Availability analysis for embedded system with N-version programming using fuzzy approach

Manju Kaushik

Department of CSE,
JECRC University,
Jaipur-303905, India
Email: manju.kaushik@jecrcu.edu.in

Gireesh Kumar*

Department of CSE,
JK Lakshmipat University,
Jaipur-302026, India
E-mail: gireesh8@gmail.com
E-mail: gireeshkaushik@jklu.edu.in
*Corresponding author

Preeti

Department of Mathematics,
NIMS University,
Jaipur-303121, India
Email: preetypro06@gmail.com

Richa Sharma

Department of Mathematics,
JK Lakshmipat University,
Jaipur-302026, India
Email: aligarh.richa@gmail.com
Email: richasharma@jklu.edu.in

Abstract: This present study deals with the conventional as well as fuzzy availability analysis of an embedded system considering both the software and hardware reliability. In this model, we have considered a specified number of subsystems composed of software and hardware components arranged in series. In N-version programming for the execution of the same task, N-versions of software are running at same time. Further, the assumption of rebooting is also taken in to consideration. Markov analysis has been done in which all the rates are exponentially distributed. To validate the proposed model, a numerical illustration is also carried out for conventional and fuzzy approaches for analysing the system performance by assuming the triangular membership function of the system descriptors.

Keywords: embedded system; N-version programming; NVP; reboot; Markov model; recursive method; fuzzy theoretic approach.

Reference to this paper should be made as follows: Kaushik, M., Kumar, G., Preeti, Sharma, R. (2015) 'Availability analysis for embedded system with N-version programming using fuzzy approach', *Int. J. Software Engineering, Technology and Applications*, Vol. 1, No. 1, pp.90–101.

Biographical notes: Manju Kaushik is an Associate Professor and currently working as the Head of Department of Computer Science and Application at JECRC University, Jaipur. She is involved in research activities in the field of software engineering. She received her PhD degree from the Mohan Lal Sukhadiya University, Udaipur. She has published 18 research papers in various national/international journals. She has also attended many conferences/seminars at national level.

Gireesh Kumar is an Assistant Professor at the Department of Computer Science and Engineering, JK Lakshmipat University, Jaipur. He has six years of academic and teaching experience. He received his MTech and BTech degrees in Computer Science and Engineering from Rajasthan Technical University, Kota and Uttar Pradesh Technical University, Lucknow, respectively. He is pursuing his PhD from JECRC University, Jaipur. He has published three research papers in international journal and conference. He has actively participated in various conferences and workshops. His research area includes software reliability and mathematical modelling.

Preeti is an Assistant Professor in Department of Mathematics in NIMS University, Jaipur. She did her MSc from CCS University, Meerut and MPhil and PhD in Mathematics from Shobhit University, Meerut. Her research areas of interest include reliability theory, probability theory, mathematical modelling, stochastic models and queuing theory. She has supervised an MPhil dissertation. She has 11 publications refereed international/national journals. She has participated and presented her research papers in 15 international/national conferences and attended six workshops.

Richa Sharma is an Assistant Professor of Mathematics in JK Lakshmipat University, Jaipur. She received her MSc and PhD in Mathematics from Dr B.R. Ambedkar University, Agra. She was awarded for the best paper presentation award in annual conference of Vijnana Parishad of India and The Global Society of Mathematical & Allied Sciences held at Meerut during 2–26 March 2011. She is a reviewer of two reputed international journals. She has six years of research and teaching experience. Her research area includes queuing theory, probability theory, stochastic modelling and optimisation. She has published 15 research papers in refereed international/national journals. She has participated and presented her research papers in 18 international/national conferences and workshops.

1 Introduction

Embedded systems have become a very important and essential part of our daily lives. An embedded system is a computer system which control many devices in common use today. Further, embedded system plays a vital role in our daily life which can be found in cell phones, microwave ovens, digital cameras, camcorders, portable video games, home security systems, washing machines and many more devices. Wattanapongsakorn and Levitan (2004) presented four models for optimising the reliability of embedded systems considering both software and hardware reliability under cost constraints. Wu et al. (2010) discussed the characteristics of the embedded software. They have analysed that the embedded software are real time and embedded which are the difference between embedded software and ordinary software.

Reliability has become an even greater concern in recent years because high-tech industrial processes with increasing levels of sophistication comprise most engineering systems today. Increasing dependence of mankind on embedded and computer-based systems attracted the attention of the software engineers and reliability researchers in the early 1970s to study the reliability aspect of embedded systems. Reliability is probably the most important characteristics inherent in the concept of 'software quality'. Software reliability helps to the software developers and users for increasing the system efficiency. To have effective quality of software and hardware, it is necessary to calculate the reliability and availability of embedded system as a whole. Software and hardware reliability has been discussed in the work of Jelinski and Moranda (1972). A study on software reliability growth model has been done by Keiller and Miller (1991), Downs and Scott (1992), Satoh and Yamada (2001), Yamada et al. (2000), respectively, in different frameworks. Elyezjy (2012) focused on applying software quality assurance practices for increasing the efficiency of the system. Recently, many prominent researchers were studied in different framework to improve the quality of the software. Some notable work done by the researchers, we refer the work of Calikh and Bener (2013), Zhang et al. (2013) and many more.

Software reliability can be improved by tolerating software faults using the concept of N-version programming (NVP). Moreover, NVP employs redundancy at all levels from design/development to execution. The main reason for applying the concept of NVP is for improving the reliability and availability of the software. To improve the quality of the software, the concept of NVP has been applied by many researchers in their study. NVP for fault tolerance was briefly suggested by Avizienis and Chen (1977). The concept of NVP through comparison of a redundancy method was used by Avizienis and Chen (1978) for hardware designers. According to this method, N-fold computation is carried out by using N independently designed software modules called versions which provide a single decision by decision algorithm. Moreover, the independent generation of $N \geq 2$ software versions are termed as NVP. The main advantage of NVP is to minimise the probability of similar errors at decision points, different algorithms, programming languages, environments and tools wherever possible. Recently, the work on NVP was done by Min et al. (2013) in their study.

Fuzzy technique is a soft computing approach which has facilitated the task of computation for many complex systems in different frameworks. Fuzzy approaches are often used in decision-making processes, especially when sources of uncertainty are involved. Ranking engineering design concepts using a fuzzy outranking preference model has been analysed by Wang (2001). A fuzzy approach is used for availability analysis of an embedded system considering both the software and hardware reliability. The suggested approach is illustrated on a model. The use of fuzzy logic to model the reasoning of the designer and the implicit subjectivity in how this designer solves the problem in practice, modelling variables as fuzzy linguistic variables. The feasibility of combining fuzzy logic with metaheuristic algorithms, i.e., applying soft computing for solving the hardware/software partitioning (HSP) problem. Verdegay et al. (2008) focused on heuristics as a fundamental constituent of soft computing. Further, Pando et al. (2013) discussed the application of fuzzy logic for HSP in embedded systems.

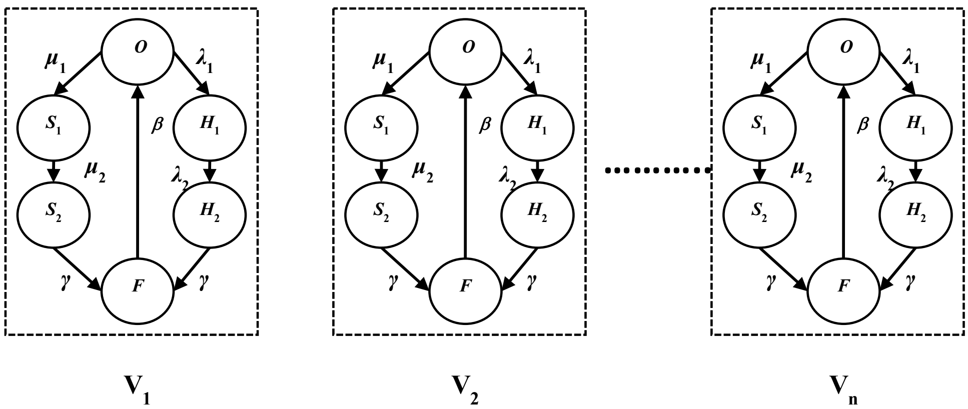
Recently, Shaout and Trivedi (2013) suggested the stage-wise fuzzy reasoning model for performance rating.

In this paper, embedded system with NVP has been analysed using fuzzy theoretic approach. In our study, a new modelling approach for determining the reliability and availability of an embedded system is proposed by considering all the components of the system and their hierarchy in the system structure. In this hierarchical structure, we have considered the concept of reboot when the system has completely failed. According to this, we try to reduce the downtime caused by reboots as much as possible. Yamakita et al. (2011) has been applied the concept of rebooting for operating system in their study. The rest of the paper is organised as follows. Section 2 presents the model description for embedded system with N-version programming with state governing equations of the given model. Section 3 presents the analysis of given model using recursive method and 4 discuss the fuzzy theoretic approach in related to the availability analysis of the model. In Section 5 we describe numerical results for validation purpose. Also the comparison between the conventional approach and the fuzzy theoretic approach is just taken to verifying the accuracy of the results. Finally, paper comes to end with concluding remarks in Section 6.

2 Model description

For the purpose of reliability analysis, we have considered an embedded system in which software and hardware units are arranged together. Markov analysis is a very powerful tool for analysing the reliability of any concern system. Moreover, Markov analysis may provide the optimistic prediction for the system reliability and safety parameters. Also, the Markov analysis helps to describe the all possible states of the system.

Figure 1 Transition rate diagram



For this purpose, the Markov analysis has been done under the consideration of reboot. There are N-version of each subsystems of the entire system. In each subsystem there are two alternates primary and secondary of each software versions and two hardware units operating and spare in each version execution. The hardware and software units of operating system is represented using symbol 'H_j' and 'S_j', respectively, where j = (1, 2).

If any one of these version is failed, the subsystem can be restored by the reboot. Some notations are used for modelling purpose which are given as:

Notations

$P_{0,i}$ steady state probability that the subsystem is in fully working state in i^{th} ($i = 1, 2, \ldots, n$) version

$P_{S_{j,i}}$ steady state probability that j^{th} (1, 2) software unit is working state in the subsystem of i^{th} ($i = 1, 2, \ldots, n$) version

$P_{H_{j,i}}$ steady state probability that j^{th} (1, 2) hardware unit is working state in the subsystem of i^{th} ($i = 1, 2, \ldots, n$) version

$P_{F,i}$ steady state probability that the subsystem is in the failed stated of i^{th} ($i = 1, 2, \ldots, n$) version

λ_1 transition rate from state 0 to state H_1

λ_2 transition rate from state H_1 to state H_2

μ_1 transition rate from state 0 to state S_1

μ_2 transition rate from state S_1 to state S_2

γ transition rate from state H_2 and S_2 to failed state F

β transition repair rate from the failed state H_2 to restoring the system to working state 0.

Under the consideration of all assumptions and using the state transition diagram as shown in Figure 1, we provide the steady state governing equations of a i^{th} ($i = 1, 2, \ldots, n$) version of the embedded system which are given as:

$$(\lambda_1 + \mu_1) P_{0,1} = \beta P_{F,1} \tag{1}$$

$$\lambda_2 P_{H_{1,i}} = \lambda_1 P_{0,i} \tag{2}$$

$$\gamma P_{H_{2,i}} = \lambda_2 P_{H_{1,i}} \tag{3}$$

$$\mu_2 P_{S_{1,i}} = \mu_1 P_{0,i} \tag{4}$$

$$\gamma P_{S_{2,i}} = \mu_2 P_{S_1,i} \tag{5}$$

$$\beta P_{F,i} = \gamma \left(P_{H_{2,i}} + P_{S_{2,i}} \right) \tag{6}$$

and the normalising condition for i^{th} version ($i = 1, 2, \ldots, n$) is

$$\left(P_{0,i} + P_{H_{1,i}} + P_{H_{2,i}} + P_{S_{1,i}} + P_{S_{2,i}} + P_{F,i} \right) = 1 \tag{7}$$

The formulation of the equations (1)–(7) is given in Appendix.

3 Recursive method

Recursive method has been applied by many researchers for analysing many complex problems arising in many real world situations. A general idea behind this is that if we have a complex situation that can be broken into basic structure as the original each time, this method calls itself; the situation becomes simpler until it converges to the simplest possible case. For the analysis purpose, recursive method has been applied for solving the steady state equations (1)–(7) for obtaining the steady state probabilities. Then, these steady state probabilities are used to evaluate the availability of the embedded system.

Thus, the availability of a i^{th} ($i = 1, 2,\ldots,n$) version is denoted by A_{V_i} and obtained by:

$$A_{V_i} = P_{0,i} + P_{H_{1,i}} + P_{H_{2,i}} + P_{S_{1,i}} + P_{S_{2,i}} = 1 - P_{F,i} \tag{8}$$

$$A_{V_i} = \frac{(\beta - \lambda_1 - \mu_1)(\lambda_2\mu_2\gamma)}{\beta\left[\beta\lambda_1\lambda_2\mu_2\gamma + \lambda_2\mu_2\gamma(\lambda_1 + \mu_1) + \gamma\lambda_1\mu_2 + \lambda_1\lambda_2\mu_2 + \mu_1\lambda_2\gamma + \mu_1\mu_2\lambda_2\right]} \tag{9}$$

Thus, the total availability of the entire embedded system is given by

$$\begin{aligned} AV &= \sum_{i=1}^{n} A_{V_i} \\ &= \frac{(\beta - \lambda_1 - \mu_1)(\lambda_2\mu_2\gamma)}{\beta\left[\beta\lambda_1\lambda_2\mu_2\gamma + \lambda_2\mu_2\gamma(\lambda_1 + \mu_1) + \gamma\lambda_1\mu_2 + \lambda_1\lambda_2\mu_2 + \mu_1\lambda_2\gamma + \mu_1\mu_2\lambda_2\right]} \times n \end{aligned} \tag{10}$$

4 The fuzzy theoretic approach

We develop a fuzzy approach that provides system characteristics for the embedded system with four fuzzified variables, i.e., failure rates of both the software and hardware. To apply the fuzzy approach in this model, the all equations and the particular relationships are used by using the fuzzy theory. Some brief of this theory is given below:

Suppose that the failure rates $\tilde{\lambda}_1$, $\tilde{\lambda}_2$, $\tilde{\mu}_1$ and $\tilde{\mu}_2$ can be represented by fuzzy sets. A fuzzy set $\tilde{A}$ with membership function $\eta_{\tilde{A}}$ is known as convex if $\eta_{\tilde{A}}(\alpha w_1 + (1-\alpha)w_2) \geq \min\{\eta_{\tilde{A}}(w_1), \eta_{\tilde{A}}(w_2)\}$ where $w_1, w_2 \in W$, W is universal set of $\tilde{A}$ and $\alpha \in [0, 1]$. Let $\eta_{\tilde{\lambda}_1}(x_1)$, $\eta_{\tilde{\lambda}_2}(x_2)$, $\eta_{\tilde{\mu}_1}(x_3)$ and $\eta_{\tilde{\mu}_2}(x_4)$ denote the membership functions of the failure rates $\tilde{\lambda}_1$, $\tilde{\lambda}_2$, $\tilde{\mu}_1$ and $\tilde{\mu}_2$, respectively. Now we have

$$\tilde{\lambda}_1 = \left\{\left(x_1, \eta_{\tilde{\lambda}_1}(x_1)\right) \middle| x_1 \in S\left(\tilde{\lambda}_1\right)\right\} \tag{11}$$

$$\tilde{\lambda}_2 = \left\{\left(x_2, \eta_{\tilde{\lambda}_2}(x_2)\right) \middle| x_2 \in S\left(\tilde{\lambda}_2\right)\right\} \tag{12}$$

$$\tilde{\mu}_1 = \left\{\left(x_3, \eta_{\tilde{\mu}_1}(x_3)\right) \middle| x_3 \in S\left(\tilde{\mu}_1\right)\right\} \tag{13}$$

$$\tilde{\mu}_2 = \left\{\left(x_4, \eta_{\tilde{\mu}_2}(x_4)\right) \middle| x_4 \in S\left(\tilde{\mu}_2\right)\right\} \tag{14}$$

where $S(\tilde{\lambda}_1), S(\tilde{\lambda}_2), S(\tilde{\mu}_1)$ and $S(\tilde{\mu}_2)$ are the support (i.e., the crisp collection of all non-zero memberships) of $\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\mu}_1$ and $\tilde{\mu}_2$, respectively. These supports denote the universal set of the failure rates.

Let $f(x, y, z)$ denote the availability of the embedded system with NVP. Since $\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\mu}_1$ and $\tilde{\mu}_2$, are the fuzzy numbers therefore $\tilde{f}(\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\mu}_1, \tilde{\mu}_2)$ is also a fuzzy measure. Now, according to Zadeh's extension principle, the membership function of performance measure $\tilde{f}(\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\mu}_1, \tilde{\mu}_2)$ is defined as

$$\eta_{\tilde{f}(\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\mu}_1, \tilde{\mu}_2)}(v) = \underset{\substack{x_1 \in X_1, \\ x_2 \in X_2, \\ x_3 \in X_3, \\ x_4 \in X_4}}{Sup} \min\left\{\eta_{\tilde{\lambda}_1}(x_1), \eta_{\tilde{\lambda}_2}(x_2), \eta_{\tilde{\mu}_1}(x_3), \eta_{\tilde{\mu}_2}(x_4) \middle| v = f(x_1, x_2, x_3, x_4)\right\} \tag{15}$$

Using equation (10), the membership function of availability is given by

$$\eta_{\tilde{A}V}(v) = \underset{\substack{x_1 \in X_1, \\ x_2 \in X_2, \\ x_3 \in X_3, \\ x_4 \in X_4}}{Sup} \min\left\{\eta_{\tilde{\lambda}_1}(x_1), \eta_{\tilde{\lambda}_2}(x_2), \eta_{\tilde{\mu}_1}(x_3), \eta_{\tilde{\mu}_2}(x_4) \middle| v = \tilde{A}V\right\} \tag{16}$$

where

$$AV = \frac{(\beta - x_1 - x_3)(x_2 x_4 \gamma)}{\beta\left[\beta x_1 x_2 x_4 \gamma + x_2 x_4 \gamma (x_1 + x_3) + \gamma x_1 x_4 + x_1 x_2 x_4 + x_3 x_2 \gamma + x_3 x_4 x_2\right]} \times n \tag{17}$$

Although fuzzy indices as given in equation (17) are theoretically valid and correct but some-times becomes very difficult to predict their shapes. This difficulty arises not due the numerical value of fuzzy theoretic approach but it occurs when it is tried to draw them graphically.

For the practical purpose we treat the failure rates as triangular fuzzy numbers. We take $\tilde{\lambda}_1 = \{a_1, a_2, a_3\}, \tilde{\lambda}_2 = \{b_1, b_2, b_3\}, \tilde{\mu}_1 = \{c_1, c_2, c_3\}$ and $\tilde{\mu}_2 = \{d_1, d_2, d_3\}$, respectively. Their membership functions are of triangular shape corresponding to the smallest possible, most promising and largest possible value. The triangular membership function is defined as

$$\eta_{\tilde{\lambda}_1}(x_1) = \begin{cases} \dfrac{x_1 - a_1}{a_2 - a_1}, & a_1 \le x_1 < a_2 \\ \dfrac{a_3 - x_1}{a_3 - a_2}, & a_2 < x_1 \le a_3 \\ 0 & , x_1 < a_1, x_1 > a_3 \end{cases} \tag{18}$$

Similarly the membership functions $\eta_{\tilde{\lambda}_2}(x_2), \eta_{\tilde{\mu}_1}(x_3)$ and $\eta_{\tilde{\mu}_2}(x_4)$ of $\tilde{\lambda}_2, \tilde{\mu}_1$ and $\tilde{\mu}_2$ are obtained by replacing b_i, c_i and d_i (i = 1, 2, 3), respectively in equation (18).

5 Numerical results

This section provides various numerical results for the total availability of the entire embedded system to simulate physical phenomena with the ever increasing demand in the computer technology. To illustrate the analytical results derived in the earlier sections, numerical illustration is given to demonstrate the effect of various parameters on the system performance measures. However, a good numerical method is required to accurately and quickly analyse the effect of various system parameters on the performance indices which are often needed for examining the behaviour of a queuing system. For this purpose, the coding of the programme has been done in 'MATLAB' software. Moreover, the default parameters of the system are chosen as $\lambda_1 = 0.4$, $\lambda_2 = 2.0$, $\mu_1 = 2.0$, $\mu_2 = 1.0$, $\gamma = 1.0$, $\beta = 4.0$ and $i = 6$ for Figures 2–5.

5.1 Effect of transition rates on total availability

Figure 2 shows that as the values of transition rate λ_1 increases the total availability of the entire embedded system decrease. Similar trend has been observed from Figure 3 for the increasing values of rate λ_2. Moreover, it can be easily observed form Figure 4 that total availability (AV) shows the decreasing trend for the increasing values of transition rate μ_1. From Figure 5, we observed that as the values of repair rate (β) takes the higher values, the total availability of the system increases which is quite observed. On the other hand, AV decreases as we increase the values of γ which can be observed from Figures 2–5 which tally with many real life situations.

Figure 2 Effect of λ_1 and γ on availability

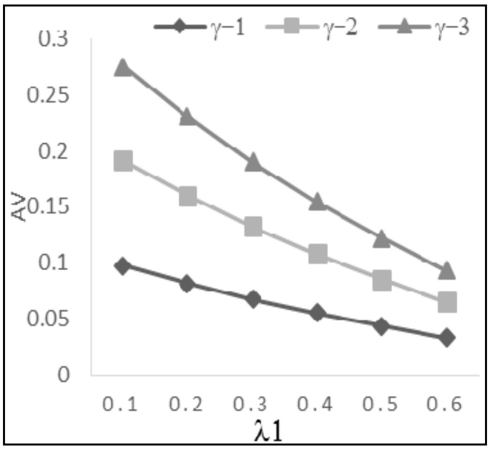


Figure 3 Effect of λ_2 and γ on availability

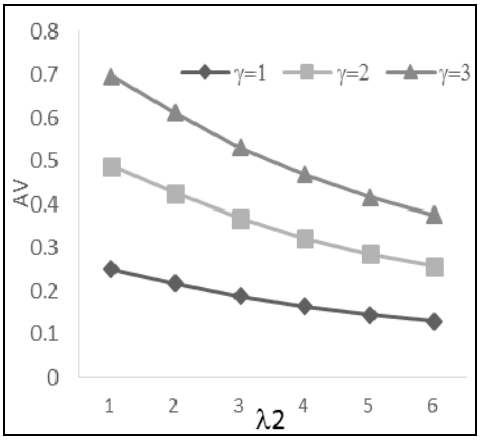


Figure 4 Effect of μ_1 and γ on availability

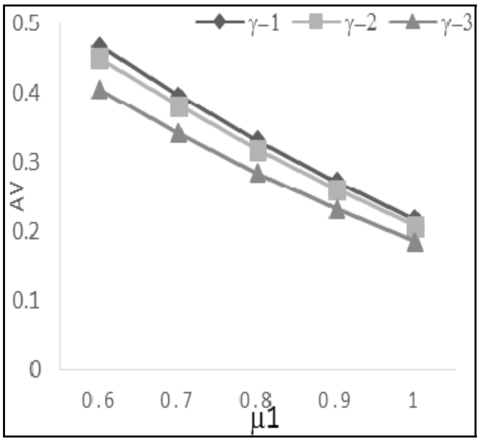
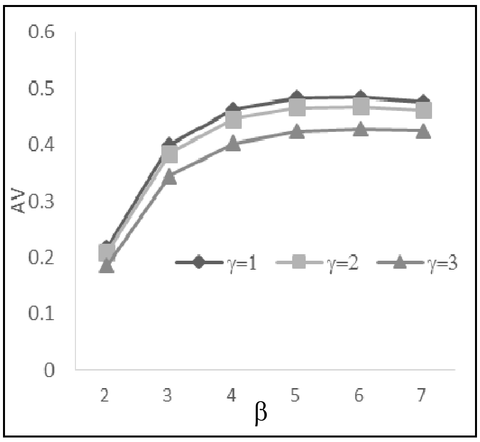


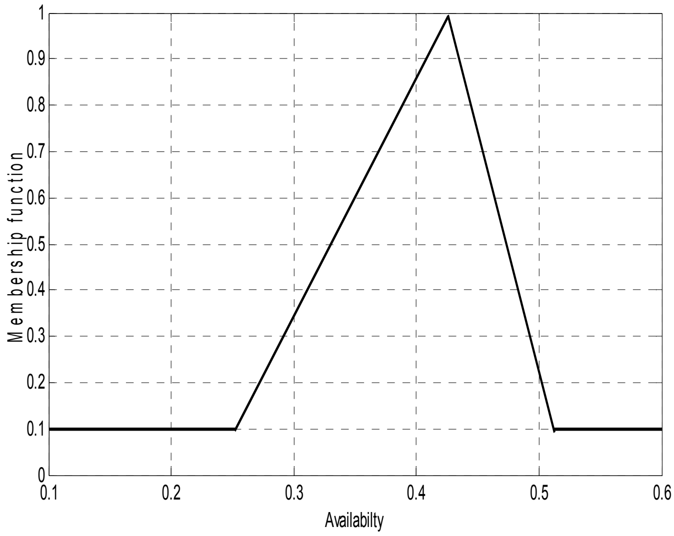
Figure 5 Effect of β and γ on availability



5.2 Effect of system parameters on membership function for availability

In this section the system parameters $\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\mu}_1$ and $\tilde{\mu}_2$ are considered as fuzzy numbers in order to find the system reliability indices. We fix γ =1.0 and β = 2.0. Then the availability is triangular fuzzy number. By setting $\tilde{\lambda}_1 = \{0.1, 0.2, 0.3\}$, $\tilde{\lambda}_2 = \{0.2, 0.3, 0.4\}$, $\tilde{\mu}_1 = \{2, 3, 4\}$ and $\tilde{\mu}_2 = \{3, 4, 5\}$, and then on fuzzyfying the parameters, we obtain the availability as triangular fuzzy number, i.e., $\tilde{A}V = \{0.2635, 0.4294, 0.5163\}$. The membership function graphs of availability is obtained and shown in Figure 6. The range of availability is approximately [0.2635, 0.5163] while the nominal value of ROCOF is 0.4294.

Figure 6 The membership function for the fuzzy availability



Overall, we concluded that the system availability can be increased up to certain label if the values of repair rate increases.

6 Conclusions

In this paper, an embedded system with N-version programming has been analysed in which software and hardware units are composed together with their standby units. Further, the concept of reboot is also taken into consideration. Also, we have done the fuzzy availability analysis of the proposed model. The availability of the embedded system is evaluated in the terms of the probabilities of subsystem of N-versions. Our assumptions considered in this paper will be helpful for improving the availability or reliability of embedded computer systems. Also, this work definitely be extended by taking different type of subversions in NVP that will give a new search direction to the system analyst and system designers.

References

Avizienis, A. and Chen, L. (1977) 'On the implementation of NVP for fault tolerance', *Proceeding of COMPSAC 77*, pp.149–155.

Avizienis, A. and Chen, L. (1978) 'N-version programming: a fault-tolerance approach to reliability of software operation', *Proceedings of FTCS 8*, pp.3–9.

Calikh, G. and Bener, A.B. (2013) 'Influence of confirmation biases of developers on software quality: an empirical study', *Software Quality Journal*, Vol. 21, No. 3, pp.317–416.

Downs, T. and Scott, A. (1992) 'Evaluating the performance of software reliability models', *IEEE Transactions on Reliability*, Vol. 41, No. 4, pp.532–538.

Elyezjy, N.T. (2012) 'Software quality assurance SQA- practices in Palestinian government', *International Journal of Computer Applications*, Vol. 58, No. 22, pp.1–4.

Jelinski, Z. and Moranda, P.B. (1972) 'Software reliability research', in Freiberger, W. (Ed.): *Statistical Computer Performance Evaluation*, pp.465–484, Academic Press, New York.

Keiller, P.A. and Miller, D.R. (1991) 'On the use and he performance of software reliability growth models', *Reliability Engineering & System Safety*, Vol. 32, Nos. 1–2, pp.95–117.

Min, X., Xiong, C. and Ng, S.H. (2013) 'A study of n-version programming and its impact on software availability', *International Journal of Systems Science*, pp.1–13.

Pando, H.D., Asensiz, S.C., Lima, R.S., Calderin, J.F. and Suarez, A.R. (2013) 'An application of fuzzy logic for hardware/software partitioning in embedded systems', *Computación y Sistemas*, Vol. 17, No. 1, pp.25–39.

Satoh, D. and Yamada, S. (2001) 'Discrete equations and software reliability growth models', *12th International Symposium on Software Reliability Engineering*, pp.176–184.

Shaout, A. and Trivedi, J. (2013) 'Performance appraisal system using a multistage fuzzy architecture', *International Journal of Computer and Information Technology*, Vol. 2, No. 3, pp.405–411.

Verdegay, J.L., Yager, R.R. and Bonissone, P.P. (2008) 'On heuristics as a fundamental constituent of soft computing', *Fuzzy Sets and Systems*, Vol. 159, No. 7, pp.846–855.

Wang, J.R. (2001) 'Ranking engineering design concepts using a fuzzy outranking preference model', *Fuzzy Sets and Systems*, Vol. 119, No. 1, pp.161–170.

Wattanapongsakorn, N. and Levitan, S.P. (2004) 'Reliability optimization models for embedded systems with multiple applications', *IEEE Transactions on Reliability*, Vol. 53, No. 3, pp.406–416.

Wu, Y., Wang, S. and Yu, Z. (2010) 'Embedded software reliability testing and its practice', *International Conference on Computer Design and Applications*, Vol. 2, pp.24–27.

Yamada, S., Tamura, Y. and Kimura, M. (2000) 'A software reliability growth model for a distributed development environment', *Electronics & Communication in Japan Part 3*, Vol. 83, No. 12, pp.1–8.

Yamakita, K., Yamada, H. and Kono, K. (2011) 'Phase-based reboot: reusing operating system execution phases for cheap reboot-based recovery', *Proceedings of IEEE/IFIP 41st International Conference on Dependable Systems & Networks*, pp.169–180.

Zhang, G., Ye, H. and Lin, Y. (2013) 'Quality attribute modeling and quality aware product configuration in software product lines', *Software Quality Journal*, Vol. 21, No. 1, pp.1–39.

Appendix

Construction of equations (1)–(7)

Under the steady-state conditions, the balance equations (1)–(7) are formulating by equating the expected rates of flow out of state to the expected rates of flow into that state as:

- *For state (0, 1):*

  The expected rate of flow out = $(\lambda_1 + \mu_1)$

  The expected rate of flow into state (0, 1) = β

To obtain equation (1), we equate the above two rates.

In the similar manner, other equations (2)–(7) for different states are constructed by equating the in-flow rate equals to out-flow rate.