnejši.

Case 11 *Izračunaj verjetnost delovanja "Consecutive 2 out of 4:F" sistema pri verjetnosti delovanja posamezne komponente p=0,95.*

$$R(p,2,4) = \left(\begin{array}{c} 5 \\ 0 \end{array} \right) (1-p)^0 p^4 + \left(\begin{array}{c} 4 \\ 1 \end{array} \right) (1-p)^1 p^3 + \left(\begin{array}{c} 3 \\ 2 \end{array} \right) (1-p)^2 p^2 = 3p^2 - 2p^3 = 0,99275.$$

Izračunavanje zanesljivosti za primere, ko so verjetnosti delovanja entitet različne in se njihovo število povzpne, postanejo dosti bolj kompleksni. Poglejmo si primer hevrističnega algoritma avtorja Shantikumar-ja, ki rešuje omenjeni problem. Koraki algoritma so sledeči:

1. Izberi (k,n) in preveri če velja $(1 \leq k \leq n)$.
2. $q_i = 1 - p_i$, $(i = 1, 2..., n)$.
3. $F(r;k) = 0$, $r = 0, 1$.
4. $Q = \Pi_{i=1}^{k} q_i$.
5. $F(k;k) = Q$.
6. Do for $r = k+1$ to n:

   $Q = Q * \frac{q_r}{q_{r-k}}$

   $F(r;k) = F(r-1;k) + (1 - F(r-k-1;k)) * P_{r-k} * Q$
7. $R(n;k) = 1 - F(n;k)$.
8. End.

Oglejmo si zgled uporabe algoritma na konkretnem zgledu povzetem po [*Els*1]

Case 12 *V računalniškem omrežju imamo zaradi slabitve signalov postavljenih v sekvenci pet ojačevalnikov. Njihove verjetnosti odpovedi so po vrsti 0,62, 0,079, 0,25, 0,22 in 0,42. Uporaba algoritma je razvidna iz naslednjih korakov:*

1.
   - $k = 2, n = 5, p_1 = 0,38, p_2 = 0,921, p_3 = 0,75, p_4 = 0,78, p_5 = 0,58.$
   - $F(0;2) = 0, F(1;2) = 0, Q = q_1 q_2 = 0,62 * 0,079 = 0,0489, F(2;2) = 0,0489.$
   - $r = 3, Q = 0,0489 * \frac{q_3}{q_1} = 0,0197, F(3;2) = F(2,2) + [1 - F(0;2)] * 0,38 * 0,0197 = 0,0563;$
   - $r = 4, Q = 0,0197 * \frac{q_4}{q_2} = 0,0548, F(4;2) = F(3,2) + [1 - F(1;2)] * 0,921 * 0,0548 = 0,1068;$
   - $r = 5, Q = 0,0548 * \frac{q_5}{q_3} = 0,0920, F(5;2) = F(4,2) + [1 - F(2;2)] * 0,75 * 0,0920 = 0,1724;$
   - $R = 1 - F(5;2) = 0,827.$

1.8 *N* modularna redundanca

1.8.1 Uvod

N-modularna redundanca temelji na glasovalnih tehnikah, odtod tudi ime *glasovalna redundanca*. Ideja temelji na predpostavki, da se dva enaka sistema na isti vhod odzivata enako. V tem primeru lahko predpostavljamo, da se oba odzivata pravilno (delujeta normalno), ali pa se oba odzivata napačno (sta oba odpovedala na isti način). Sistem kot celota je preko različnih odzvivov lahko sposoben detektirati napako, ne zna pa je odpraviti. Najverjetnejši sta inačici ko oba sistema delujeta pravilno (enak odziv), ali en sistem deluje pravilno, drugi pa je v odpovedi (različna odziva). Tretja inačica - oba sistema sta v enaki odpovedi (enak odziv), je najmanj verjetna. Seveda imamo poleg 2-modularne redundance z glasovalno tehniko tudi 3-modularne, itd. V primeru ko večina modulov da eno odločitev, manjši preostanek pa drugačno, se večinoma zanašamo na večinsko odločitev, pri čemer prehajamo na *korekcijske* mehanizme in pa na koncept *večinske glasovalne redudnance*.

1.8.2 "Triple" modularna redundanca (TMR)

Najpogostejše sistemi modularne redundance temeljijo na treh podsistemih (N=3). Na sliki 1.3 je prikazan tovrstni sistem. Podsistemi A, B in C so neodvisni in ekvivalentni, glasovalnik (angl. *voter*) pa predstavlja glasovalni podsistem, kjer se primerjajo rezultati. Ugodne kombinacije delujočih naprav so ABC, $AB\overline{C}$, $\overline{A}BC$, $A\overline{B}C$, neugodne kombinacije pa $\overline{ABC}$, $\overline{A}BC$, $A\overline{BC}$ in $\overline{A}B\overline{C}$.

Izračun zanesljivosti v odvisnosti od verjetnosti delovanja posameznega podsistema p je podan v izrazu

$$R = B(3:3) + B(2:3) = \begin{pmatrix} 3 \\ 3 \end{pmatrix} p^3 + \begin{pmatrix} 3 \\ 2 \end{pmatrix} p^2(1-p) = p^2(3-2p). \quad (1.43)$$

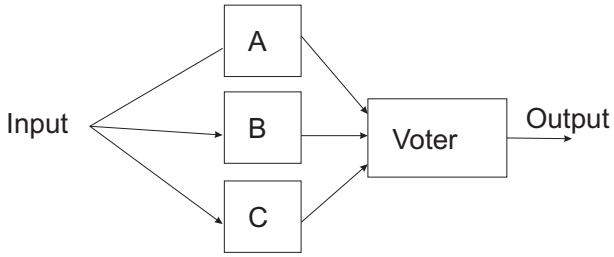

Figure 1.3: "'Triple' modularna redudnanca (N=3).

pri tem čemer smo predpostavljali, da glasovalnik ne more odpovedati. Omenjeni izraz sovpada z verjetnostjo delovanja "2 out of 3" sistema.

Case 13 *Predpostavimo, da imamo TMR sistem sestavljen iz treh podsistemov z verjentostjo delovanja p=0,9. Z uporabo gornjega izraza bi prišli do sistemske verjetnosti delovanja 0,972.*

Ob predpostavki, da ima posamezen podsistem konstantno intenzivnost odpovedovanja $\lambda(t)$, pridemo do izrazov

$$R_p(t) = e^{-\lambda t}, \tag{1.44}$$

$$R_{sys}(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t}. \tag{1.45}$$

pri čemer $R_p(t)$ predstavlja zanesljivost posameznega podsistema, $R_{sys}(t)$ pa zanesljivost sistema kot celote. MTTF bi izračunali po izrazu

$$MTTF = \int_0^\infty R(t)dt = \frac{3}{2\lambda} - \frac{2}{3\lambda} = \frac{5}{6\lambda}. \tag{1.46}$$

1.8.3 *N* modularna redundanca (NMR)

Število N je praviloma liho in v nekaterih primerih tudi večje od 3, za glasovalnik pa bomo predpostavljali, da je še vedno idealen, čemur v praksi ni tako. Za tovrstne sisteme veljajo naslednji izrazi:

$$N = 2n + 1, \tag{1.47}$$

$$R = \sum_{i=n+1}^{2n+1} B(i : 2n+1) = \sum_{i=n+1}^{2n+1} \left(\begin{array}{c} 2n+1 \\ i \end{array} \right) p^i * (1-p)^{2n+1-i} \tag{1.48}$$

Ob predpostavljeni konstantni intenzivnosti odpovedovanja $\lambda(t)$ pridemo do slike 1.4, ki prikazuje poteke funkcij zanesljivosti v odvisnosti od izbranega N-ja v času.

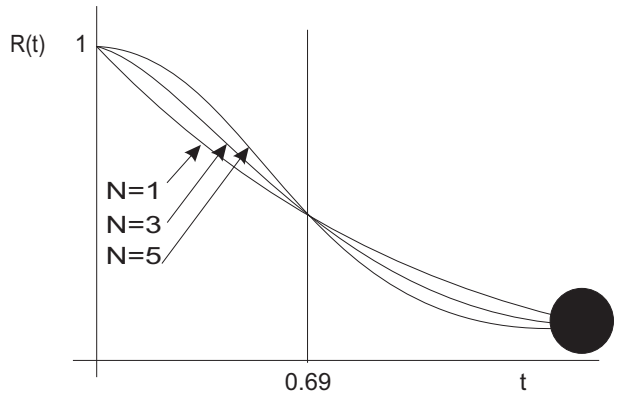

Figure 1.4: "Potek zanesljivosti v odvisnosti od števila N.

Iz slike je razvidno, da so vse zanesljivosti dosežene z redundanco "ugodne" do točke pri kateri velja, da je $\lambda t = 0,69$. Po tej točki so slabše od neredundantega sistema, zato moramo venomer za predvideno življensko dobo T preveriti, če produkt λT ne sega preko omenjene konstante. Glede na povedano, je potrebno za sistem najprej določiti predvideno življensko dobo (angl. *mission time*), šele nato pa N in λ.

1.8.4 Serijska vezava N modularnih sistemov

V nekaterih sistemih prenesemo glasovanje na podsistemske sklope, ki so vezani serijsko. Predpostavimo serijo m podsistemov z verjetnostjo delovanja posamezne komponente p. Omenjeni sistem je predstavljen na sliki 1.5.

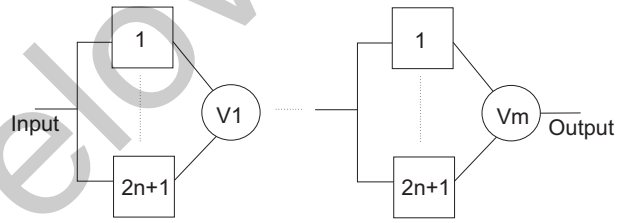

Figure 1.5: "Serijska vezava "N" modularnih sistemov.

Za takšen sistem izračunamo verjetnost delovanja po izrazu

$$P_{sys} = \left[\sum_{i=n+1}^{2n+1} \left(\begin{array}{c} 2n+1 \\ i \end{array} \right) p^i * (1-p)^{2n+1-i} \right]^m .$$

1.8.5 Glasovalni sistemi z neidealnim glasovalnikom

V realnih sistemih lahko odpove tudi glasovalnik. Vključimo možnost odpovedi glasovalnega sistema v naše izraze TMR sistema. Ob znani verjetnosti p za delovanje posamezne komponente pridemo do dopolnjenega izraza

$$p_{sys} = p_v(3p^2 - 2p^3),$$

kjer je p_v verjetnost pravilnega delovanja glasovalnika. Potek funkcije predhodnjega izraza je ponazorjen na sliki 1.6.

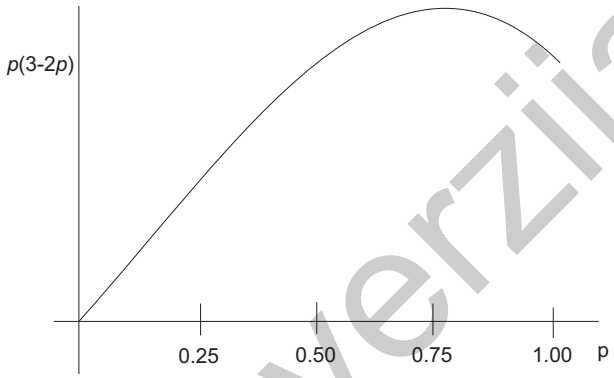

Figure 1.6: "Potek verjetnosti delovanj ob pokvarljivem glasovalniku.

Cilj redundance je, da postane sistemska verjentost p_{sys} večja od verjetnosti p posamezne komponente. Iz omenjenega dejstva sledi, da je naš cilj doseganje relacije

$$p_{sys} \geq p \rightarrow \frac{p_{sys}}{p} \geq 1 \rightarrow p * p_v(3 - 2p) \geq 1.$$

Slednji cilj je dosežen, ko je minimalni p_v (njegov minimum nas zanima zaradi cene realizacije) takšen, da zadostimo izrazu $p * p_v(3 - 2p) = 1$, kar je doseženo glede na maksimum funkcije pri $p = 3/4$ (glej sliko **??**): v tem primeru bi moral biti p_v večji od $8/9$, v ostalih primerih pa še večji.

1.9 Literatura

[$Con1$] L.W.Condra: Reliability Improvement with Design of Experiments, Marcel Dekker Inc., New York 2001.

[$Els1$] E A.Elsayed: Reliability Engineering, Addison Wesley Longman Inc., Reading, MA, 1996.

[$Lal1$] Parag K. Lala: Self - Checking and Fault Tolerant Digital Design, Morgan Kaufmann Publishers, USA, 2001.

[$Pet1$] I.Petersen: Fatal Defect - Chasing Killer Computer Bugs, Vintage Books, New York, 1996.

[$Sch1$] W.G.Schneeweiss: Petri Nets for Reliability Modeling, LiLoLe Verlag GmbH, 1999.

[$Xie1$] M.Xie, Y.S.Dai, K.L.Poh: Computing systems reliability, Kluwer Academic, 2004