Poglavje 1

Teorija zanesljivosti

V pričujočem poglavju bomo skušali razložiti in formalizirati pomembnejše pojme s področja teorije zanesljivosti. Začnimo z dvema najpogostejšima definicijama pojma *zanesljivosti*. Prvo najdemo v viru [1], drugo pa v virih [1], [2] in [3].

Definicija 1 *Zanesljiv je tisti sistem, ki počne natanko tisto, česar si želi uporabnik (kupec) sistema in to natanko takrat, ko to uporabnik od sistema zahteva.*

Navedena definicija je *kvalitativna* ter *uporabniško orientirana* in deloma sovpada s pojmom kvalitete sistema, saj uporabnik pogosto enači zanesljivost delovanja s pojmom kvalitete. Kakršnakoli odpoved sistema ali njegovo neustrezno delovanje uporabniku znižuje zaupanje do sistema in s tem posredno tudi zaupanje do proizvajalca/prodajalca sistema.

Definicija 2 *Zanesljivost je definirana kot verjetnost, da bo sistem vršil predvideno funkcijo v vnaprej podanem časovnem intervalu in v vnaprej podanih delovnih pogojih brez odpovedi.*

Druga definicija je *kvantitativna* ter *snovalsko orientirana* in meri zanesljivost v obliki verjetnosti pravilnega delovanja (delovanja brez odpovedi) sistema v opazovanem časovnem intervalu ali opazovani časovni točki. Pogojena je z *intenzivnostjo odpovedovanja*, ki jo označujemo z $\lambda(t)$ (angl. *failure rate function*, *hazard function* ali *hazard rate*), merimo pa s številom odpovedi na opazovanem 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 posameznih integriranih vezij.

Zanesljivost kot verjetnost, da bo sistem v časovnem intervalu $[0, t]$ svojo funkcijo vršil uspešno (torej ne bo odpovedal), zapišemo z izrazom

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

pri čemer T predstavlja časovno točko odpovedi [3]. *Verjetnost odpovedi* sistema v časovnem intervalu $[0, t]$ ali njegovo *nezanesljivost* zapišemo z izrazom

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

Če z $f(t)$ označimo funkcijo gostote verjetnosti časa odpovedi T, se zanesljivost $R(t)$ izraža kot integral gostote verjetnosti časa odpovedi [3] po izrazu

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

Osnovna *zanesljivostna metrika*[1], ki nas zanima pri opazovanju sistema, je *pričakovani čas do odpovedi* ali *pričakovana življenska doba* sistema. Metrika nastopa v dveh inačicah in sicer v prvi za *nepopravljive* in v drugi za *popravljive* sisteme. Prvo poimenujemo s kratico MTTF (angl. *mean time to failure*) in drugo s kratico MTBF (angl. *mean time between failures*). Obe inačici metrike izračunamo po izrazu

$$MTTF = MTBF = \int_0^\infty R(t)dt. \tag{1.4}$$

Druga zanesljivostna metrika, ki jo pogosto uporabljamo, je MTTR (angl. *mean time to repair*), predstavlja pa povprečni potrebni čas za izvedbo popravila ali menjavo komponente v popravljivih sistemih. Pri metriki MTBF v strokovni literaturi naletimo na neenotno pomensko obravnavo. MTBF namreč po definiciji predstavlja povprečni čas delovanja sistema med dvema zaporednima odpovedima. Pri tem bi po strogi uporabi definicije čas MTBF moral predstavljati čas delovanja sistema in čas sistema v odpovedi vse do odprave napake in ponovnega delovanja. V večini primerov čas odprave napake (MTTR) ni vštet v celotno vrednost MTBF. Pri tem poudarimo tudi to, da je čas MTTR običajno relativno majhen v primerjavi s predvidenim časom do odpovedi. V pričujočem delu časa MTTR v oceni časa MTBF ne bomo upoštevali.

Pojmov MTBF, MTTF in MTTR ne smemo mešati s pojmoma EOL (angl. *end of life*) in EOSL (angl. *end of service life*). EOL nam predstavlja časovno točko, ko proizvajalec neha proizvajati nek produkt, ni pa še ukinil svoje podpore produktu, EOSL pa časovno točko, ko proizvajalec ukine tudi podporo produktu. V splošnem bi relacije med omenjenimi pojmi lahko zapisali z izrazom

$$EOL \leq EOSL \leq MTTF. \tag{1.5}$$

Soroden pojem zanesljivosti je pojem *dosegljivosti*[2] (angl. *availability*), ki nam za opazovani računalniški sistem podaja pričakovan delež časa, v katerem

[1] Metrika - mera s katero izmerimo neko količino.

[2] Dosegljivost - merimo jo z deležem časa nahajanja sistema v delujočem stanju gledano preko daljšega časovnega obdobja.

bo sistem na razpolago za normalno delovanje. O dosegljivosti npr. običajno govorijo ponudniki internetnih storitev (angl. *internet service provider* - ISP), ki si za cilj postavljajo 99,999% dosegljivost svojega omrežja, kar bi v praksi pomenilo, da so internetne storitve nedosegljive približno pet minut na leto, čemur v praksi seveda ni tako.

1.1 Napaka, vgrajena hiba, odpoved

Večina zanesljivostnih problemov na področju programske in strojne opreme izhaja iz *napak* (angl. *error*), *vgrajenih hib* (angl. *fault*) in *odpovedi* (angl. *failure*) [4]. Združenje IEEE[3] pomene trojice pojmov definira na sledeče načine:

- 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 ali nedelovanje sistema kot celote, ki je rezultat vgrajene hibe;

Zaradi lažjega razumevanja in širšega pogleda na delovanje sistemov, bomo v nadaljevanju govorili le o napakah, ki bodo pokrivale prva dva zgoraj navedena termina in o odpovedih, kot eventuelnih posledicah napak.

Poljuben računalniški sistem je sestavljen iz dveh osnovnih sklopov in sicer iz *aparaturne* oziroma *strojne* in *programske* opreme. *Intenzivnost odpovedanja* računalniškega sistema kot celote, ki neposredno vpliva na njegovo zanesljivost, je pogojena z naslednjimi dejavniki:

- z intenzivnostjo porajanja odpovedi programske opreme;
- z intenzivnostjo porajanja odpovedi aparaturne opreme;
- z intenzivnostjo porajanja primerov nepravilnega načina interakcije med uporabnikom in sistemom;
- z intenzivnostjo porajanja ostalih zunanjih vplivnih dejavnikov (neustrezna temperatura, prevelika vlaga, udar strele itd.);

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

Nezanesljivost strojne opreme izhaja iz odpovedi posameznih elektronskih komponent, kar vodi do *odpovedi*, *nepravilnega* ali pa *degradiranega delovanja* aparaturne opreme kot celote. Pod pojmom degradiranega delovanja imamo v

[3] IEEE - Institute of Electrical and Electronics Engineers.

mislih sicer pravilno, a upočasnjeno delovanje sistema. Pri aparaturni opremi predpostavljamo, da vse komponente zapuščajo fazo testiranja brez napak ali okvar. Do napake in s tem posredno do odpovedi pride šele v *fazi eksploatacije* in sicer bodisi pod vplivom zunanjih dejavnikov, dotrajanosti materialov ali neustrezne tehnološke realizacije posameznih komponent.

1.2 Stanja delovanja opazovanega računalniškega sistema

Z vidika zanesljivosti opazovanega sistema lahko stanje njegovega delovanja uvrstimo v eno od sledečih skupin:

- sistem kot celota deluje pravilno in normalno;
- zaradi pojavitve napake sistem kot celota ne deluje oziroma je v *v odpovedi*;
- zaradi pojavitve napake sistem kot celota deluje nepravilno oziroma je *v odpovedi*;
- zaradi pojavitve napake odpove le del sistema, čigar funkcije prevzamejo preostali še delujoči deli sistema; v tem primeru običajno pride do upočasnitve delovanja sistema kot celote in pravimo, da sistem deluje *degradirano*;

Z vidika uporabnika je v drugem in tretjem primeru sistem neuporaben, v zadnjem primeru pa je sistem še uporaben, zmanjša pa se njegova *zmogljivost*[4].

1.3 Življenjska doba opazovanega računalniškega sistema

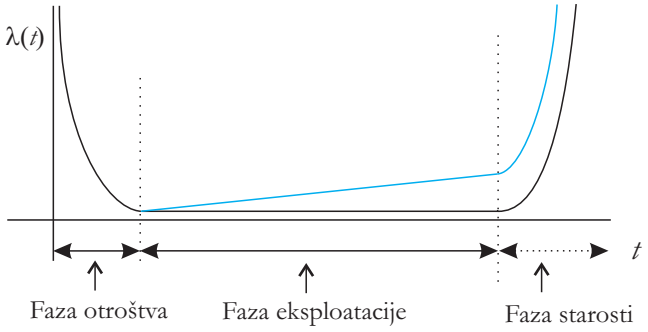
Življensko dobo opazovanega računalniškega sistema ali njegovih sestavnih delov (komponent) v splošnem delimo na sledeče tri faze:

- *faza otroštva*: v tej fazi intenzivnost odpovedovanja $\lambda(t)$ skozi čas hitro pada zaradi testiranja, ki iz sistema eliminira okvarjene komponente (če niso popravljive), ali pa se omenjene komponente popravijo (npr. aparaturne komponente in programsko opremo);
- *faza eskploatacije* (moderneje *produkcijska faza*): v tej fazi imamo opravka s konstantno ali rastočo (npr. linearno) obliko intenzivnosti odpovedovanja $\lambda(t)$; konstantna $\lambda(t)$ se uporablja pri vrednotenju elektronskih komponent, kot so integrirana vezja, linearno rastoča pa pri vrednotenju mehansko obrabljivih delov opazovanega sistema, kot so npr. avtomobilske gume, računalniški ventilatorji itd.;

[4] Zmogljivost - sposobnost česa, določena z zgornjo mejo učinkovitosti pri opravljanju nekega dela ali funkcije (Vir:SSKJ).

- *faza starosti*: v tej fazi začne intenzivnost odpovedovanja $\lambda(t)$ zelo hitro naraščati predvsem pri strojnih komponentah, ker prihaja do dotrajanosti materialov in s tem posledično do odpovedi komponent;

Vse tri faze so s funkcijo „kopalnične kadi“ (angl. *bathbub function*) prikazane na sliki 1.1, pri čemer je v fazi eksploatacije konstanten potek $\lambda(t)$ označen s črno, linearna rast $\lambda(t)$ pa z modro črto.



Slika 1.1: Različne faze življenske dobe računalniškega sistema z vidika intenzivnosti odpovedovanja.

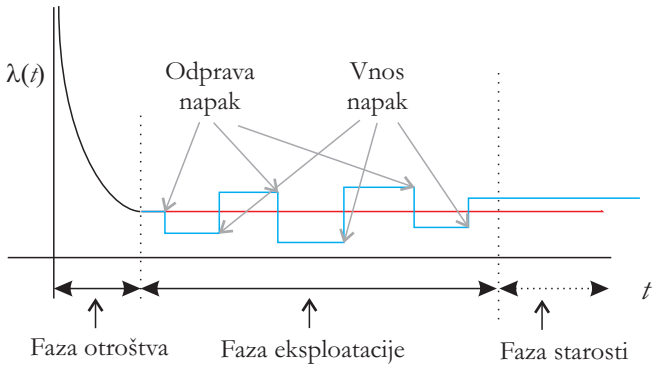
Poseben model intenzivnosti odpovedovanja lahko pripišemo programski opremi, ki je predstavljen na sliki 1.2. V otroški fazi je ta običajno zaradi najdb napak in njihove sprotne odprave padajoč, v eksploatacijski dobi pa imamo za intenzivnost odpovedovanja sledeči možnosti:

- programske opreme ne popravljamo in ne nadgrajujemo (rdeč potek na sliki 1.2); tako je intenzivnost odpovedovanja tako v eksploatacijski, kot tudi v fazi starosti konstantna;
- programsko opremo popravljamo in nadgrajujemo v obliki dodajanja novih funkcionalnosti); v tem primeru dobimo stopničasto funkcijo (moder potek na sliki 1.2), kjer vsak padec funkcije $\lambda(t)$ predstavlja odpravo določenega števila napak, vsak vzpon pa vnos novih napak preko dodajanja novih funkcionalnosti;

Ob predpostavki, da programske opreme v fazi starosti niti ne popravljamo, niti ne nadgrajujemo, rastočo $\lambda(t)$ v fazi starosti nadomesti konstanta intenzivnost odpovedovanja. S tem ponazorimo dejstvo, da se programska oprema ne ”obrablja”.

1.4 Modeli intenzivnosti odpovedovanja

Že v uvodu pričujočega poglavja smo se seznanili s funkcijo intenzivnosti odpovedovanja $\lambda(t)$ (angl. *failure rate function*, *hazard function*). Tako z vidika



Slika 1.2: Različne faze življenske dobe programske opreme z vidika intenzivnosti odpovedovanja.

uporabnika, kot tudi z vidika vzdrževalca, je najzanimivejša intenzivnost odpovedovanja v eksploatacijski dobi. Relacijo med $\lambda(t)$, $f(t)$ (funkcije gostote verjetnosti časa odpovedi) in $R(t)$ po viru [3] zapišemo z izrazom

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{R(t) - R(t + \Delta t)}{\Delta t R(t)} = \frac{f(t)}{R(t)}. \tag{1.6}$$

V sledečih razdelkih si bomo ogledali različne vrste intenzivnosti odpovedovanja $\lambda(t)$ v fazi eksploatacije.

1.4.1 Konstantna intenzivnost odpovedovanja v fazi eksploatacije

Konstantna $\lambda(t)$ je v eksploatacijski dobi tipična predvsem za elektronske komponente kot so tranzistorji, upori, kondenzatorji in integrirana vezja. Za slednje velja [2], da traja faza otroštva približno 10^4 delovnih ur oziroma približno eno delovno leto. V fazi testiranja posameznih komponent se ta doba umetno zmanjšuje s slabšanjem delovnih pogojev (angl. *burn-in procedures, accelerated testing*), s katerimi nadomestimo relativno dolgotrajno izpostavljanje običajnim pogojem delovanja. Funkcija intenzivnosti odpovedovanja se v eksploatacijski dobi izraža z izrazom

$$\lambda(t) = \lambda, \tag{1.7}$$

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

$$f(t) = \lambda e^{-\lambda t}, \tag{1.8}$$

funkciji zanesljivosti in nezanesljivosti pa po vrsti

$$R(t) = e^{-\lambda t}, \tag{1.9}$$

$$F(t) = 1 - e^{-\lambda t}. \tag{1.10}$$

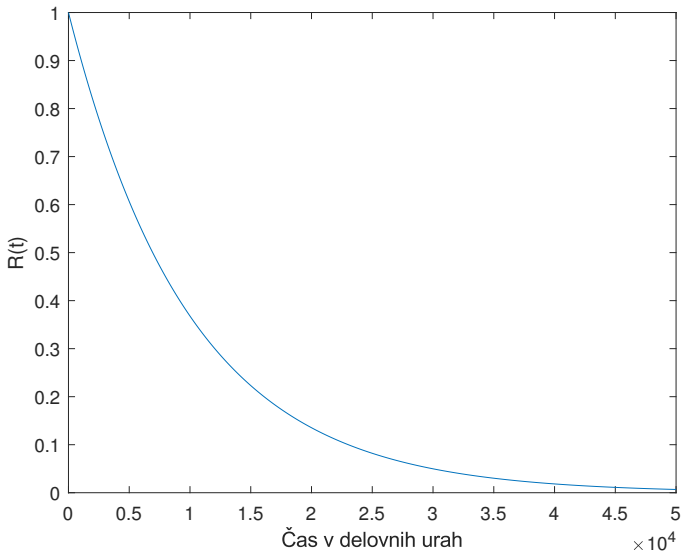
Po izrazu

$$MTBF = MTTF = \int_0^\infty R(t)dt = \int_0^\infty e^{-\lambda t}dt = \frac{1}{\lambda}, \tag{1.11}$$

izpeljemo pričakovano vrednost funkcije $f(t)$. Interpretiramo jo kot pričakovani čas do odpovedi. Čas za popravilo pri popravljivih sistemih se izraža kot

$$MTTR = \frac{1}{\mu}, \tag{1.12}$$

pri čemer μ predstavlja intenzivnost servisiranja ali inverzno vrednost časa servisiranja MTTR. Na sliki 1.3 je prikazan potek padajoče funkcije zanesljivosti ob intenzivnosti odpovedovanja $\lambda = 10^{-4}$ odpovedi na uro v času opazovanja $t = \frac{1}{2} * 10^5$ delovnih ur.



Slika 1.3: Potek funkcije zanesljivosti v odvisnosti od konstantne intenzivnosti odpovedovanja $\lambda = 10^{-4}$ in časa.

V nadaljevanju podajamo zgled izračuna zanesljivosti ob konstantni intenzivnosti odpovedovanja v eksploatacijski dobi.

Zgled 1 *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 urah delovanja in kolikšno je pričakovano število odpovedi po 5.000 delovnih urah na seriji velikosti 2.000 kosov?
Rešitev:

$$\lambda(t) = 3 * 10^{-8} odp./uro \rightarrow R(10^4) = e^{-3*10^{-8}*10^4} = 0.99970. \quad (1.13)$$

Za izračun števila odpovedi po 5.000 delovnih urah vpeljemo spremenljivke n (celotno število komponent), n_s (pričakovano število komponent, ki preživijo 5.000 delovnih ur) in n_f (pričakovano število komponent, ki ne preživijo 5.000 delovnih ur). Velja, da je $n = n_s + n_f$. Tako lahko izračunamo, da je

$$n_s = e^{-\lambda t} * n = e^{-3*10^{-8}*5000} * 2000 = 1999,7, \quad (1.14)$$

$$n_f = 2000 - 1999,7 = 0,3. \quad (1.15)$$

Glede na izračun lahko predpostavimo, da bo po 5.000 delovnih urah odpovedala 1 komponenta.

1.4.2 Linearno rastoča intenzivnost odpovedovanja v fazi eksploatacije

Značilnost linearne rasti intenzivnosti odpovedovanja $\lambda(t)$ v fazi eksploatacije se pojavlja predvsem pri obrabljivih mehanskih komponentah sistemov. Od računalniških komponent so to predvsem ventilatorji, obremenjeni konektorji, releji (kjer še nastopajo) itd. Funkcija intenzivnosti odpovedovanja se v tem primeru izraža kot

$$\lambda(t) = \lambda * t, \quad (1.16)$$

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

$$f(t) = \lambda * t * e^{-\frac{\lambda t^2}{2}}, \quad (1.17)$$

in funkciji zanesljivosti ter nezanesljivosti

$$R(t) = e^{-\frac{\lambda t^2}{2}}, \quad (1.18)$$

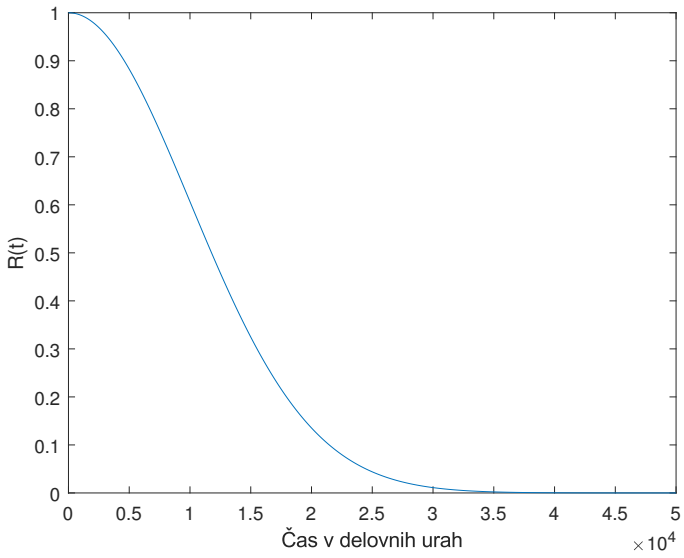
$$F(t) = 1 - e^{-\frac{\lambda t^2}{2}}. \quad (1.19)$$

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

$$MTBF = MTTF = \int_0^\infty R(t)dt = \sqrt{\frac{\pi}{2\lambda}}, \quad (1.20)$$

$$\rho^2 = \frac{2}{\lambda}(1 - \frac{\pi}{4}). \quad (1.21)$$

Na sliki 1.4 je prikazan potek padajoče funkcije zanesljivosti ob intenzivnosti odpovedovanja $\lambda(t) = t * 10^{-8}$ odpovedi na uro v času opazovanja $t = \frac{1}{2} * 10^5$ delovnih ur.



Slika 1.4: Potek funkcije zanesljivosti v odvisnosti od linearno naraščajoče intenzivnosti odpovedovanja $\lambda(t) = t * 10^{-8}$ in časa.

V nadaljevanju podajamo zgled izračuna zanesljivosti ob linearno rastoči intenzivnosti odpovedovanja v eksploatacijski dobi.

Zgled 2 *Proizvajalec ob testiranju vzorca 150 pnevmatik 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 onevmatike iz testirane serije po enem letu uporabe in kakšen je pričakovani čas življenske dobe (MTTF), ter njena standardna deviacija?*
***Rešitev**: Zanesljivost pnevmatike po enem letu izračunamo po izrazu*

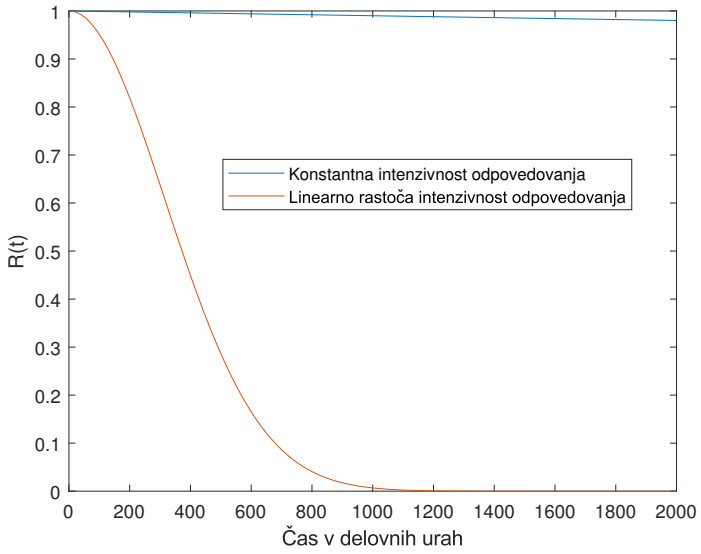
$$R(10^4) = e^{-\frac{0.5}{2} * 10^{-8} * 10^8} = 0,7788, \tag{1.22}$$

pričakovano življensko dobo in njeno standardno deviacijo pa po izrazih

$$MTTF = MTBF = \sqrt{\frac{\pi}{2\lambda}} = \sqrt{\frac{\pi}{2 * 0.5 * 10^{-8}}} = 17.724 \text{ delovnih ur}, \tag{1.23}$$

$$\rho = \sqrt{\frac{2}{\lambda}(1 - \frac{\pi}{4})} = 9.265 \ \textit{delovnih ur.} \tag{1.24}$$

Predhodno smo na slikah 1.3 in 1.4 predstavili poteka zanesljivostnih funkcij v odvisnosti od konstantne in linearno rastoče intenzivnosti odpovedovanja. Da bi dobili vtis, kako hitreje pada zanesljivostna funkcija ali verjetnost delovanja pri linearno rastoči intenzivnosti odpovedovanja, na sliki 1.5 predstavimo potek obeh vrst zanesljivostnih funkcij pri intenzivnosti odpovedovanja 1 odpovedi na 10^5 delovnih ur na opazovanem časovnem intervalu 2.000 delovnih ur.



Slika 1.5: Neposredna primerjava zanesljivostnih funkcij konstantne ($\lambda_1(t) = 10^{-5}$) in linearno rastoče intenzivnosti odpovedovanja ($\lambda_2(t) = 10^{-5}t$).

1.4.3 Linearno padajoča intenzivnost odpovedovanja v otroški fazi

Linearno padajoča intenzivnost odpovedovanja je tipična za pozno otroško dobo tako pri mehanskih, kot tudi pri elektronskih komponentah. Intenzivnost odpovedovanja se v tem primeru izraža kot

$$\lambda(t) = a - bt, \ a \geq bt, \tag{1.25}$$

kjer sta a in b konstanti.

1.4.4 Weibullov model intenzivnosti odpovedovanja

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

$$\lambda(t) = \frac{\alpha}{\beta} t^{\alpha-1}, \tag{1.26}$$

funkcija gostote verjetnosti časa odpovedi pa kot

$$f(t) = \frac{\alpha}{\beta} t^{\alpha-1} e^{-\frac{t^\alpha}{\beta}}, t > 0. \tag{1.27}$$

Pri tem sta α in β pozitivni števili, po vrsti pa predstavljata karakteristični oblikovni in življenski porazdelitvi. Omenjeni model lahko pokriva več različnih intenzivnosti odpovedovanj. Pri $\alpha = 1$ tako dobimo konstantno, pri $\alpha > 1$ monotono rastočo, pri $\alpha < 1$ monotono padajočo, pri $\alpha = 2$ pa linearno rastočo intenzivnost odpovedovanja. Obe konstanti se določita glede na izmerjene podatke iz testiranj ali natančneje iz časov odpovedovanj komponent. Povedano drugače, se skuša najti ustrezno prileganje funkcije $\lambda(t)$ k podatkom o odpovedovanju. $MTBF$ ali $MTTF$ se v tem primeru izračunata po izrazu

$$MTBF = MTTF = \int_0^\infty R(t)dt = \int_0^\infty e^{-\frac{t\alpha}{\beta}} dt = \beta^{\frac{1}{\alpha}} \Gamma(1 + \frac{1}{\alpha}), \tag{1.28}$$

pri čemer Γ predstavlja *gama* funkcijo. Definirana je z izrazom

$$\Gamma(n) = \int_0^\infty x^{n-1} e^{-x} dx. \tag{1.29}$$

V nadaljevanju si oglejmo zgled računske naloge povzete po viru [2].

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

$$MTBF' = 100^{\frac{1}{1,5}} \Gamma(1 + \frac{1}{1,5}) = 19,383. \tag{1.30}$$

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*

$$20.000 = \beta^{\frac{1}{1,5}} \Gamma(1 + \frac{1}{1,5}), \tag{1.31}$$

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

1.5 Redundanca kot metoda izboljševanja zanesljivosti

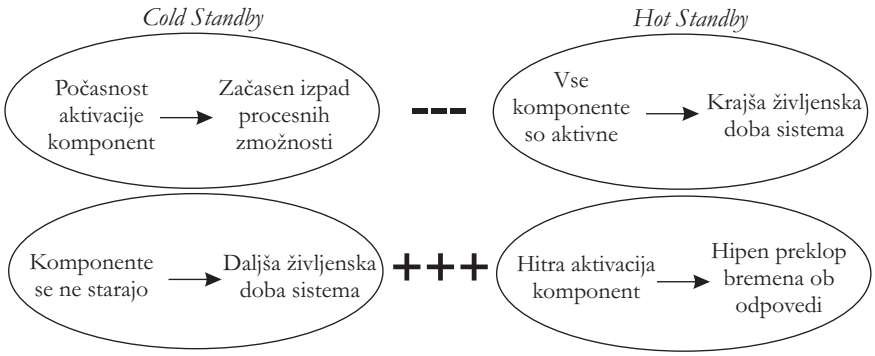
Na zanesljivost sistema kot celote vplivajo zanesljivosti posameznih sestavnih delov ali komponent sistema. Ena od osnovnih metod za doseganje željene zanesljivosti opazovanega sistema ob razpoložljivosti manj zanesljivih komponent je vgradnja *redundantnih* oziroma na prvi pogled odvečnih ali rezervnih komponent. Ob odpovedi posamezne komponente, bodisi zaradi napake ali izteka življenske dobe, funkcijo delujoče komponente prevzame redundantna oziroma rezervna komponenta. Z logičnega vidika je tako vezava redundantnih komponent paralelna. Z vidika delovanja sistema kot celote ni potrebno delovanje vseh redundantnih komponent. Običajno je dovolj, da je ob odpovedi ene komponente sistemsko breme zmožna prevzeti zgolj ena od redundantnih komponent. Tako ločujemo več možnih stanj posamezne komponente v redundančni vezavi. Le ta so sledeča:

- *aktivno delujoče stanje*: takšna komponenta z vidika sistema opravlja svojo funkcijo;
- *aktivno nedelujoče stanje*: takšna komponenta z vidika sistema ne opravlja svoje funkcije, jo je pa hipno zmožna prevzeti ob odpovedi predhodno omenjene aktivne delujoče komponente, ki je do tedaj opravljala svojo funkcijo (angl. *hot standby*);
- *pasivno nedelujoče stanje*: takšna komponenta z vidika sistema ne opravlja svoje funkcije, za njeno aktiviranje pa je potreben nek zagonski čas (angl. *cold standby*); šele po zagonu je komponenta zmožna prevzeti funkcijo aktivne delujoče komponente;
- *stanje v odpovedi*: takšne komponente brez servisiranja ali zamenjave ne moremo spraviti v eno od predhodno naštetih stanj;

Sisteme, v katerih imamo vgrajenih več komponent, 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 bo potovalo servisno breme (preusmeritev bremena na delujoče komponente). V obeh primerih za opravilo poskrbi hipotetična naprava, ki jo bomo poimenovali *stikalo* za (angl. *switch*). Tudi slednja ni idealna, tako da moramo računati tudi na njeno (ne)zanesljivost.

Slaba plat "cold standby" konfiguracije je v počasnosti časa zagona posamezne komponente in s tem tudi potencialen začasni izpad procesnih zmožnosti sistema. Dobra plat "cold standby" konfiguracije je v tem, da so komponente, ki ne servisirajo sistemskega bremena, v pasivnem nedelujočem stanju in njihova življenska doba ne teče (se ne "obrabljajo"), vse dokler niso aktivirane. Dobra

plat "hot standby" konfiguracije je v hitrosti preklopa, slaba pa v hitrem iztekanju življenske dobe več komponent hkrati. Vse naštete prednosti in slabosti so grafično prikazane na sliki 1.6, veljajo pa le za strojne oziroma aparaturne komponente, ne pa za programsko opremo.



Slika 1.6: Primerjava prednosti in slabosti "cold standby" in "hot standby" konfiguracije aparaturnih komponent.

1.6 Določanje zanesljivosti večkomponentnega sistema

Do sedaj smo se seznanili z osnovnimi metrikami ocenjevanja zanesljivosti kot so *funkcija zanesljivosti*, *funkcija intenzivnosti odpovedovanja* in *pričakovani povprečni čas do odpovedi*. Navedene metrike 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. V splošnem velja, da s konfiguracijo večjega števila manj zanesljivih ter posledično cenejših redundantno povezanih komponent lahko dosežemo enako zanesljivost sistema, kot s konfiguracijo manjšega števila bolj zanesljivih ter posledično dražjih komponent. Glede na načine povezanosti komponent ločujemo med sledečimi konfiguracijami sistemov [2]:

- serijska vezava komponent;
- paralelna vezava komponent;
- paralelno serijska vezava komponent;
- serijsko paralelna vezava komponent;
- mešana vezavo komponent;

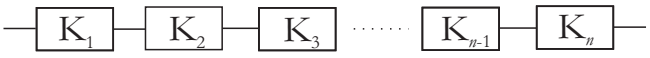
Vsak računalniški sistem z vidika snovanja in njegovih specifikacij temelji na sledečih treh množicah zahtev:

- **funkcionalne zahteve** definirajo funkcije, ki naj bi jih sistem opravljal;
- **zmogljivostne zahteve** definirajo hitrost odzivanja sistema;
- **zanesljivostne zahteve** definirajo njegovo dosegljivost in zanesljivostno funkcijo;

Ko je sistem dokončno zasnovan ali zgrajen, je potrebno izračunati njegovo *sistemsko zanesljivost* v odvisnosti od zanesljivosti posameznih komponent in načina njihovih medsebojnih povezav. *Izračunano sistemsko zanesljivost* nato primerjamo z *željeno zanesljivostjo* sistema. Če slednja ni dosežena, je potrebno konfiguracijo sistema ustrezno popraviti. Pri uvedbi popravkov moramo seveda paziti, da sistem kot celota še zmeraj izpolnjuje tako *funkcionalne*, kot tudi *zmogljivostne* zahteve naročnika.

1.6.1 Serijska vezava komponent

V primeru, da je sistem sestavljen iz n ($n > 1$) nujno potrebnih komponent, ga imenujemo za serijski sistem ali serijo komponent. Odpoved posamezne komponente rezultira v odpoved sistema kot celote. Za normalno delovanje sistema mora tako delovati vseh n komponent. Vsaka komponenta je lahko na nekem natančnejšem nivoju opazovanja zopet sestavljena iz množice manjših komponent z njim lastnimi zanesljivostmi. Grafično serijsko vezavo komponent ponazorimo s sliko 1.7. Za matematični izračun zanesljivosti delovanja serije komponent



Slika 1.7: Grafična ponazoritev konfiguracije s serijsko vezavo n komponent.

vpeljimo naslednje razlage numeričnih spremenljivk:

- x_i : i-ta komponenta sistema;
- $R_{x_i}(t)$: zanesljivost oziroma verjetnost delovanja i-te komponente v času t;
- $F_{x_i}(t)$: nezanesljivost oziroma verjetnost nedelovanja i-te komponente v času t;
- $R_{sys}(t)$: zanesljivost oziroma verjetnost delovanja sistema kot celote v času t;
- $F_{sys}(t)$: nezanesljivost oziroma verjetnost nedelovanja sistema kot celote v času t;

Ob predpostavki, da odpoved posamezne komponente ne vpliva na odpovedi ostalih komponent v konfiguraciji, izraze za sistemsko zanesljivost in nezanesljivost v času t, sistemsko intenzivnost odpovedovanja λ_{sys} in MTTF po viru [5]

zapišemo z izrazi

$$R_{sys}(t) = R_{x_1}(t) * R_{x_2}(t) ... * R_{x_n}(t) = \prod_{i=1}^{n} R_{x_i}(t), \tag{1.32}$$

$$F_{sys}(t) = 1 - R_{sys}(t), \tag{1.33}$$

$$\lambda_{sys} = \sum_{i=1}^{n} \lambda_i, \tag{1.34}$$

$$MTTF = \frac{1}{\sum_{i=1}^{n} \lambda_i}. \tag{1.35}$$

Izraza (1.34) in (1.35) veljata le takrat, ko so intenzivnosti odpovedovanja neodvisne od časa. Na tem mestu omenimo, da je zanesljivost serijskega sistema v času t venomer manjša ali največ enaka zanesljivosti komponente z najmanjšo zanesljivostjo v času t, kar zapišemo z izrazom

$$R_{sys}(t) \leq \min_{i=1...n} R_i(t). \tag{1.36}$$

Glede na povedano se izboljševanja zanesljivosti serije kompontent običajno polotimo z zvečanjem zanesljivosti najmanj zanesljive komponente v seriji.

Zgled 4 *Opravka imamo s serijsko vezanim sistemom iz treh elektronskih komponent, pri čemer so njihove deklarirane intenzivnosti odpovedovanja konstantne in podane z* $\lambda_1 = 10^{-4} odp./uro$, $\lambda_2 = 10^{-5} odp./uro$ *in* $\lambda_3 = 10^{-6} odp./uro$. *Kolikšni sta zanesljivost in nezanesljivost sistema kot celote po 10.000 delovnih urah?*
Rešitev: *Zanesljivost in nezanesljivost tovrstnega sistema v odvisnosti od časovne točke* t *se izračunata po izrazih*

$$R_{sys}(10^4) = R_{x_1}(10^4) * R_{x_2}(10^4) * R_{x_3}(10^4) =$$

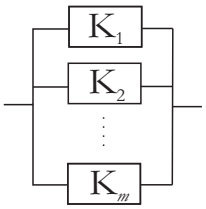
$$= 0,3679 * 0,9048 * 0,99 = 0,3296, \tag{1.37}$$

$$F_{sys}(10^4) = 1 - R_{sys}(10^4) = 0,6704. \tag{1.38}$$

1.6.2 Paralelna vezava komponent

Paralelni sistem je sestavljen iz m komponent, ki so vezane paralelno in vsaka od komponent v paralelni vezavi predstavlja eno od možnih *bremenskih poti*. Odpoved posamezne komponente oziroma bremenske poti ne vodi do odpovedi sistema kot celote. Do slednje pride šele takrat, ko odpove vseh m paralelno

vezanih komponent. Paralelno vezavo si tako interpretiramo kot m možnih paralelnih bremenskih poti po katerih se servisirajo zahteve in odraža *redundanco* komponent, pri čemer paralelna vezava kot celota deluje vse dotlej, dokler deluje vsaj ena od bremenskih poti. Na sliki 1.8 je predstavljena grafična ponazoritev konfiguracije s paralelno vezavo m komponent.



Slika 1.8: Grafična ponazoritev konfiguracije s paralelno vezavo m komponent.

Ob predpostavki, da odpoved ene od komponent ne vpliva na odpoved druge, velja za izračun nezanesljivosti v časovni točki t izraz

$$F_{sys}(t) = \prod_{i=1}^{m} F_{x_i}(t) = \prod_{i=1}^{m}(1 - R_{x_i}(t)), \tag{1.39}$$

preko njega pa pridemo do izraza za izračun sistemske zanesljivosti

$$R_{sys}(t) = 1 - F_{sys}(t) = 1 - \prod_{i=1}^{m}(1 - R_{x_i}(t)). \tag{1.40}$$

Če predpostavimo, da so vse komponente identične z vidika intenzivnosti odpovedovanja ($\forall i, 1 \leq i \leq m : R_{x_i}(t) = R_x(t)$), se predhodni izraz poenostavi v izraz

$$R_{sys}(t) = 1 - F_x(t)^m = 1 - (1 - R_x(t))^m, \tag{1.41}$$

kjer $R_x(t)$ predstavlja verjetnost delovanja posamezne komponente v časovni točki t. Velja, da je zanesljivost paralelnega sistema v časovni točki t večja ali enaka zanesljivosti najbolj zanesljive entitete v konfiguraciji v omenjeni časovni točki. Glede na povedano se običajno polotimo izboljševanja zanesljivosti paralelne vezave na ta način, da povečamo zanesljivost najbolj zanesljive komponente.

Zgled 5 *Opravka imamo s paralelno vezavo treh komponent s konstantnimi intenzivnostmi odpovedovanja* $\lambda_1 = 10^{-4} odp./uro$, $\lambda_2 = 10^{-5} odp./uro$ *in* $\lambda_3 = 10^{-6} odp./uro$. *Kakšen je potek zanesljivostne funkcije sistema kot celote skozi čas in kakšna je zanesljivost sistema po 10.000 delovnih urah?*
Rešitev:

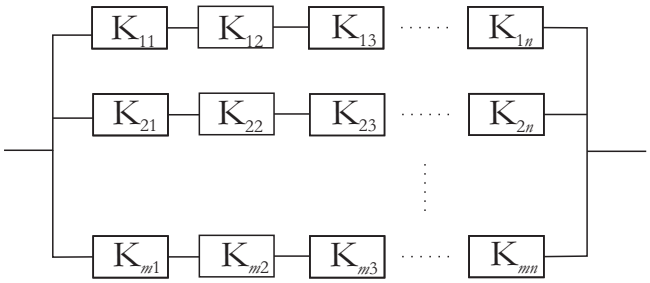
$$R_{sys}(t) = 1 - \prod_{i=1}^{3}(1 - R_{x_i}(t)) =$$

$$= 1 - (0,6321) * (0,0952) * (0,01) = 0,9994. \quad (1.42)$$

Pričujoči izračun je bil narejen na osnovi predpostavke, da je paralelna vezava v hot-standby konfiguraciji.

1.6.3 Paralelno serijska vezava komponent

Paralelno serijski sistem je sestavljen iz m paralelnih bremenskih poti, vsaka od njih pa iz serijsko spojenih n komponent, kot je to prikazano na sliki 1.9. Tovrstni sistem je tako sestavljen iz $n * m$ komponent. Predpostavimo, da je $R_{x_{ij}}(t)$ verjetnost pravilnega delovanja j-te komponente na i-ti bremenski poti $(i = 1, 2, ..., m, j = 1, 2, ..., n)$ v časovni točki t.



Slika 1.9: Grafična ponazoritev paralelno serijske vezave komponent.

Zanesljivost delovanje i-te bremenske poti v časovni točki t izračunamo po izrazu

$$R_i(t) = \prod_{j=1}^{n} R_{x_{ij}}(t), \ i = 1, ...m. \quad (1.43)$$

Z izrazom $F_i(t)$ označimo verjetnost nedelovanja oziroma nezanesljivost i-te poti v časovni točki t in odtod izračunamo sistemsko zanesljivost po izrazu

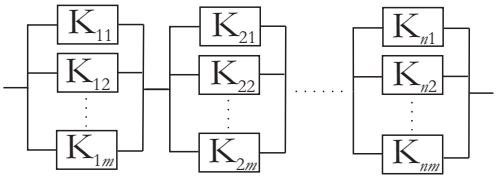
$$R_{sys}(t) = 1 - \prod_{i=1}^{m} F_i(t) = 1 - \prod_{i=1}^{m}(1 - \prod_{j=1}^{n} R_{x_{ij}}(t)). \quad (1.44)$$

Če predpostavimo, da imajo vse komponente enako intenzivnost odpovedovanja $(\forall(i,j), 1 \leq i \leq m, 1 \leq j \leq n : R_{x_{ij}}(t) = R_x(t))$, dobimo za izračun sistemske zanesljivosti v časovni točki t izraz

$$R_{sys}(t) = 1 - (1 - R_x(t)^n)^m. \quad (1.45)$$

1.6.4 Serijsko paralelna vezava komponent

Serijsko paralelni sistem tvori serija n podsistemov, vsak od njih pa je sestavljen iz m paralelno vezanih komponent. Grafična ponazoritev tovrstnega sistema je predstavljena na sliki 1.10. Zanesljivost takšnega sistema v časovni točki t se



Slika 1.10: Grafična ponazoritev serijsko paralelne vezava komponent.

izračuna po izrazu

$$R_{sys}(t) = \prod_{i=1}^{n} \left[1 - \prod_{j=1}^{m} (1 - R_{x_{ij}}(t)) \right]. \tag{1.46}$$

Pri tem $R_{x_{ij}}(t)$ predstavlja verjetnost, da j-ta kompontenta v i-tem podsistemu v časovni točki t deluje pravilno. Ob predpostavki, da imajo vse komponente enako intenzivnost odpovedovanja, se sistemska zanesljivost izračuna po izrazu

$$R_{sys}(t) = \left[1 - (1 - R_x(t))^m \right]^n. \tag{1.47}$$

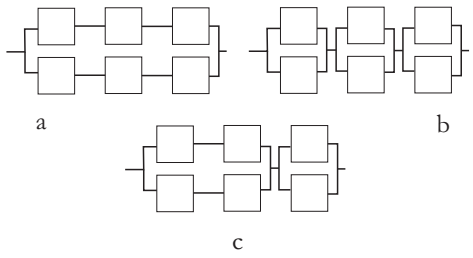
Ob predpostavki, da imamo enako število komponent v konfiguraciji in imajo vse entitete enako intenzivnost odpovedovanja velja, da ima serijsko paralelna konfiguracija sistema višjo zanesljivost, kot paralelno serijska konfiguracija.

1.6.5 Mešane vezave komponent

Mešane vezave vsebujejo tako serijsko, kot tudi paralelno vezane komponente, pri čemer so slednje lahko razporejene neenakomerno po posameznih blokih vezav. V zgledu v nadaljevanju prikažemo izračune zanesljivosti treh vzorčnih mešanih vezav povzetih po viru [2], ki so prikazane na sliki 1.11.

Zgled 6 *Imamo tri različne konfiguracije vezave šestih enakih komponent s konstantno intenzivnostjo odpovedovanja v izvedbi paralelno serijske vezave (a), serijsko paralelno vezave (b) in mešane vezave (c), ki so prikazane na sliki 1.11. Zanesljivost delovanja posamezne komponente v časovni točki t je $R_x(t) = 0,85$. Izračunaj verjetnosti delovanja vseh treh sistemov v opazovani časovni točki t.*
Rešitev:

$$R_x(t) = 0,85, \tag{1.48}$$



Slika 1.11: Zgledi treh različnih razporeditev šestih zanesljivostno ekvivalentnih komponent.

$$R_{sys_a}(t) = 1 - (1 - 0,85^3)^2 \cong 0,85, \tag{1.49}$$

$$R_{sys_b}(t) = \left(1 - (1 - 0,85)^2\right)^3 \cong 0,93, \tag{1.50}$$

$$R_{sys_c}(t) = \left(1 - (1 - 0,85^2\right)^2) * (1 - (1 - 0,85)^2) \cong 0,9022. \tag{1.51}$$

1.6.6 Optimalna razporeditev komponent v konfiguraciji

Zanesljivost sistema kot celote je ob vnaprej razpoložljivi kvoti komponent odvisna predvsem od njihovega načina vezave. Problem iskanja optimalne vezave je kombinatorično težak že v primeru uporabe komponent, ki imajo enako intenzivnost odpovedovanja, še kompleksnejši pa, če imamo opravka s komponentami, katerih intenzivnosti odpovedovanja so različne. V nadaljevanju si bomo ogledali enega od enostavnejših algoritmov za iskanje razporeditve komponent, ki nas glede na svojo naravo ne pripelje vedno do optimalne oziroma najzanesljivejše rešitve. Njegov cilj je 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 $P_1, P_2, ..., P_n$, v vsakem od njih pa m paralelno vezanih komponent, pri čemer so komponente funkcionalno ekvivalentne. Celotno število komponent v sistemu je tako $u = n * m$, pri čemer imajo komponente različne intenzivnosti odpovedovanja. Algoritem naj bi komponente razvrstil v serijsko paralelno vezavo tako, da maksimiziramo zanesljivost sistema kot celote. Glede na naravo serijske vezave je cilj algoritma sestaviti podsisteme tako, da bodo njihove zanesljivosti čimbolj podobne. Če ne bi bile, bi tako dobili v seriji po eni plati nekaj zelo zanesljivih podsistemov, po drugi plati pa nekaj dosti manj zanesljivih podsistemov. Slednji bi seriji kot celoti slabšali zanesljivost. V nadaljevanju predstavljeni algoritem avtorjev Baxter&Harche iz leta 1992 je "top-down" hevristične narave in je uporaben samo za razvrščanje komponent z enakim tipom intenzivnosti odpovedovanja. Z slednjim imamo v mislih to, da so vse intenzivnosti odpove-

dovanja bodisi konstantne, bodisi linearno naraščajoče itd. Korake algoritma strnemo v naslednje alinee:

1. Razvrsti komponente po kriteriju zanesljivosti v časovni točki t v padajočem vrstnem redu ($R_1(t) \geq R_2(t) \geq ... \geq R_u(t)$).
2. Uvrsti entitete $R_j(t)$ v podsistem $P_j : j = 1, 2, ..., n$.
3. Uvrsti entitete $R_j(t)$ v podsistem $P_{2n+1-j} : j = n+1, ..., 2n$.
4. $v = 2$.
5. Izračunaj $R(t)_i^v = 1 - \Pi_{j \in P_i} F_j(t)$ za $i = 1, 2, ...n$. Uvrsti $R_{vn+i}(t)$ v podsistem P_i, za katerega je $R_i^v(t)$ 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.

Oglejmo si uporabo algoritma na konkretnem zgledu povzetem po viru [2].

Zgled 7 *Predpostavimo, da gradimo sistem iz šestih komponent s konstantno intenzivnostjo odpovedovanja in podanimi zanesljivostmi 0,95, 0,75, 0,85, 0,65, 0,4 in 0,55 v opazovani časovni točki t, pri čemer jih imamo namen razvrstiti v konfiguracijo* ($n = 2, m = 3$) *oziroma serijo dveh podsistemov s po tremi paralelno vezanimi komponentami.*
Rešitev: *Upoštevajoč korake algoritma je rešitev sledeča:*
1. $R_1(t) = 0,95$, $R_2(t) = 0,85$, $R_3(t) = 0,75, R_4(t) = 0,65, R_5(t) = 0,55, R_6(t) = 0,4$),
2. Komponento z zanesljivostjo $R_1(t)$ *uvrstimo v podsistem* P_1, *komponento z* $R_2(t)$ *v* P_2.
3. Komponento z zanesljivostjo $R_3(t)$ *uvrstimo v podsistem* P_2, *komponento z* $R_4(t)$ *pa v* P_1.
4. $v = 2$.
5. $R(t)_1^{(2)} = 1-(1-0,95)(1-0,65) = 0,9825, R(t)_2^{(2)} = 1-(1-0,85)(1-0,75) = 0,9625$. *Ker je* $R(t)_2^{(2)} \leq R(t)_1^{(2)}$, *komponento z zanesljivostjo* $R_5(t)$ *uvrstimo v* P_2.
6. $v = 3$, *komponento z zanesljivostjo* $R_6(t)$ *uvrstimo v* P_1. *S tem je postopek končan. Končna zanesljivost sistema* $R_{sys}(t)$ *v opazovani časovni točki t je 0,972802.*

1.6.7 "*k* out of *n*" konfiguracija sistema

Sistem "k out of n" kot celota deluje, če v njem deluje vsaj k od n razpoložljivih entitet ($1 \leq k \leq n$). Tako konfiguracija "n out of n" predstavlja serijsko vezavo, "1 out of n" pa paralelno vezavo. Tipičen primer "k out of n" sistemskih konfiguracij so redundantne konfiguracije motorjev na potniških letalih [6]:

- Boeing 737, 777 in Airbus 320 imajo "1 out of 2" konfiguracijo nabora motorjev: plovila imajo dva motorja, za normalen potek leta pa je dovoljšen en delujoč motor; zaradi zagotavljanja varnosti v primeru odpovedi enega motorja skuša letalo pristati na najbližjem letališču;

- DC-10 ima "2 out of 3" konfiguracijo nabora motorjev: plovilo ima tri motorje, za normalen potek leta pa sta dovoljšnja dva delujoča motorja; tudi v tem primeru se ob odpovedi enega motorja skuša pristati na najbližjem letališču;

- Boeing 747, Airbus 340 in Airbus 380 imajo "3 out of 4" konfiguracijo nabora motorjev: vsa plovila imajo štiri motorje, za normalen potek leta pa so dovoljšnji trije delujoči motorji; tudi v tem primeru se ob odpovedi enega motorja skuša pristati na najbližjem letališču;

Verjetnost delovanja natanko k od n komponent v časovni točki t se izračuna v odvisnosti od zanesljivosti posamezne komponente $R_x(t)$ po izrazu

$$R_{sys}(t,k) = \left(\begin{array}{c} n \\ k \end{array} \right) R_x(t)^k (1 - R_x(t))^{n-k}, \tag{1.52}$$

verjetnost delovanja, kjer deluje k ali več kot k komponent pa po izrazu

$$R_{sys}(t) = \sum_{r=k}^{n} \left(\begin{array}{c} n \\ r \end{array} \right) R_x(t)^r (1 - R_x(t))^{n-r}. \tag{1.53}$$

Pri tem smo predpostavili, da imajo vse komponente enako intenzivnost odpovedovanja in s tem posledično enako funkcijo zanesljivosti $R_x(t)$. V nadaljevanju podajamo še računski zgled povzet po viru [2].

Zgled 8 *Opravka imamo s komunikacijskim sistemom s štirimi zanesljivostno identičnimi paralelnimi komunikacijskimi kanali. Sistem kot celota deluje, če delujejo vsaj trije kanali. Verjetnost delovanja posameznega kanala v opazovani časovni točki t je $0,9$ ($R_x(t) = 0,9$). Kolikšna je zanesljivost sistema kot celote v časovni točki t?*
Rešitev: *Verjetnost delovanja sistema kot celote v časovni točki t se izračuna po izrazu*

$$R_{sys}(t) = \sum_{r=3}^{4} \left(\begin{array}{c} 4 \\ r \end{array} \right) R_x(t)^r (1 - R_x(t))^{4-r} =$$

$$= 4R_x(t)^3(1 - R_x(t)) + R_x(t)^4 = 4R_x(t)^3 - 3R_x(t)^4 = 0,9477. \tag{1.54}$$

1.6.8 "Consecutive k out of n:F" konfiguracija sistema

Sistem v "Consecutive k out of n:F" konfiguraciji kot celota odpove, ko odpove k ali več njegovih sosednjih komponent. Termin "Consecutive" usmerja naše opazovanje na odpovedovanje komponet, ki so si v fizičnem smislu sosednje.

Tipičen primer "Consecutive k out of n:F" konfiguracij komponent so razporeditve satelitov po eliptični tirnici, ki obkroža zemljo, preko katerih se emitirajo signali. Običajno so sateliti postavljeni tako, da izpad enega satelita ne povzroči prekinitve komunikacije z uporabnikom, če le ne pride istočasno ali kasneje tudi do odpovedi kakšnega od njegovih neposrednih sosedov. V tem primeru govorimo o "Consecutive 2 out of n:F" konfiguraciji sistema.

Za primer, ko je $k = 2$, ter ob predpostavki, da so intenzivnosti odpovedovanja posameznih komponent v konfiguraciji enake in s tem posredno enake tudi zanesljivosti, sistemsko zanesljivost tovrstnih sistemov izračunamo po izrazu

$$R_{sys}(R_x(t), k = 2, n, t) =$$

$$= \sum_{j=0}^{\lfloor (n+1)/2 \rfloor} R\left[\text{sistem v časovni točki } t \text{ deluje, } j \text{ entitet je v odpovedi}\right]. \quad (1.55)$$

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

$$\left(\begin{array}{c} (j+1) + (n-2j+1) - 1 \\ n-2j+1 \end{array} \right) = \left(\begin{array}{c} n-j+1 \\ j \end{array} \right). \quad (1.56)$$

Tako lahko za izračun sistemske zanesljivosti "Consecutive 2 out of n:F" sistema zapišemo izraz

$$R_{sys}(R_x(t), k = 2, n, t) = \sum_{j=0}^{\lfloor (n+1)/2 \rfloor} \left(\begin{array}{c} n-j+1 \\ j \end{array} \right) (1-R_x(t))^j R_x(t)^{n-j}. \quad (1.57)$$

Navedeni izraz velja le za primer, ko so intenzivnosti odpovedovanja komponent enake, izračun za primer, ko pa so slednje različne, pa bi bil dosti kompleksnejši.

Zgled 9 *Opravka imamo s "Consecutive 2 out of 4:F" konfiguracijo sistema, ki je zgrajen iz enako zanesljivih komponent, zanesljivost posamezne komponente v časovni točki t pa je 0,95 ($R_x(t) = 0,95$). Kolikšna je zanesljivost sistema kot celote?*
Rešitev: *Sistemsko zanesljivost v časovni točki t izračunamo po izrazu*

$$R_{sys}(R_x(t), 2, 4, t) = \left(\begin{array}{c} 5 \\ 0 \end{array} \right) (1 - R_x(t))^0 R_x(t)^4 +$$

$$+\begin{pmatrix} 4 \\ 1 \end{pmatrix}(1-R_x(t))^1 R_x(t)^3 + \begin{pmatrix} 3 \\ 2 \end{pmatrix}(1-R_x(t))^2 R_x(t)^2 =$$
$$= 3R_x(t)^2 - 2R_x(t)^3 = 0,99275. \quad (1.58)$$

Izračunavanje zanesljivosti za primere, ko so zanesljivosti delovanja komponent različne (pri vseh pa je prisoten enak tip intenzivnosti delovanja) in se njihovo število povzpne, postane dosti bolj kompleksno. V nadaljevanju si bomo ogledali primer Shantikumarjevega hevrističnega algoritma, ki rešuje omenjeni problem. Koraki algoritma so sledeči:

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

   $Q = Q * \frac{F_r(r)}{F_{r-k}(t)}$

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

V nadaljevanju podajamo zgled uporabe algoritma na konkretnem primeru povzetem po viru [2].

Zgled 10 *V računalniškem omrežju imamo zaradi slabitve signalov postavljenih v sekvenci pet signalnih ojačevalnikov. Njihove verjetnosti nedelovanja v časovni točki t so po vrsti* $0,62$, $0,079$, $0,25$, $0,22$ *in* $0,42$.
Rešitev: *Uporaba algoritma je razvidna iz korakov navedenih v nadaljevanju:*

- *Korak 1:* $k = 2, n = 5, R_1(t) = 0,38, R_2(t) = 0,921, R_3(t) = 0,75, R_4(t) = 0,78, R_5(t) = 0,58$*;*
- *Koraki 2, 3, 4, 5:* $F(0;2) = 0, F(1;2) = 0, Q = F_1(t) * F_2(t) = 0,62 * 0,079 = 0,0489, F(2;2) = 0,0489$*;*

- *Korak 6:* $r = 3, Q = 0,0489 * \frac{F_3(t)}{F_1(t)} = 0,0197, F(3;2) = F(2,2) + [1 - F(0;2)] * 0,38 * 0,0197 = 0,0563$*;*
- *Korak 6:* $r = 4, Q = 0,0197 * \frac{F_4(t)}{F_2(t)} = 0,0548, F(4;2) = F(3,2) + [1 - F(1;2)] * 0,921 * 0,0548 = 0,1068$*;*
- *Korak 6:* $r = 5, Q = 0,0548 * \frac{F_5(t)}{F_3(t)} = 0,0920, F(5;2) = F(4,2) + [1 - F(2;2)] * 0,75 * 0,0920 = 0,1724$;
- $R_{sys}(t) = 1 - F(5;2) = 0,827$*;*

1.7 Glasovalni redundantni sistemi

Glasovalni redundantni sistemi (angl. *voting systems*) temeljijo na tehniki glasovanja posameznih komponent, ki sestavljajo glasovalni sistem. Pri tem se sistemski odziv oziroma njegova decizija formira na osnovi večinskega glasu [7].

Predpostavimo, da imamo v glasovalni sistem vgrajeni dve funkcionalno ekvivalentni komponenti. Osnovna ideja glasovalnih redundantnih sistemov temelji na predpostavki, da se bosta dve funkcionalno ekivalentni komponenti na isti vhod odzvali enako (bosta formirali enak odziv oziroma izhod). V primeru dveh komponent v glasovalnem sistemu smo soočeni s tremi možnimi situacijami:

- obe komponenti se odzoveta enako: pri tem je zelo velika verjetnost, da sta odziva pravilna (komponenti delujeta normalno) in zelo majhna verjetnost, da sta odziva napačna; v tem primeru bi morali komponenti odpovedati na enak način;
- komponenti se odzoveta različno: v tem primeru se glasovalni sistem zaveda prisotnosti napake, ne zna pa je odpraviti; omenjeno situacijo razrešimo z večjim številom vezanih komponent v glasovalnem sistemu;

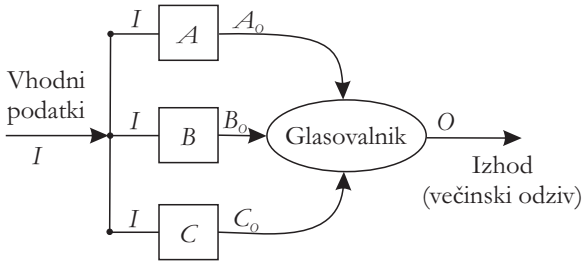
Običajno v glasovalne sisteme vgrajujemo več kot dve komponenti in praviloma je njihovo število liho. V tem primeru je sistem kot celota preko različnih odzivov komponent zmožen detektirati napako odziva manjšinskega števila komponent (npr. nekaj komponent se je odzvalo drugače, kot večina komponent). Pri tem predpostavljamo, da je večinski odziv komponent pravilen in se ga posreduje na izhod modularne vezave, preostanek odzivov, ki se od prej navedenih razlikuje, pa nas vodi do sklepa, da so komponente, ki so sprocesirale preostanek odzivov, v takšni ali drugačni odpovedi.

V strokovni literaturi poleg termina *glasovalne redundance* večkrat srečamo tudi pojem *modularne redundance*, ki je pomensko enak, s terminom modularnosti pa se skuša poudariti, da je število komponent v sistemu modularne narave oziroma razširljivo in s tem posledično skalabilno[5].

[5]Skalabilen sistem: sistem, ki ga je z vidika njegovih sestavnih delov - resursov možno

1.7.1 TMR modularna redundanca

Najpogosteje so sistemi modularne glasovalne redundance sestavljeni iz treh komponent ($N = 3$) (angl. *triple modular redundancy* - TMR). Na sliki 1.12 je prikazan TMR sistem s tremi glasovalnimi komponentami A, B in C, ki so običajno funkcijsko, zmogljivostno in zanesljivostno ekvivalentne, ter med seboj z vidika delovanja neodvisne. Vse komponente sistema sprejemajo enake vhode



Slika 1.12: „Triple“ modularna redundanca ($N = 3$).

(I), ter glede na njihove vrednosti procesirajo odzive oziroma izhode, ki jih posredujejo *glasovalniku* (angl. *voter*). Glasovalnik pregleda vse posredovane izhode (A_O, B_O, C_O) s strani komponent in na svoj izhod prenese večinski izhod glasovanja (O).

Predpostavimo, da oznaka A_O predstavlja pravilno sprocesiran izhod na pravilno delujoči komponenti A, oznaka $\overline{A}_O$ pa nepravilno sprocesiran izhod na nepravilno delujoči komponenti A, ki je v odpovedi. Enak sistem označevanja predpostavimo tudi za napravi B in C. Z vidika zanesljivega delovanja TMR sistema kot celote, so ugodne kombinacije izhodov $A_O B_O C_O$, $A_O B_O \overline{C}_O$, $\overline{A}_O B_O C_O$ in $A_O \overline{B}_O C_O$, neugodne pa kombinacije izhodov $\overline{A}_O \overline{B}_O \overline{C}_O$, $\overline{A}_O \overline{B}_O C_O$, $A_O \overline{B}_O \overline{C}_O$ in $\overline{A}_O B_O \overline{C}_O$. Pri slednjih kombinacijah je problematično to, če naprave v odpovedi sprocesirajo odziv na enak način, kar pa je malo verjetno, ker praviloma naprave ne odpovedujejo na enak način.

Izračun zanesljivosti TMR sistema v odvisnosti od verjetnosti delovanja posamezne komponente $R_x(t)$ v časovni točki t je podan z izrazom

$$R_{sys}(t) = B(3:3) + B(2:3) =$$

$$\left(\begin{array}{c} 3 \\ 3 \end{array} \right) R_x(t)^3 + \left(\begin{array}{c} 3 \\ 2 \end{array} \right) R_x(t)^2 (1 - R_x(t)) =$$

$$= R_x(t)^2 (3 - 2R_x(t)), \tag{1.59}$$

pri čemer smo predpostavili, da je glasovalnik idealen, kar pomeni, da ne more odpovedati). Navedeni izraz sovpada z zanesljivostjo delovanja "2 out of 3" sistema. V nadaljevanju podajamo še zgled konkretnega izračuna zanesljivosti.

razširiti tako, da je zmožen opravljanja večje količine dela.

Zgled 11 *Predpostavimo, da imamo TMR sistem sestavljen iz treh komponent, pri čemer je verjetnost delovanja posamezne komponente v opazovani časovni točki t* $0,9$ *(*$R(t) = 0,9$*). Kolikšna je zanesljivost TRM vezave tovrstnih komponent v časovni točki t?*
Rešitev: *Do izračuna zanesljivosti TMR sistema v časovni točki t pridemo na osnovi uporabe izraza*

$$= R_x(t)^2(3 - 2R_x(t)) = 0,972. \tag{1.60}$$

1.7.2 N modularna redundanca

N modularna redundanca (angl. *N modular redundancy* - NMR) definira redundančni glasovalni sistem sestavljen iz N komponent, pri čemer je N liho število večje od 3. Za glasovalnik bomo še vedno predpostavljali, da je idealen, čemur v praksi ni tako. Za tovrstne sisteme veljata izraza

$$N = 2n + 1, n = 2, 3, ..., \tag{1.61}$$

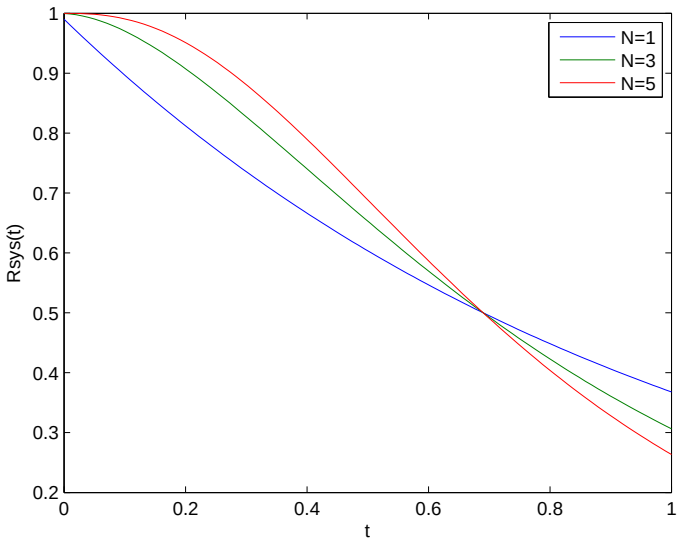
$$R_{sys}(t) = \sum_{i=n+1}^{2n+1} B(i : 2n + 1) =$$

$$= \sum_{i=n+1}^{2n+1} \left(\begin{array}{c} 2n+1 \\ i \end{array} \right) R_x(t)^i * (1 - R_x(t))^{2n+1-i}. \tag{1.62}$$

Ob predpostavki, da so intenzivnosti odpovedovanja konstantne, pridemo do grafičnih predstavitev potekov zanesljivostnih funkcij za različne vrednosti N, ki so predstavljene na sliki 1.13, povzeti po viru [7]. Iz slike je razvidno, da so vse zanesljivosti glasovalnih sistemov ($N > 1$) dosežene z redundanco "ugodne" le do neke točke na časovni osi, pri kateri velja, da je $\lambda * t = 0,69$. Po tej točki so redundantno dosežene zanesljivosti slabše od neredundantega sistema, zato moramo venomer za *predvideno življensko dobo T* preveriti, če produkt $\lambda * T$ ne sega preko omenjene kritične časovne točke. Glede na povedano, je potrebno za sistem najprej določiti predvideno življensko dobo (angl. *mission time, life time*), šele nato pa N in λ.

1.7.3 Serijska vezava N modularnih sistemov

V nekaterih sistemih je smiselno prenesti glasovanje na podsistemske sklope, ki so vezani serijsko. Predpostavimo, da imamo opravka s serijo m glasovalnih podsistemov, pri čemer je verjetnost delovanja posamezne komponente v podsistemu podana s funkcijo $R_x(t)$. Primer takšngea sistema je prikazan na na sliki 1.14. Za takšen sistem izračunamo verjetnost delovanja v časovni točki t



Slika 1.13: Primerjava funkcij zanesljivosti v odvisnosti od števila glasovalnih komponent N.

po izrazu

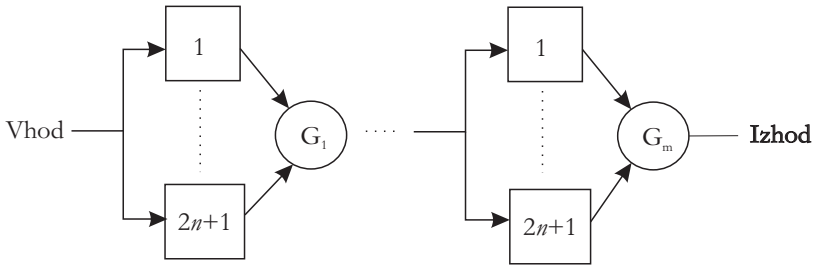
$$R_{sys}(t) = \left[\sum_{i=n+1}^{2n+1} \binom{2n+1}{i} R_x(t)^i * (1 - R_x(t))^{2n+1-i} \right]^m . \tag{1.63}$$

1.7.4 Glasovalni sistemi z neidealnim glasovalnikom

V realnih sistemih lahko odpove tudi glasovalnik, tako da moramo ob natančnem razčlenjevanju zanesljivosti sistema upoštevati tudi to možnost. Predpostavimo, da imamo opraviti s TMR sistemom in da funkcija zanesljivosti ni monotono padajoča, temveč v času konstantna [7]. Slednja predpostavka ni najbolj realna, a bomo nanjo pristali zaradi računskih poenostavitev. Zanesljivostno funkcijo posamezne komponente, ki smo jo do sedaj označevali z $R_x(t)$, bomo od tega mesta naprej označevali z oznako p, sistemsko zanesljivost celotnega TMR sistema pa z oznako p_{sys}. Ob dodatni predpostavki, da so vse tri glasovalne komponente zanesljivostno ekvivalentne, pridemo do izraza

$$p_{sys} = p_v(3p^2 - 2p^3), \tag{1.64}$$

pri čemer p_v predstavlja verjetnost pravilnega delovanja glasovalnika in tudi zanjo predpostavimo, da je skozi čas konstantna. Cilj vpeljave glasovalne redundance je, da postane sistemska verjetnost pravilnega delovanja glasovalnega



Slika 1.14: Serijska vezava N modularnih glasovalnih redundantnih sistemov.

sistema p_{sys} večja od verjetnosti p delovanja posamezne komponente. Iz omenjenega dejstva sledi, da je naš cilj doseganje veljavnosti relacije iz izraza

$$p_{sys} \geq p \implies \frac{p_{sys}}{p} \geq 1 \implies p * p_v(3 - 2p) \geq 1. \tag{1.65}$$

Pri tem nas zaradi doseganja čimmanjše cene izvedbe glasovalnika zanima minimalna vrednost p_v, pri kateri je relacija iz izraza (1.65) veljavna. Slednji cilj je dosežen, ko zadostimo izrazu

$$p * p_v * (3 - 2p) = 1, \tag{1.66}$$

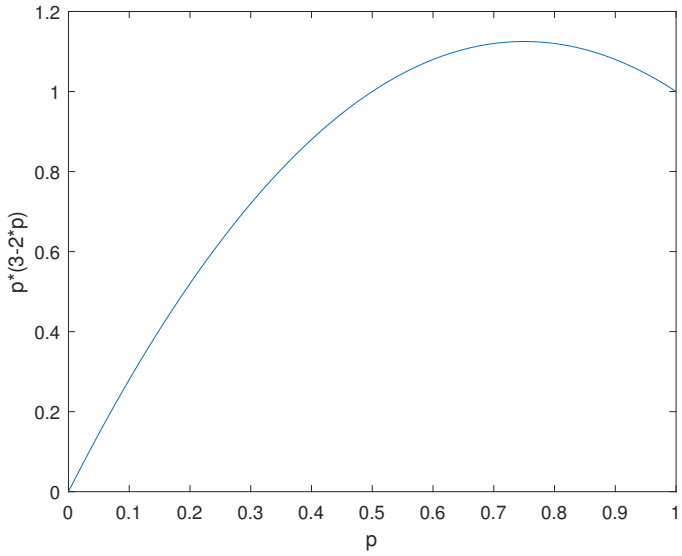
ki je pri maksimumu funkcije $p * (3 - 2p)$ pri $p = 3/4$ (glej sliko 1.15) dosegljiv z vrednostjo p_v, ki bi bila enaka ali večja od $8/9$, pri ostalih vrednostih p pa bi morala biti vrednost p_v še večja. Potek funkcije $p * (3 - 2p)$ je predstavljen na sliki 1.15.

1.8 Sistemski pogled na zanesljivost delovanja

V pričujočem poglavju smo spoznali osnove teorije zanesljivosti. Pri njenem spoznavanju smo se sklicevali po eni plati na zanesljivosti delovanja posameznih komponent sistema, po drugi plati pa na zanesljivost delovanja sistema kot celote. Pri izračunih celotne zanesljivosti opazovanega računalniškega sistema bi tako pod pojmom posameznih komponent morali upoštevati sledeče dejavnike:

- zanesljivost programske opreme;
- zanesljivost strojne opreme;
- zanesljivost upravitelja sistema (npr. administratorja, tehnika, vzdrževalca) ali uporabnika, pri čemer oba sodita med t.i. „človeške faktorje“;
- pogostost zunanjih nepredvidenih dogodkov (npr. udar strele in s tem posledično izpad električnega napajanja itd.);

Glede na povedano lahko naredimo sklep, da je določanje zanesljivosti v praksi bolj kompleksen problem, kot je predstavljen v teoriji izračuna zanesljivosti.



Slika 1.15: Potek funkcije $p * (3 - 2p)$.

Literatura

[1] L. W. Condra, *Reliability improvement with design of experiments.* Marcel Dekker Inc., 2001.

[2] E. A. Elsayed, *Reliability Engineering.* Addison Wesley Longman Inc., 1996.

[3] M. Xie, Y. S. Dai, and K. L. Poh, *Computing systems reliability: Models and analysis.* Kluwer Academic, 2004.

[4] I. Peterson, *Fatal defect - chasing killer computer bugs.* Vintage Books, USA, 1996.

[5] B. Dodson and D.Nolan, *Reliability engineering handbook.* Marcel Dekker, Inc., 1999.

[6] M. Rausand and A. Hoyland, *System reliability theory: Models, statistical methods, and applications.* John Wiley & Sons, Inc., 2004.

[7] M. L. Shooman, *Reliability of computer systems and networks: fault tolerance, analysis, and design.* J.Wiley and Sons, 2002.