Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: www.jscit.nit.ac.ir
DOI:
Volume 9, Number 2, Summer 2020, pp. 16-33
Received: 11/07/2018; Revised: 09/19/2019; Accepted: 10/24/2019

# A method for modeling and analysis of fault propagation in hybrid systems using stochastic activity networks

Arman Sanahmadi[1], Mohammad Abdollahi Azgomi[2*]
1- School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.
2*- School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.
[1]arman_sanahmadi@comp.iust.ac.ir, [2*]azgomi@iust.ac.ir

Corresponding author address: Mohammad Abdollahi Azgomi, Room 308, School of Computer Engineering, Iran University of Science and Technology, Hengam St., Resalat Sq., Tehran, Iran, Postal Code: 16846-13114.

**Abstract**- Hybrid systems consist of both continuous and discrete parts. These systems include several different components. A fault in one of these components can be activated and propagate to other components. Due to the advancement of technology and intelligent systems such as driverless car, health control devices and automated factories, the occurrence of a fault in one component and its propagation to other components can lead to financial and human-life losses. It is necessary to design a fault propagation model before construction of a system. With such a model, we can observe the propagation of the effects of a fault in a component to other components, before the construction of the system. It is also possible to identify critical components of the system. In this paper, a method for modeling fault propagation based on stochastic activity networks is presented. Based on this model, it is possible to identify the critical points of the system, the effect of different components on each other and the component failure behavior. The model has been applied to an aircraft fuel system, and its simulation and quantitative results are presented.

**Keywords**- Modeling, fault propagation, hybrid systems, stochastic activity networks (SANs), quantitative evaluation.

مجله علمی پژوهشی رایانش نرم و فن‌آوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: www.jscit.ac.ir
شناسگر اشیاء دیجیتالی (DOI) :
جلد ۹، شماره ۲، تابستان ۱۳۹۹، صفحه ۱۶-۳۳
دریافت: ۱۳۹۷/۰۸/۱۶ بازنگری: ۱۳۹۸/۰۶/۲۸ پذیرش:۱۳۹۸/۰۸/۰۲

# روشی برای مدل‌سازی انتشار خطا در سیستم‌های هیبرید با استفاده از شبکه‌های فعالیت تصادفی

آرمان سان احمدی[۱]، محمد عبداللهی ازگمی*[۲]
۱- دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران.
*۲- دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران.
[1]arman_sanahmadi@comp.iust.ac.ir, [2*]azgomi@iust.ac.ir

* نشانی نویسنده مسئول: محمد عبداللهی ازگمی، تهران، میدان رسالت، خیابان هنگام، دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر، اتاق ۳۰۴

**چکیده- سیستم‌های هیبرید از دو بخش پیوسته و گسسته تشکیل شده‌اند. این سیستم‌ها شامل چندین مؤلفه متفاوت هستند. وجود خطا در یکی از این مؤلفه‌ها و فعال شدن آن، می‌تواند به سایر مؤلفه‌ها انتشار پیدا کند. با توجه به پیشرفت تکنولوژی و به وجود آمدن سیستم‌های هوشمند هم‌چون ماشین خودران، دستگاه‌های کنترل سلامت و کارخانه‌های خودکار، وقوع خطا در یک مؤلفه و انتشار آن به سایر مؤلفه‌ها می‌تواند منجر به وقوع فاجعه و خسارات مالی و جانی فراوانی شود. بنابراین به منظور شناخت نقاط حساس سیستم و نحوه انتشار خطا بین مؤلفه‌ها، لازم است قبل از طراحی و بهره‌برداری از این سیستم‌ها، به مدل‌سازی انتشار خطا در آن ها پرداخت. در این مقاله روشی برای مدل‌سازی انتشار خطا بر اساس شبکه‌های فعالیت تصادفی ارائه شده‌است. بر اساس این مدل می‌توان به شناسایی نقاط حساس سیستم، تأثیر مؤلفه‌های مختلف بر روی همدیگر و رفتار خرابی مؤلفه‌ها پرداخت. مدل ارائه شده بر روی یک سیستم سوخت‌رسان هواپیما اعمال شده است و نتایج شبیه‌سازی و ارزیابی کمّی آن آورده شده است.**

**واژه‌های کلیدی:** مدل‌سازی، انتشار خطا، سیستم‌های هیبرید، شبکه‌های فعالیت تصادفی، ارزیابی کمّی

## ۱- مقدمه

سیستم‌های هیبرید[۱] از چندین مؤلفه فیزیکی و رایانشی تشکیل شده‌اند. این مؤلفه‌ها به منظور انجام هدف سیستم با همدیگر همکاری می‌کنند [۲,۱]. امروزه سیستم‌های هیبرید در حوزه‌های مختلفی مانند کنترل سلامت، حمل‌ونقل خودکار و کارخانه‌های صنعتی کاربرد فراوانی دارد. با توجه به کاربرد حساس این سیستم‌ها، وقوع خطا[۲] در یک بخش سیستم می‌تواند به سایر بخش‌ها انتشار[۳] پیدا کند و منجر به وقوع فاجعه گردد. در هر یک از این مؤلفه‌ها می‌تواند یک یا چندین خطا وجود داشته باشد. وجود خطا در هر یک از این مؤلفه‌ها می‌تواند به دلیل نقص فنی، فرسودگی مؤلفه‌ها، عمدی یا غیرعمدی باشد [۳].

چگونگی انتشار خطا در بین مؤلفه‌ها و بررسی پارامترهای مختلف اتکاپذیری، در محیط آزمایشگاهی و واقعی، هزینه‌بر و غیرممکن است [۵,۴]. از طریق مدل انتشار خطا می‌توان به مشاهده پارامترهای مختلفی از جمله رفتار خرابی مؤلفه‌ها در حضور و عدم حضور خطا در یک مؤلفه دیگر، قابلیت اطمینان[۴] کل سیستم و شناسایی مؤلفه‌های حساس سیستم پرداخت. در اینجا منظور از مؤلفه‌های حساس سیستم، مؤلفه‌هایی هستند که در صورت خرابی

منجر به توقف کامل سیستم می‌شوند. می‌توان قبل از طراحی سیستم‌ها با مدل‌سازی انتشار خطا به بررسی مناسب بودن پارامترهای قابلیت اطمینان، ایمنی و امنیت قطعات بخش‌های مختلف سیستم پرداخت. با استفاده از نتایج مدل‌سازی می‌توان مشخص کرد چه نوع مؤلفه‌هایی برای ساخت این سیستم مناسب هستند [۶].

در کارهای انجام‌گرفته معمولاً معیارهای کافی جهت مدل‌سازی انتشار خطا در نظر گرفته نشده است. همچنین منطق انتشار خطا به درستی پیاده نشده است،که با توجه به پیچیدگی سیستم‌ها این مدل‌ها نمی‌توانند دقیق باشند. مدل ارائه شده باید دارای ویژگی‌هایی همچون مقیاس‌پذیری[۵] باشد که بتوان علاوه بر مدل‌سازی انتشار خطای یک سیستم به صورت جداگانه، رفتار انتشار خطای سیستم‌های مختلف هنگامی‌که با همدیگر همکاری می‌کنند را مشاهده کرد. همچنین مدل ارائه‌شده باید پویا باشد. مشکل مدل‌های ایستا شناسایی مسیرهای انتشار خطا در سیستم است که در بسیاری از مواقع هزینه‌بر و غیرممکن است. همچنین مدل باید معیارهای کافی همچون وابستگی‌های مستقیم و غیرمستقیم موجود در بین مؤلفه‌های سیستم را در نظر بگیرد.

در روش پیشنهادی این مقاله ابتدا بر اساس همبندی[۶] و مشخصات سیستم، گراف داده‌ای، گراف غیرداده‌ای و گراف ترتیب اجرایی مؤلفه‌های سیستم استخراج می‌شود. همچنین عبارت‌های احتمالی مربوط به رفتار هر یک از مؤلفه‌ها استخراج می‌گردد. سپس بر اساس این سه گراف و عبارت‌های احتمالی، مدل شبکه‌های فعالیت تصادفی[۷] هر یک از مؤلفه‌ها با استفاده از قوانین تبدیلی که ارائه خواهد شد استخراج می‌گردد. در نهایت با استفاده از صورت‌بندی تکرار-الحاق[۸]، مؤلفه‌های مختلف سیستم با همدیگر الحاق می‌شوند و مدل نهایی سیستم به دست خواهد آمد. به منظور ارزیابی سیستم، با تزریق خطا در بخش‌های مختلف سیستم و فعال‌سازی آن می‌توان به بررسی و مشاهده رفتار سایر مؤلفه‌ها در حضور و عدم حضور خطا و به دست آوردن قابلیت اطمینان کل سیستم پرداخت.

## ۲- مفاهیم پایه

در این بخش مفاهیم پایه‌ای که در حوزه این مقاله لازم است، توضیح داده شده‌است.

### ۲-۱- خطا، اشکال و خرابی

خطا به نقصی گفته می‌شود که در داخل سیستم وجود دارد. هر خطا می‌تواند فعال گردد و به یک اشکال تبدیل شود. خطا می‌تواند عوامل مختلفی از جمله: مشخصه‌سازی نادرست سیستم، طراحی نادرست، استفاده از ابزار نامناسب و عامل‌های بیرونی داشته باشد. اشکال به خطایی گفته می‌شود که فعال شده باشد. درواقع علت یک اشکال، همان خطا است. اشکال بخشی از وضعیت کل سیستم است که می‌تواند منجر به خرابی کل سیستم شود. حالت سیستم مجموعه‌ای از حالت‌های تک تک مؤلفه‌ها است. هنگامی‌که یک اشکال درون یک مؤلفه وجود دارد تا زمانی که آن اشکال به یک حالت خارجی آن مؤلفه انتقال پیدا نکند نمی‌تواند باعث خرابی کل سیستم شود [۷].

یک سیستم دچار خرابی شده است هرگاه خدمات مربوط به این سیستم از حالت صحیح آن انحراف داشته باشد. درواقع یک اشکال در صورت انتشار به محیط خارج از مؤلفه باعث ایجاد خرابی می‌شود. آسیب‌شناسی[۹] خرابی که نحوه ارتباط خطا، اشکال و خرابی را مشخص می‌کند در شکل ۱ آورده شده است [۷].


**شکل ۱: آسیب‌شناسی خرابی [۷].**

### ۲-۲- شبکه‌های فعالیت تصادفی

شبکه فعالیت تصادفی یک توسعه از شبکه‌های پتری است. شبکه‌های فعالیت تصادفی به‌منظور ارزیابی کارایی و اتکاپذیری سیستم‌ها در دهه ۸۰ گسترش داده شده است [۸]. هر شبکه فعالیت تصادفی یک ۱۲-تایی به‌صورت $SAN = (P, A, I, O, \gamma, \tau, i, o, \mu_0, C, F, G)$ است که تعریف هر یک از این نمادها عبارتند از:

- **P**: مجموعه متناهی از مکان‌ها.
- **A**: مجموعه‌ای متناهی از فعالیت‌ها.
- **I**: مجموعه متناهی از دروازه‌های ورودی.
- **O**: مجموعه متناهی از دروازه‌های خروجی.
- $\boldsymbol{\gamma}$: نشان‌دهنده تعداد اقدام‌های احتمالی متفاوت برای هر فعالیت است.
- $\boldsymbol{\tau}$: مشخص‌کننده نوع هر فعالیت است (فوری یا زمانی).
- **i**: نگاشت‌کننده دروازه‌های ورودی به فعالیت‌ها است.
- **o**: نگاشت‌کننده دروازه‌های خروجی به فعالیت‌ها است.
- $\boldsymbol{\mu_0}$: نشانه‌گذاری اولیه است که باید پایدار باشد.
- **C**: تابع تخصیص توزیع‌های احتمالی مربوط به اقدام‌های احتمالی فعالیت‌ها است.
- **F**: تابع تخصیص نرخ‌های احتمالی فعالیت‌ها است.
- **G**: تابع تخصیص فعال‌سازی مجدد فعالیت‌ها است.

## ۳- کارهای مرتبط

در این بخش به بررسی کارهایی که تاکنون در این زمینه انجام شده است پرداخته می‌شود.

### ۳-۱- مدل‌های با دید گرافی یا شبکه‌ای به سیستم

در کار شماره [۹] به مدل‌سازی انتشار خطا بین زیرساخت‌های حیاتی و وابسته به هم پرداخته شده است. بسیاری از زیرساخت‌های حیاتی به یکدیگر وابسته هستند. این وابستگی‌ها بین زیرساخت‌ها اجتناب‌ناپذیر است و برای عملکرد صحیح کل شبکه نیاز است. با توجه به این وابستگی، وقوع خطا در یک زیرساخت می‌تواند انتشار پیدا کند و سایر زیرساخت‌های مرتبط را تحت تأثیر قرار دهد. این انتشار می‌تواند منجر به وقوع فاجعه‌های بزرگ‌تری شود. به‌طور مثال در سال ۲۰۰۳ وقوع یک خطا در شبکه توزیع برق ایتالیا منجر به خاموشی گسترده و قطعی زیرساخت اینترنت ایتالیا شد. در این مقاله یک چارچوب زنجیره مارکوف[۱۰] با وابستگی متقابل ارائه شده است. این چارچوب، یک چارچوب احتمالی برای بررسی و ارزیابی تأثیر وجود خطا و رخ دادن خرابی بین زیرساخت‌های حیاتی است. هدف آن مدل‌سازی نحوه انتشار خرابی و بررسی تأثیر آن بر روی قابلیت اطمینان کل سیستم است. زنجیره مارکوف با وابستگی متقابل به این صورت است که ابتدا زنجیره مارکوف هر زیرساخت کشیده می‌شود سپس به روش خاصی این زنجیره مارکوف‌ها در کنار یکدیگر قرار می‌گیرند. زنجیره مارکوف حاصل‌شده از کنار قرار گرفتن این دو زنجیره مارکوف نیز ارگودیک است. این چارچوب بیشتر برای شبکه‌های زیرساختی مثل شبکه‌های توزیع برق، گاز و آب مناسب است و در آن تمامی گره‌های موجود در زیرساخت یکسان فرض می‌شود و تفاوتی از لحاظ ساختاری بین مؤلفه‌های رایانشی و فیزیکی در نظر گرفته نشده است. برای ارزیابی مدل ارائه شده به بررسی تعداد خطوط خراب‌شده در شبکه توزیع برق و گستردگی قطعی در شرایط مختلف پرداخته شده است. همین افراد در مقاله [۱۰] به بررسی انتشار خطا همراه با در نظر گرفتن تأثیر خطای انسانی پرداخته‌اند.

در مقاله [۱۱] همانند مقاله قبلی به بررسی انتشار خطا در سیستم‌های شبکه‌ی توزیعی پرداخته شده است. ازآن‌جهت که سیستم‌های هیبرید دارای دو زیرشبکه شامل مؤلفه‌های رایانشی و فیزیکی است، برای مدل‌سازی با این روش به دو زیرشبکه با نام A و B نیاز است. ابتدا بر اساس همبندی سیستم ارتباط بین گره‌های A و B کشیده می‌شود. سپس هنگامی‌که یک گره از یکی از زیرشبکه‌ها خراب می‌شود تمامی یال‌هایی که از آن گره به سایر گره‌های زیرشبکه داخلی یا خارجی وصل شده است، حذف می‌گردد. پس از حذف یال‌های مربوطه، هر یک از زیر شبکه‌ها را می‌توان به چندین خوشه تبدیل کرد. در اینجا منظور از خوشه مجموعه گره‌هایی است که فقط با خود ارتباط دارند و با سایر گره‌های موجود در زیرشبکه خود ارتباطی ندارند. این فرایند آن‌قدر ادامه پیدا خواهد کرد تا شبکه به یک حالت پایدار برسد و دیگر نتوان گره یا یالی را حذف کرد. مشکلی که این روش دارد این است که حذف گره‌ها و ارتباط آن با سایر گره‌ها بر اساس یک منطق ساده است و ممکن است یک گره در سیستم‌های پیچیده هیبرید صرفاً به خاطر از دست دادن ارتباط خود با یک گره خاص کارایی خود را از دست ندهد و هنوز بتواند در شبکه سیستم فعالیت کند. البته این نحوه‌ی مدل‌سازی بیشتر برای سیستم‌های هیبرید تورین مثل شبکه‌های توزیع برق و انرژی کاربرد دارد

در مقاله [۱۲] روشی برای ارزیابی انتشار خطا در سیستم‌های سایبر-فیزیکی ارائه شده است. در این روش مانند روش قبلی رو زیر شبکه در نظر گرفته شده است. تمامی مؤلفه‌های هر شبکه یکسان فرض شده است. در این روش هنگامی‌که یک مؤلفه دچار خرابی می‌شود گره مربوط به آن حذف می‌شود. بعد از حذف یال‌های مربوط به گره حذف‌شده درصورتی‌که یک گره‌ای ایجاد شود که به هیچ مؤلفه‌ای وصل نباشد آن گره نیز حذف خواهد شد. این کار ادامه پیدا خواهد کرد تا سیستم به یک حالت پایدار برسد. روش ارزیابی مدل به‌صورت قابلیت اطمینان مرتبه k است. قابلیت اطمینان مرتبه k به این معنا است که درصورتی‌که یکی از شبکه‌های سایبری یا فیزیکی بیشتر از k گره سالم داشته باشد آن شبکه می‌تواند به عملکرد خود ادامه دهد. یکی از مشکلات این روش این است که پیچیدگی سیستم‌های سایبر-فیزیکی[۱۱]، تفاوت مؤلفه‌های آن‌ها و ارتباط پیچیده آن‌ها مخصوصاً ارتباط بین مؤلفه‌های سایبری و فیزیکی درست دیده نشده و مدل بسیار سطح بالا و دور از واقعیت است.

در مقاله [۱۳] به ارزیابی انتشار خرابی در سیستم‌های سایبر-فیزیکی پرداخته شده است. این مقاله برخلاف مقاله قبلی بیشتر بر روی نحوه‌ی ارتباط گره‌های سایبری و فیزیکی کارکرده است و ارتباط آن‌ها را صرفاً وجود یال در بین آن‌ها را در نظر نگرفته است. در این روش یک قانونی در نظر گرفته شده است که هر گره محاسباتی باید توسط یک گره فیزیکی پشتیبانی شود، یعنی یک گره محاسباتی در صورتی خراب فرض می‌شود که ارتباط خود را با بخش فیزیکی از دست بدهد. اما هر گره محاسباتی می‌تواند توسط چندین گره فیزیکی پشتیبانی شود و درصورتی‌که ارتباط خود را با بخش محاسباتی از دست بدهد عملکرد خود را از دست می‌دهد. نحوه‌ی انتشار خرابی نیز در این مدل با روش‌های دیگر متفاوت

است. در این مدل یک گره به‌صورت تصادفی حذف خواهد شد و تمامی لینک‌های خارجی و داخلی آن حذف می‌شود سپس بزرگ‌ترین زیرشبکه را در هریک از شبکه‌های سایبری و فیزیکی در نظر گرفته می‌شود و سایر زیرشبکه‌ها نیز حذف می‌گردد. این زیرشبکه‌ها گره‌های در معرض خطر نامیده می‌شود. در کار [۱۴] به بررسی وابستگی و انتشار خرابی در سیستم‌های تورین بر اساس نظریه شبکه‌های پیچیده پرداخته شده است.

### ۳-۲- مدل‌های با سطح انتزاع مؤلفه

در مقاله [۶] روشی جدید برای ارزیابی هم‌زمان ایمنی و امنیت سیستم‌های سایبر-فیزیکی در قالب درخت خطا حالت/رخداد ارائه شده است. از این مدل برای مشاهده چگونگی انتشار خطا در سیستم‌های پیچیده استفاده شده است. برخلاف روش‌های دیگر که همه‌ی مؤلفه‌ها را یکسان فرض می‌کردند، این روش هر مؤلفه را به‌صورت جداگانه و همراه با ویژگی‌های خاص آن طراحی نموده است. ویژگی‌های مختلف هر مؤلفه مانند نرخ خرابی و نرخ تعمیر در این مدل آورده شده است. ارتباط مؤلفه‌ها در این روش از طریق رخدادهای متفاوت بین آن‌ها است. این رخدادها از طریق درگاه‌های موجود بر روی مؤلفه‌ها انتقال پیدا می‌کنند. مشکلی که این روش دارد این است که همانند درخت خطا باید رخداد نهایی مشخص باشد و فقط مشاهده کرد که چه رخدادهایی پشت سر هم اتفاق خواهند افتاد که منجر به وقوع آن خواهند شد و ساختار مدل یک ساختار کاملاً ایستا است.

در مقاله [۱۵] روشی برای مدل‌سازی انتشار خطا در سیستم‌های هیبرید ارائه شده است. در این روش از آتاماتای هیبرید برای مدل‌سازی فرایند انتشار خطا استفاده شده است. هر مؤلفه به‌صورت مجزا همراه با تمام جزییات مدل‌سازی می‌شود. در این روش چهار نوع وابستگی شامل وابستگی فیزیکی، سایبری، جغرافیایی و منطقی در نظر گرفته شده است.

مدل هر یک از مؤلفه‌ها طراحی می‌شود. سپس با خراب کردن یک مؤلفه نشان داده می‌شود که این خرابی چگونه در سطح سیستم انتشار پیدا می‌کند. با استفاده از این روش می‌توان به مطالعه انتشار خطا به دلیل وابستگی‌های مختلف در سیستم، ارزیابی آسیب‌پذیری سیستم و توسعه راهبردهای[۱۲] نگه‌داری

مشکل اصلی این روش جزییات فراوان مؤلفه‌ها است. بنابراین طراحی و ساخت این مدل پیچیده می‌شود. عملاً باید کل سیستم مورد مطالعه ازنظر ساختاری و رفتاری مدل‌سازی کرد.

در مقاله‌های [۱۶, ۱۷] روشی برای مدل‌سازی انتشار اشکال در سیستم‌های مکاترونیک ارائه شده است. سیستم‌های مکاترونیک معمولاً غیرهمگن و دارای مؤلفه‌های متفاوت نرم‌افزاری و فیزیکی هستند. در این مقاله نشان داده شده است برای اینکه بتوانیم یک مدل صحیح انتشار اشکال در این سیستم‌ها داشته باشیم باید جریان داده و جریان کنترل به‌صورت جداگانه دیده شود. ایده اصلی این روش بر روی دو گراف کنترل داده و جریان داده قرار گرفته است. این دو گراف در هر فازی از طراحی سیستم می‌توانند تولید شوند و انتشار اشکال ارزیابی شود.

در مقاله [۱۸] به ارزیابی پویای ایمنی سیستم‌های پیچیده پرداخته شده است. مؤلفه‌های موجود در سیستم‌های پیچیده پویا دارای وابستگی‌های وابسته به زمان و عملکردی هستند. بنابراین برای مشاهده رفتار شکست یک سیستم باید علاوه بر مشاهده چگونگی ترکیب شکست مؤلفه‌ها، ترتیب زمانی آن‌ها نیز مشاهده شود. در این مقاله یک روش ارزیابی مبتنی بر مدل درخت خطاهای زمانی با استفاده از ترکیب Hip-hops با pandora ارائه شده است. این دو روش در مقالات [۱۹, ۲۰] توضیح داده شده است. برای مدل‌سازی با این روش سه فاز مختلف باید به ترتیب انجام شود. در فاز اول مدل سیستم و نحوه‌ی خرابی آن باید مشخص سپس ارتباط مؤلفه‌ها و چگونگی قرار گرفتن آن‌ها در کنار یکدیگر تعیین گردد. درنهایت وابستگی‌های ترتیبی باید در مدل اضافه شود. بعد از مشخص کردن معماری مؤلفه‌ها و عبارت‌های خرابی، با استفاده از Hip-Hops درخت خطای زمانی استخراج می‌شود. سپس این درخت زمانی خطا به مدل شبکه پتری یا شبکه بیزی تبدیل می‌شود و بر روی آن تحلیل‌های لازم انجام می‌گردد. این کار یکی از کارهای نوین و خوبی است که در این زمینه انجام شده است. مشکلی که این مدل دارد این است که همانند درخت خطا تمامی رخدادهای خطا را باید مشخص کرد و حتی ترتیب آن‌ها را هم باید در نظر گرفت. تمامی رخدادها در این مدل مربوط به ورودی و خروجی است و نرخ آن‌ها ثابت است. این روش در مقاله [۲۱] ارزیابی شده است.

در مقاله [۲۲] روشی برای تولید خودکار مدل انتشار خطا ارائه شده است. در این مقاله ابتدا با مشخص کردن نحوه‌ی خرابی مؤلفه‌ها در برابر خطاهای گوناگون، حاشیه‌نویسی خرابی را برای هر مؤلفه ایجاد می‌کند. سپس با استفاده از این حاشیه‌نویسی‌ها درخت خطای مؤلفه‌ها تولید می‌شوند که می‌تواند برای تحلیل انتشار خطا استفاده گردد. مشکلی که این مقاله دارد این است که ساختار آن ایستا است و این مدل برای ارزیابی کمّی مناسب نیست.

در مقاله [۲۳] به ارزیابی انتشار خطاهای گذرا در سیستم‌های کنترل شبکه پرداخته شده است. از مشکلات این مدل سختی پیاده‌سازی و وارد کردن جزییات هر مؤلفه است.

در مقاله [۲۴] روشی برای ارزیابی ریسک با توجه به انتشار خطا در سیستم‌های سایبر-فیزیکی پرداخته شده است. در این روش

پارامترهای مختلفی ازجمله نقش انسان در آن در نظر گرفته شده است مقایسه کارهای انجام‌شده در این حوزه در جدول ۱ آورده شده است.

**جدول ۱: مقایسه کارهای مرتبط**

| شماره مقاله | کاربرد | مدل | تحلیل | یکسان درنظر گرفتن گره‌ها |
|---|---|---|---|---|
| [۹][۱۰] | زیرساخت‌های حیاتی | مارکوف | کمّی | بله |
| [۱۱] | زیرساخت‌های حیاتی | شبکه‌ای (گرافی) | کمّی | بله |
| [۱۲][۱۳][۱۴] | سایبر-فیزیکی | شبکه‌ای (گرافی) | کمّی | بله |
| [۷] | سیستم‌ها هیبرید | درخت خطا | کمّی | خیر |
| [۱۵] | سیستم‌ها هیبرید | آتاماتای هیبرید | کمّی | خیر |
| [۱۶][۱۷] | مکاترونیک | پتری | کمّی | خیر |
| [۱۸][۱۹][۲۰] [۲۱] | سیستم‌های پیچیده | درخت خطا | کمّی | خیر |
| [۲۲] | سیستم‌های پیچیده | درخت خطا | کیفی | خیر |
| [۲۳] | سیستم‌های کنترلی | درخت خطا | کمّی | خیر |
| [۲۴] | سیستم‌های سایبر-فیزیکی | ارزیابی ریسک | کمّی | خیر |

## ۴- روش پیشنهادی

در این روش ساختار مدل بر اساس سه گراف شکل می‌گیرد که این گراف‌ها و نحوه‌ی تبدیل آن‌ها به مدل در ادامه توضیح داده شده است. هر مؤلفه و نحوه‌ی رفتار آن در برابر خطاهای مختلف به‌صورت جداگانه مدل‌سازی می‌شود. درصورتی‌که از قوانین تبدیل گراف‌ها به ساختار مدل که در ادامه توضیح داده خواهد شد پیروی شود، تمامی مؤلفه‌ها به‌راحتی در کنار یکدیگر قرار می‌گیرند و مدل نهایی شکل می‌گیرد. برای ارزیابی مدل پس از ساخت آن با توجه به نوع آزمایش، در یک یا چندین مؤلفه خطاهایی تزریق می‌شود و نحوه‌ی انتشار آن در کل سیستم و تأثیری که بر روی مؤلفه‌های دیگر می‌گذارد، مورد ارزیابی قرار می‌گیرد.

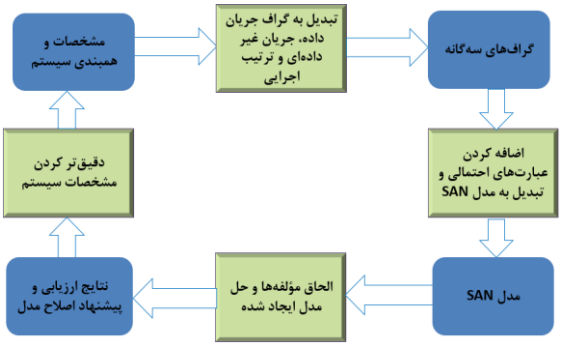
### ٤-١- فرایند پیشنهادی برای مدل‌سازی انتشار خطا

فرآیند پیشنهادی راه‌حلی برای مدل‌سازی انتشار خطا در سیستم‌های هیبرید را ارائه می‌کند. در این روش برخلاف بیشتر روش‌های بررسی‌شده در بخش مروری بر کارهای پیشین تمامی مؤلفه‌های سیستم یکسان در نظر گرفته نمی‌شوند. با توجه به اینکه در سیستم‌های هیبرید ساختار هر مؤلفه در فرایند انتشار خطا تأثیری فراوانی دارد، برای داشتن مدلی دقیق نمی‌توان به ساختار مؤلفه‌ها توجه نکرد و همه‌ی گره‌ها را یکسان فرض کرد. سیستم‌های هیبرید از مؤلفه‌های رایانشی و فیزیکی تشکیل شده است. مؤلفه‌های رایانشی دارای نرخ خرابی نیستند و فقط ورودی‌های غیرمتعارف مؤلفه، می‌تواند باعث خرابی آن شود.

چرخه پیشنهادی روش مدل‌سازی انتشار خطا به این صورت است که با توجه به شکل ۲ ابتدا با استفاده از مشخصات و همبندی سیستم، سه گراف اولیه از سیستم تهیه می‌شود که این گراف‌ها عبارت‌اند از:

- **گراف وابستگی داده‌ای:** این گراف مشخص می‌کند هر مؤلفه‌ای از چه مؤلفه‌هایی ورودی می‌گیرد و به کدام مؤلفه‌ها خروجی می‌فرستد.
- **گراف وابستگی غیرداده‌ای:** این گراف مشخص می‌کند که خرابی یک مؤلفه بر روی کدام مؤلفه‌های دیگر تأثیرگذار خواهد بود. این وابستگی به خاطر ورودی و خروجی ناصحیح نیست بلکه به خاطر حضور یا عدم حضور یک مؤلفه است.
- **گراف ترتیب اجرایی:** این گراف مشخص می‌کند که در صورت خرابی یا اتمام کار یک مؤلفه چه مؤلفه‌ای شروع به کار می‌کند.

در ادامه باید عبارت‌های احتمالی مربوط به هر یک از مؤلفه‌ها، استخراج شود. این عبارت‌ها همانند عبارت‌های مدل احتمالی FPTA[۱۳] است که مشخص می‌کند درصورتی‌که ورودی‌های مؤلفه به یک صورت خاص باشند با چه احتمالی، خروجی چگونه خواهد بود. این عبارت‌ها در ادامه توضیح داده خواهد شد.




**شکل ۲: فرایند پیشنهادی مدل‌سازی انتشار خطا.**

این گراف‌ها و عبارت‌های احتمالی از طریق قوانین تبدیلی که در ادامه توضیح داده می‌شود به مدل SAN تبدیل می‌گردد. سپس مدل مؤلفه‌ها از طریق صورت‌بندی تکرار-الحاق به همدیگر الحاق می‌شوند. درنهایت مدل به‌دست‌آمده حل می‌شود و نتایج موردنظر استخراج می‌شود. با بررسی نتایج ممکن است مشخص شود که مشخصات سیستم درست تعیین نشده است. در این حالت باید در جزییات سیستم بازنگری صورت بگیرد و فرایند مدل‌سازی تکرار شود.

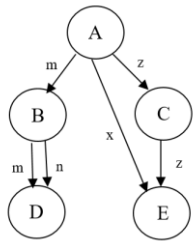
### ٤-۲- گراف وابستگی داده‌ای

اولین ارتباطی که مؤلفه‌های یک سیستم باهمدیگر دارند، ارتباط ورودی و خروجی است. بنابراین وجود گرافی که نحوه ارتباط مؤلفه‌ها را مشخص کند لازم و ضروری است. گراف وابستگی داده‌ای را می‌توان از همبندی شبکه به دست آورد این گراف مشخص می‌کند که هر یک از مؤلفه‌های سیستم به کدام یک از مؤلفه‌های دیگر داده می‌فرستد و از کدام مؤلفه‌ها ورودی می‌گیرد. درواقع ساده‌ترین وابستگی که بین مؤلفه‌ها وجود دارد این نوع وابستگی است و جزء بدیهی‌ترین پارامترهایی است که باید در مسئله انتشار خطا در نظر گرفته شود. این گراف درواقع همان ساختار همبندی سیستم است. هر مؤلفه می‌تواند چندین ورودی و خروجی داشته باشد. درصورتی‌که یک مؤلفه چندین ورودی برای یک مؤلفه دیگر فراهم کند باید به تعداد ورودی‌ها میان آن‌ها یال کشیده شود. به‌طوری‌که هر یال نشان‌دهنده یک ارتباط است. شکل 3 یک نمونه از گراف وابستگی داده‌ای را نشان می‌دهد. تعریف گراف وابستگی در فرمول ۱ آورده شده است.

$$\begin{matrix} G^{DF}=[N,D^{DF}] \\ N=\{n_1,\dots,n_n\} \\ D^{DF}=\{(n_i,n_j,V_{i,j}),\dots,(n_k,n_l,V_{k,l})\} \end{matrix} \quad (۱)$$

N مجموعه‌ای است که نشان‌دهنده رأس‌های گراف است. هر یک از این رأس‌ها، یک مؤلفه سیستم است. یال‌های بین آن‌ها که با مجموعه $D^{DF}$ مشخص‌شده است ارتباط ورودی و خروجی مؤلفه‌ها است.

با توجه به شکل ۳ می‌توان نتیجه گرفت که ورودی مؤلفه B، C، E توسط مؤلفه A تأمین می‌شود و مؤلفه B دو ورودی برای D ارسال می‌کند همچنین ورودی E از طریق مؤلفه C تأمین می‌شود.




**شکل ۳: یک نمونه گراف وابستگی داده‌ای.**

### ٤-۳- گراف وابستگی غیرداده‌ای

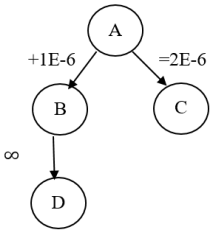
مؤلفه‌های یک سیستم ممکن است به صورت غیرمستقیم بر روی همدیگر تأثیر بگذارند. و این وابستگی بر روی مدل‌سازی انتشار خطا تأثیرگذار خواهد بود. گراف وابستگی غیرداده‌ای مشخص می‌کند که چه مؤلفه‌هایی به یکدیگر وابستگی دارند. این وابستگی از نوع وابستگی داده‌ای بخش قبل نیست. در اینجا منظور از وابستگی تأثیری است که حضور یا عدم حضور یک مؤلفه بر نرخ خرابی یک مؤلفه دیگر می‌گذارد. به‌طور مثال خرابی مؤلفه خنک‌کننده باعث می‌شود نرخ خرابی یک مؤلفه دیگر بالاتر رود. در این گراف هر گره در صورت وجود وابستگی تنها یک یال به گره دیگر دارد. یک نمونه از گراف وابستگی غیرداده‌ای در شکل ۴ نشان داده شده است. این نوع وابستگی پارامتری است که در کارهای پیشین دیده نشده است. درحالی‌که پارامتری مهمی بوده و در بسیاری از سیستم‌ها این وابستگی غیرمستقیم بین مؤلفه‌ها وجود دارد. هر مؤلفه‌ای داخل سیستم دارای یک نرخ خرابی است. عددی که بر روی یال‌های گراف وابستگی غیرداده‌ای وجود دارد، اگر بدون علامت ریاضی باشد به این معنا است که درصورت خرابی مؤلفه پدر، نرخ خرابی گره فرزند با مقدار روی یال جمع می‌شود.

همچنین درصورتی‌که قبل از عدد روی یال گراف، علامت مساوی«=» قرار داده شده باشد. به این معنا است که نرخ خرابی مؤلفه برابر با آن مقدار می‌شود. اگر بر روی یال گراف علامت بی‌نهایت قرار داده شده باشد به این معنا است که در صورت خرابی مؤلفه پدر، مؤلفه فرزند آن فوراً خراب خواهد شد. همچنین درصورتی‌که علامت «*» وجود داشته باشد به این معنا است که نرخ خرابی در عدد نوشته‌شده ضرب خواهد شد.

تعریف گراف وابستگی غیرداده‌ای در فرمول ۲ نشان داده شده است.

$$\begin{matrix} G^{NDF}=[N,D^{NDF}] \\ N=\{n_1,\dots,n_n\} \\ D^{NDF}=\{(n_i,n_j,v_{i,j}),\dots,(n_k,n_l,v_{k,l})\} \end{matrix} \quad (۲)$$

با توجه به شکل ۴ هنگامی‌که مؤلفه A خراب شود، نرخ خرابی مؤلفه B برابر با مقدار 2E-6 و نرخ خرابی مؤلفه C به اندازه‌ی 2E-6 افزایش پیدا خواهد کرد. همچنین در صورت خرابی مؤلفه B مؤلفه D فوراً دچار خرابی می‌گردد.



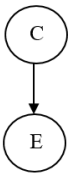

**شکل ۴: یک نمونه گراف وابستگی غیرداده‌ای.**

### ٤-٤- گراف ترتیب اجرایی

گراف ترتیب اجرایی مشخص می‌کند که بعد از خرابی یک مؤلفه چه مؤلفه‌هایی باید کار خود را شروع می‌کنند. در سیستم‌های هیبرید معمولاً افزونگی مؤلفه وجود دارد تا در صورت خرابی یک مؤلفه، مؤلفه دیگری کار خود را شروع کند و جایگزین آن شود این

نوع ترتیب اجرایی با یک گراف نشان داده خواهد شد. در شکل ۵ یک گراف ترتیب اجرایی کوچک نشان داده شده است. شکل ۵ نشان می‌دهد که کار مؤلفه E در صورتی شروع می‌شود که مؤلفه C خراب شود. در فرمول ۳ تعریف گراف ترتیب اجرایی نشان داده شده است.

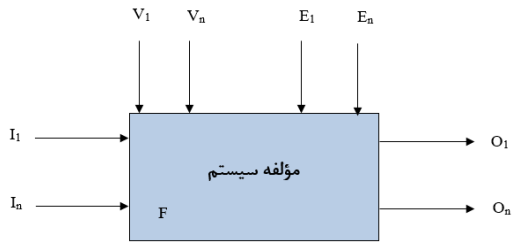
$$\begin{matrix} G^{CF}=[N,D^{CF}] \\ N=\{n_1,\dots,n_n\} \\ D^{CF}=\{(n_i,n_j),\dots,(n_k,n_l)\} \end{matrix} \quad (۳)$$




**شکل ۵: یک نمونه گراف ترتیب اجرایی.**

### ٤-٥- رابط‌های مؤلفه‌های سیستم

شکل ۶ رابط‌هایی که هر یک از مؤلفه‌های سیستم می‌توانند داشته باشند را نشان می‌دهد. این رابط‌ها برای مؤلفه‌های رایانشی و فیزیکی صدق می‌کند. در هر مؤلفه علاوه بر رابط‌ها، یک حالت به نام F وجود دارد که نشان‌دهنده سالمی یا خرابی مؤلفه است. هرگاه یک مؤلفه دچار خرابی شود و نتواند فعالیت عادی خود را دنبال کند مقدار این حالت تنظیم می‌شود.




**شکل ۶: رابط‌های هر مؤلفه سیستم.**

با توجه به شکل ۶ هر مؤلفه سه دسته ورودی و یک دسته خروجی مختلف دارد که عبارت‌اند از:

- **ورودی‌های داده‌ای:** $I_i$ها در شکل ۶ ورودی‌های داده‌ای را مشخص می‌کند که از طریق مؤلفه‌های دیگر تأمین می‌شود. این مقادیر از گراف وابستگی داده‌ای استخراج می‌شود
- **ورودی‌های غیرداده‌ای:** ورودی‌های $E_i$ تأثیری است که هر مؤلفه از سایر مؤلفه‌ها می‌پذیرد. به‌طور مثال خرابی یک مؤلفه باعث افزایش نرخ خرابی مؤلفه‌ی دیگری می‌شود این مقادیر از گراف وابستگی غیرداده‌ای استخراج می‌شود
- **خروجی‌های داده‌ای:** $O_i$ ها در شکل ۶ خروجی‌های داده‌ای را مشخص می‌کند که برای مؤلفه‌های دیگر تأمین می‌شود. این مقادیر از گراف وابستگی داده‌ای استخراج می‌شود
- **ورودی‌های ترتیب اجرایی:** ورودی‌های $V_i$ مشخص می‌کند که هم‌اکنون این مؤلفه باید کار خود را انجام دهد یا در حالت آماده‌به‌کار باشد. درواقع هر مؤلفه با توجه به این مقادیر تصمیم می‌گیرد که هم‌اکنون در حالت مشغول به کار باشد یا خیر.

### ٤-٦- عبارت‌های احتمالی

مؤلفه‌هایی که ورودی دارند طبق روش‌های احتمالی FPTA که قبلاً در مقاله [۲۵] ارائه شده است. می‌توانند رفتار احتمالی داشته باشند. در این مقاله نیز از عبارت‌های احتمالی FPTA با تغییراتی جزئی استفاده شده است. در روش پیشنهادی ما، فرض شده است که هر ورودی یا خروجی دو حالت صحیح یا ناصحیح دارد. هر مؤلفه می‌تواند چندین ورودی داشته باشد. هریک از این ورودی‌ها می‌تواند صحیح یا ناصحیح باشد. رفتار مؤلفه با توجه به حالت‌های مختلفی که این ورودی‌ها دارند، می‌تواند احتمالی باشد. درواقع این عبارت‌ها نحوه‌ی انتشار خرابی از یک مؤلفه به مؤلفه‌های دیگر را به‌صورت احتمالی مشخص می‌کند. تمامی عبارت‌های احتمالی نیازی نیست که نوشته شوند و اگر فقط برای حالت‌هایی که خروجی صحیح است نوشته شود سایر عبارت‌های احتمالی قابل استنتاج است. بنابراین درصورتی‌که یک مؤلفه n ورودی و m خروجی مختلف داشته باشد به تعداد $2^n * 2^m$ عبارت احتمالی متفاوت نیاز داریم که $2^m$ حالت آن را می‌توان به‌صورت خودکار استخراج نمود. عبارت‌های احتمالی به‌صورت مجموعه معادله ۴ نشان داده می‌شود.

$$\begin{matrix} 1)\ \text{Input1.T, Input2.}T \rightarrow \text{Output1.T, 1.0} \\ 2)\ \text{Input1.}F\text{, Input2.T} \rightarrow \text{Output1.T, 0.6} \\ 3)\ \text{Input1.}T\text{, Input2.F} \rightarrow \text{Output1.T, 0.7} \\ 4)\ \text{Input1.}F\text{, Input2.F} \rightarrow \text{Output1.T, 0.02} \end{matrix} \quad (۴)$$

معادله‌های ۴ صحیح یا ناصحیح بودن خروجی مؤلفه در حالت‌های مختلفی که ورودی‌ها می‌توانند داشته باشند را نشان می‌دهد. به‌طور مثال با توجه به عبارت دوم هنگامی‌که ورودی اول ناصحیح باشد و ورودی دوم صحیح باشد این مؤلفه به‌احتمال ۰.۶ خروجی صحیح تولید می‌کند و به احتمال ۰.۴ مقدار خروجی این مؤلفه ناصحیح است.

### ٤-٧- تبدیل گراف وابستگی داده‌ای به مدل SAN

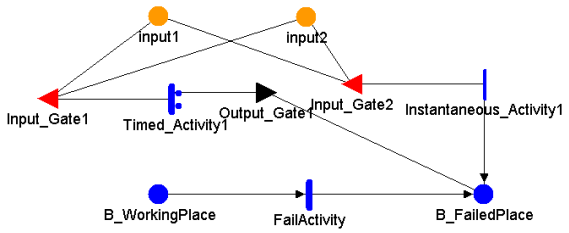
برای تبدیل این گراف به مدل SAN هرگاه از مؤلفه نوعی A به مؤلفه نوعی B یالی وجود داشته باشد، به معنای این است که A باید برای B ورودی ایجاد کند و آن را در اختیار B قرار دهد. این وابستگی از طریق یک مکان بسط‌یافته[۱۴] در مدل SAN ایجاد می‌گردد. ورودی‌ها و خروجی‌ها در مدل پیشنهادی می‌تواند دو حالت مختلف داشته باشد، حالتی که ورودی صحیح است و حالتی که ورودی ناصحیح است. بنابراین نشانه موجود در این مکان می‌تواند یک عدد صحیح باشد که عدد یک به معنای داده صحیح و عدد دو به معنای داده ناصحیح است.

$$O_i = \begin{cases} 1, & \text{مقدار خروجی ناصحیح باشد} \\ 2, & \text{مقدار خروجی صحیح باشد} \end{cases} \quad (۴)$$

$$I_i = \begin{cases} 1, & \text{مقدار ورودی ناصحیح باشد} \\ 2, & \text{مقدار ورودی صحیح باشد} \end{cases} \quad (۵)$$

همچنین می‌توان از مکان بسط‌یافته برای نشان دادن مقدار یک پارامتر استفاده کرد. به‌طور مثال برای نشان دادن تعداد کار انجام شده توسط یک پردازنده می‌توان از یک مکان بسط‌یافته برای شمارش این کارها استفاده کرد.

پس از ساخت مؤلفه‌های سیستم، هنگام الحاق مؤلفه‌ها به یکدیگر به منظورتشکیل مدل نهایی، مکان‌های بسط‌یافته مؤلفه‌هایی که به همدیگر وابستگی داده‌ای دارند باید به اشتراک گذاشته شود. یکی از مؤلفه‌ها با یک نرخ مشخص مقادیر خروجی را در آن مکان قرار می‌دهد و مؤلفه دیگر نیز با یک نرخ مشخص این مقادیر را به‌عنوان ورودی برمی‌دارد. در شکل ۷ مدل یک مؤلفه با ۲ ورودی مختلف نشان داده شده است.

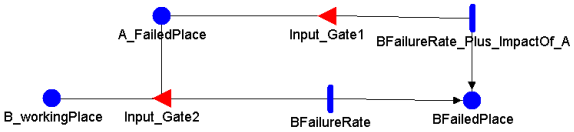



**شکل ۷: مؤلفه A با دو ورودی از سایر مولفه‌ها.**

### ٤-٨- تبدیل گراف وابستگی غیرداده‌ای به مدل SAN

برای تزریق وابستگی غیرداده‌ای به مدل SAN باید به‌صورت زیر عمل کرد:

- درصورتی‌که مؤلفه B به مؤلفه A وابستگی غیرداده‌ای داشته باشد باید حالت خرابی F مؤلفه A در مدل مؤلفه B قرار گیرد و این مکان بین این دو مؤلفه به اشتراک گذاشته شود. با استفاده از این مکان خرابی زمانی که مؤلفه A خراب شد مؤلفه B از طریق این مکان متوجه خرابی مؤلفه A شود.
- به ازای هر وابستگی غیرداده‌ای یک فعالیت جدید با نرخ جدیدی که از گراف وابستگی غیرداده‌ای محاسبه می‌شود در مدل مؤلفه قرار داده می‌شود.
- به ازای هر فعالیت باید یک دروازه ورودی قرار داده شود که تابع شرطی آن با توجه به حالت خرابی مؤلفه A فعالیت مربوطه را فعال یا غیرفعال کند. به‌صورتی که قبل از خرابی مؤلفه A نرخ خرابی مؤلفه B از فعالیت اول پیروی کند و درصورتی‌که نشانه‌ای در مکان خرابی مؤلفه A قرار داده شد باید از فعالیت جدید پیروی کند. در آن واحد باید یکی از تابع‌های شرطی مربوط به دروازه‌های ورودی درست باشد و در هیچ حالتی دو تابع شرطی نباید هم‌زمان صحیح باشد.




**شکل ۸: مدل مؤلفه B در صورت داشتن وابستگی غیرداده‌ای به مؤلفه A**

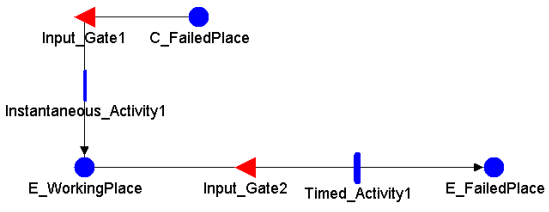
همان‌طور که در شکل ۸ نشان داده شده است به دلیل وابستگی مؤلفه B به مؤلفه A باید مکان مربوط به خرابی مؤلفه A در مدل مؤلفه B قرار داده شود. این مکان به اسم A_FailedPlace در شکل ۸ نشان داده شده است. تا زمانی که مؤلفه A سالم باشد نرخ خرابی مؤلفه B از فعالیت BFailureRate پیروی می‌کند و هنگامی‌که مؤلفه A خراب شد، نرخ خرابی مؤلفه B از فعالیت BFailureRate_Plus_ImpactOf_A پیروی می‌کند. هرکدام از دروازه‌های ورودی Input_Gate1 و Input_Gate2 نظارت می‌کنند که فعالیت خرابی مؤلفه کدام یک از فعالیت‌ها باشد.

### ٤-٩- تبدیل گراف ترتیب اجرایی به مدل SAN

درصورتی‌که کار مؤلفه E بعد از خرابی مؤلفه C شروع شود برای تبدیل گراف ترتیب اجرایی به مدل SAN باید به‌صورت زیر عمل کرد:

- مکان مربوط به حالت خرابی مؤلفه C باید در داخل مؤلفه E قرار گیرد و این مکان بین این دو مؤلفه به اشتراک گذاشته شود.
- یک فعالیت باید به مؤلفه E اضافه شود تا درصورتی‌که نشانه‌ای داخل مکان خرابی مؤلفه C قرار گرفت این فعالیت فعال شود و یک نشانه داخل مکان مربوط به در حال کار بودن این مؤلفه قرار دهد.

همان‌طور که در شکل ۹ نشان داده شد هنگامی‌که نشانه‌ای در مکان C_FailedPlace قرار بگیرد، فعالیت فوری Activity1 یک نشانه در مکان E_WorkingPlace قرار می‌دهد و این به معنای آغاز به کار کردن این مؤلفه است. فعالیت Activity1 زمانی شلیک می‌کند که در داخل C_FailedPlace نشانه‌ای وجود داشته باشد و در مکان‌های E_WorkingPlace و E_FailedPlace نشانه‌ای وجود نداشته باشد.




**شکل ۹: مدل مؤلفه E در صورت وجود ترتیب اجرایی بین مؤلفه C و E**

### ۴-۱۰- تزریق عبارت‌های احتمالی به مدل SAN

برای تزریق عبارت‌های احتمالی مربوط به یک مؤلفه در داخل مدل آن مؤلفه، می‌توان از فعالیت‌های احتمالی با چندین خروجی استفاده کرد. فعالیت‌های احتمالی به این صورت است که می‌توان مشخص کرد در صورت فعال بودن فعالیت، در p درصد مواقع نشانه را از طریق خروجی اول بفرستد و در 1-p درصد مواقع نشانه را از طریق خروجی دوم ارسال کند. هر فعالیت می‌تواند n خروجی مختلف داشته باشد ولی مجموع احتمال آن‌ها باید یک باشد. به ازای هر عبارت احتمالی و برعکس آن یک فعالیت چندتایی نیاز داریم و هر یک از این فعالیت‌ها به یک دروازه ورودی متصل شده است که با توجه به ورودی‌های مؤلفه مشخص می‌کند این فعالیت هم‌اکنون باید فعال باشد یا خیر. در هرلحظه یکی از چندین فعالیت احتمالی باید فعال باشد. به‌طور مثال برای دو عبارت ۶ یک فعالیت احتمالی با ۲ خروجی نیاز داریم.

$$\begin{aligned} &1)\ \mathrm{Input1}.F, \mathrm{Input2}.T \rightarrow \mathrm{Output1.T}, 0.6 \\ &2)\ \mathrm{Input1}.F, \mathrm{Input2.T} \rightarrow \mathrm{Output1.F}, 0.4 \end{aligned} \quad (۶)$$

در شکل ۱۰ یک فعالیت احتمالی با ۲ خروجی نشان داده شده‌است. در تعریف این فعالیت می‌توان مشخص کرد که هنگام فعال بودن در ۶۰ درصد مواقع نشانه از خروجی اول انتقال پیدا کند و در ۴۰ درصد مواقع از خروجی دوم منتقل شود. مجموع این احتمالات برای هر فعالیت باید برابر با یک شود.


**شکل ۱۰: یک نمونه فعالیت احتمالی با دو خروجی**

### ۴-۱۱- الحاق مؤلفه‌ها از طریق صورت‌بندی تکرار-الحاق

برای الحاق مؤلفه‌ها از قوانین زیر استفاده می‌شود. قوانین اتصال مؤلفه‌ها به یکدیگر به‌صورت زیر دسته‌بندی می‌شود:

- اتصال مؤلفه‌هایی که به همدیگر وابستگی داده‌ای دارند.
- اتصال مؤلفه‌هایی که به همدیگر وابستگی غیرداده‌ای دارند از طریق اشتراک مکان خرابی هر یک از مؤلفه‌ها.
- اتصال مؤلفه‌هایی که به‌منظور افزونگی در سیستم استفاده شده است از طریق اشتراک مکان خرابی مؤلفه‌های پیش‌نیاز.

### ۴-۱۲- روش به دست آوردن نتایج ارزیابی

برای محاسبه مقادیر میانگین زمان تا خرابی، احتمال خرابی و مدت زمان بیکار بودن هر یک از مؤلفه‌ها به صورت زیر عمل شده است.

روش استخراج میزان MTTF به این صورت است که اگر ما حد آستانه خرابی مؤلفه را $0 \leq x \leq 1$ در نظر بگیریم و همچنین هنگام انجام آزمایش، تعداد دفعات اجرای مدل بعد از همگرایی نتایج در عدد $n$ متوقف شود. میانگین زمان تا خرابی مؤلفه برابر می‌شود با نقطه زمانی که در آن نقطه، $nx$ از آزمایش‌های مورد نظر در حالت خرابی باشد.

به منظور محاسبه احتمال خرابی، اگر احتمال خرابی در ثانیه $s$ مد نظر باشد مقدار این احتمال از معادله زیر بدست می‌آید:

$$p(s) = \frac{\text{تعداد آزمایش‌هایی که در نقطه } s \text{ زمانی در حالت خرابی هستند}}{\text{تعداد کل آزمایش‌ها}}$$

لازم به ذکر است تعداد کل آزمایش‌ها در نقطه‌ای متوقف شده است که اجرای بیشتر آزمایش‌ها تغییری در احتمالات به وجود نیاورد و در واقع نتایج همگرا شده است.

در مواقعی که نیاز به محاسبه میزان زمان بیکاری را داریم. یک سطح آستانه به طور مثال $0 \leq S \leq 1$ را مشخص می‌کنیم. این سطح آستانه مشخص می‌کند، درصورتی که $s$ درصد از مؤلفه‌ها در حالت فعال باشند ما آن مؤلفه را دیگر فعال در نظر می‌گیریم. نقطه‌ای در طول زمان انتخاب می‌شود که در آن نقطه، مؤلفه مورد نظر در $ns$ از آزمایش‌های انجام شده در حالت فعال باشد یا به بیانی دیگر فقط در $1 - ns$ آزمایش مؤلفه در حالت بیکار باشد.

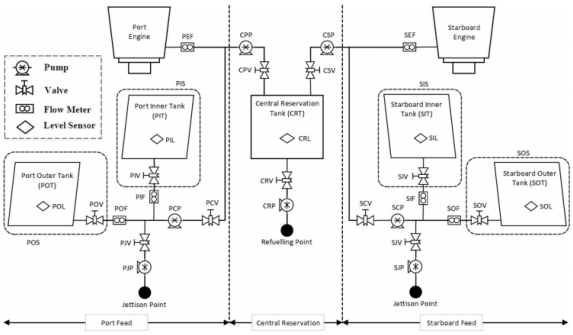
## ۵- مطالعه موردی

در بخش قبل روش پیشنهادی مدل‌سازی انتشار خطا در سیستم‌های هیبرید شرح داده شد. در این فصل به مدل‌سازی سیستم سوخت‌رسان هواپیما پرداخته شده است. ابتدا مشخصات این سیستم شرح داده می‌شود. سپس روش ساخت مدل هر یک از مؤلفه‌های سیستم نشان داده می‌شود. درنهایت برای ساخت مدل نهایی، مؤلفه‌ها با روش گفته‌شده در بخش قبل به یکدیگر الحاق می‌شوند. این هواپیما دارای دو موتور است که هر یک از این موتورها دارای یک مخزن سوخت داخلی و یک مخزن سوخت خارجی است. سوخت‌رسانی به هریک از موتورها از طریق مخزن داخلی انجام می‌شود و درصورت بروز مشکل برای مخزن داخلی از مخزن خارجی استفاده می‌شود. یک مخزن رزرو شده نیز برای سیستم وجود دارد که در مواقع اضطراری می‌تواند فرایند سوخت رسانی به هر دو موتور را انجام می‌دهد. این سیستم دارای یک خنک‌کننده و چندین حسگر است کخ در ادامه جزییات این سیستم و مشخصات کامل‌تر آن آورده شده است.

### ۵-۱- مشخصات سیستم سوخت‌رسان

مثال بررسی‌شده در این قسمت یک سیستم سوخت‌رسان است که در مقاله [۱۹] ارائه شده است. همبندی مؤلفه‌های این سیستم در

شکل ۱۱ نشان داده شده است. این سیستم دارای دو موتور و سه زیرسیستم مختلف است. در زیرسیستم بخش راست یک موتور با دو مخزن سوخت آن قرار داده شده است. همچنین در زیرسیستم سمت چپ موتور دیگر و مخازن مربوط به آن وجود دارد. در زیرسیستم وسط مخزن سوخت رزرو شده قرار دارد. نحوه سوخت‌رسانی به موتورها به این‌گونه است که هر موتور یک مخزن سوخت داخلی و یک مخزن سوخت خارجی دارد در حالت عادی سوخت هر یک از موتورها از طریق مخزن خارجی زیرسیستم مربوطه تأمین می‌شود. در صورت خرابی مخزن خارجی، موتور از مخزن داخلی استفاده می‌کند. هنگامی‌که هر دو مخزن سوخت دچار خرابی شدند، موتور از مخزن سوخت رزرو شده استفاده می‌کند. دو حسگر فشار و دما و همچنین یک سیستم خنک‌کننده برای هر زیرسیستم وجود دارد، که خنک‌کننده با توجه به فشار و دمای محیط، میزان خنک‌سازی محیط را تنظیم می‌کند.




**شکل ۱۱: سیستم سوخت‌رسان**

### ۵-۲- گراف‌ها و عبارت‌های احتمالی مربوط به سیستم

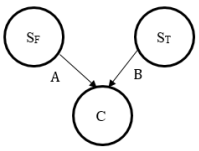
در این بخش تشریح هر یک از سه گراف مربوط به این سیستم قرار داده شده است. با توجه به اینکه ساختار هر دو زیرسیستم سمت چپ و راست دقیقاً همانند یکدیگر است، یکی از این زیرسیستم‌ها در نظر گرفته شده است. مخفف‌هایی که برای اشاره به مؤلفه‌های مختلف استفاده شده است در جدول ۲ نشان داده شده است.

**جدول ۲: مخفف اسامی مؤلفه‌های سیستم سوخت‌رسان.**

| مخفف | نام مؤلفه |
|---|---|
| خنک‌کننده | C |
| مخزن داخلی | $T_I$ |
| مخزن خارجی | $T_O$ |
| مخزن رزرو شده | $T_R$ |
| حسگر فشار | $S_P$ |
| حسگر دما | $S_T$ |

#### ◘ گراف وابستگی داده‌ای سیستم سوخت‌رسان

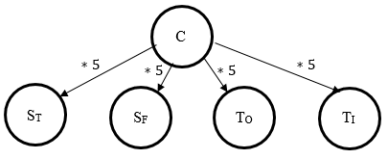
در زیرسیستم سمت چپ، حسگرهای فشار و دما اطلاعات اندازه‌گیری شده را به خنک‌کننده ارسال می‌کنند. گراف مربوط به آن در شکل ۱۲ نشان داده شده است.




**شکل ۱۲: گراف وابستگی داده‌ای سیستم سوخت‌رسان**

#### ◘ گراف وابستگی غیرداده‌ای سیستم سوخت‌رسان

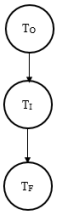
درصورتی‌که سیستم خنک‌کننده دچار خرابی شود، نرخ خرابی مؤلفه‌های حسگر فشار، حسگر دما، مخزن داخلی و مخزن خارجی افزایش پیدا می‌کند.گراف وابستگی غیرداده‌ای مربوط به این سیستم در شکل ۱۳ نشان داده شده است.




**شکل ۱۳: گراف وابستگی غیرداده‌ای سیستم سوخت‌رسان**

#### ◘ گراف ترتیب اجرایی سیستم سوخت‌رسان

گراف ترتیب اجرایی مربوط به این سیستم در شکل ۱۴ نشان داده شده است. همان‌طور که در توضیحات سیستم گفته شد؛ ابتدا از مخزن خارجی استفاده می‌شود در صورت خرابی این مخزن، پمپ مربوط به مخزن داخلی فعال می‌شود و بعد از خرابی این مؤلفه از مخزن رزرو شده استفاده می‌گردد.




**شکل ۱۴: گراف ترتیب اجرایی سیستم سوخت‌رسان**

#### ◘ عبارت‌های احتمالی مربوط به سیستم سوخت‌رسان

هنگامی‌که حسگر فشار و حسگر دما درست کار نکنند خنک‌کننده نمی‌تواند عملکرد صحیح خود را انجام دهد. اما درصورتی‌که یکی از این دو حسگر درست کار نکنند در ۹۵ درصد مواقع خنک‌کننده

درست کار می‌کند. عبارت‌های احتمالی مربوط به این مؤلفه در معادله‌های ۷ آورده شده است.

$$S_T.False, S_F.False \rightarrow C.Fail, 1$$
$$S_T.True, S_F.False \rightarrow C.Fail, 0.05 \quad (۷)$$
$$S_T.False, S_F.True \rightarrow C.Fail, 0.01$$

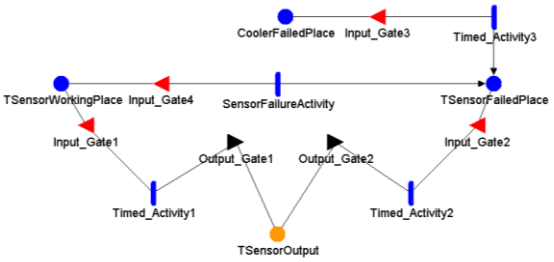
### ۳-۵- طراحی مدل SAN مؤلفه‌های سیستم سوخت‌رسان

در این بخش هر یک از مؤلفه‌های سیستم سوخت‌رسان طراحی شده است. در طراحی هر یک از مؤلفه‌ها، نحوه‌ی استفاده از گراف‌های تولیدشده شرح داده است. تمامی متغیرهایی که برای نرخ خرابی یا تأثیر بر روی نرخ خرابی در نظر گرفته شده است در هنگام شبیه‌سازی مقداردهی می‌شود.

#### ◘ طراحی مؤلفه حسگر حرارتی

مدل SAN مربوط به حسگر حرارتی در شکل ۱۵ نشان داده شده است. مدل مؤلفه حسگر فشار نیز به همین صورت است. با توجه به شکل ۱۵، مکان‌های مدل مؤلفه SAN به‌صورت زیر است:

- **TCoolerFailedPlace:** به دلیل اینکه حسگر حرارتی به خنک‌کننده وابستگی غیرداده‌ای دارد بنابراین باید مکان خرابی خنک‌کننده با حسگر به اشتراک گذاشته شود.
- **TSensorFailedPlace:** این مکان نشان‌دهنده خرابی حسگر است.
- **TSensorWorkingPlace:** این مکان نشان‌دهنده سالم بودن حسگر است.
- **TSensorOutput:** در این مکان خروجی حسگر حرارتی قرار می‌گیرد. هنگامی‌که حسگر سالم باشد مقدار ۱ را در این مکان قرار می‌دهد و هنگامی‌که حسگر دچار خرابی شود مقدار ۲ را در این مکان قرار می‌دهد.




**شکل ۱۵: مدل SAN حسگر حرارتی**

جداول مربوط به دروازه‌های ورودی در جدول ۳ آمده است همچنین مشخصات دروازه‌های ورودی این مدل به این صورت است:

- **Input_Gate1:** این دروازه هنگامی اجازه شلیک به فعالیت Time_Activity1 را می‌دهد که حسگر خراب نشده باشد و مقدار داخل مکان TsensorOutput یک نباشد.
- **Input_Gate2:** این دروازه هنگامی اجازه شلیک به فعالیت Time_Activity2 را می‌دهد که حسگر خراب شده باشد
- **Input_Gate3:** این دروازه هنگامی اجازه شلیک به فعالیت Time_Activity3 را می‌دهد که خنک‌کننده خراب شده باشد و حسگر در این لحظه سالم باشد.
- **Input_Gate4:** این دروازه هنگامی اجازه شلیک به فعالیت SensorFailureActivity را می‌دهد که خنک‌کننده سالم باشد و حسگر نیز خراب نشده باشد.

**جدول ۳: جدول دروازه‌های ورودی مدل مؤلفه حسگر حرارتی**

| | |
|---|---|
| **Input_Gate1** | **عبارت شرطی** |
| | TSensorOutput->IOvalue->Mark()!=1&&TSensorWorkingPlace->Mark()==1 |
| **Input_Gate2** | **عبارت شرطی** |
| | TSensorOutput->IOvalue->Mark()!=2 && TSensorFailedPlace->Mark()==1 |
| **Input_Gate3** | **عبارت شرطی** |
| | CoolerFailedPlace->Mark()==1 && TSensorWorkingPlace->Mark()==1 && TSensorFailedPlace->Mark()==0 |
| **Input_Gate4** | **عبارت شرطی** |
| | CoolerFailedPlace->Mark()==0 && TSensorWorkingPlace->Mark()==1 && TSensorFailedPlace->Mark()==0 |
| | **تابع ورودی** |
| | TSensorWorkingPlace->Mark()=0; |

جداول مربوط به دروازه‌های خروجی در جدول ۴ آمده است. همچنین نرخ فعالیت‌های این مؤلفه در جدول ۵ نشان داده شده است. تمامی نرخ‌ها از توزیع نمایی پیروی می‌کنند.

مشخصات دروازه‌های خروجی این مدل به‌صورت زیر است:

- **Output_Gate1:** هنگامی‌که فعالیت Time_Activity1 شلیک کند این دروازه مقدار مکان TsensorOutput را برابر یک قرار می‌دهد.
- **Output_Gate2:** هنگامی‌که فعالیت Time_Activity2 شلیک کند این دروازه مقدار مکان TsensorOutput را برابر دو قرار می‌دهد.

**جدول ۴: جدول دروازه‌های خروجی مدل مؤلفه حسگر حرارتی**

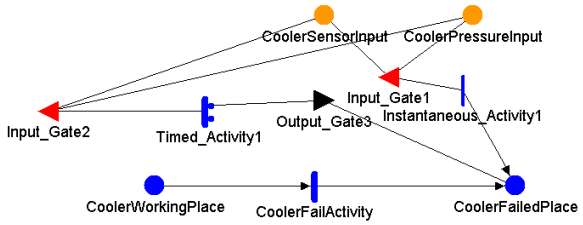
| | |
|---|---|
| **Output_Gate1** | **تابع خروجی** |
| | TSensorOutput->IOvalue->Mark()=1; |
| **Output_Gate2** | **تابع خروجی** |
| | TSensorOutput->IOvalue->Mark()=2; |

**جدول ۵: نرخ فعالیت‌های مدل مؤلفه حسگر حرارتی**

| نام فعالیت | نرخ شلیک |
|---|---|
| Time_Activity1 | 10 |
| Time_Activity2 | 10 |
| Time_Activity3 | TSensorFailureRate + TSensorFailureRate*4 |
| SensorFailureActivity | TSensorFailureRate |

### طراحی مؤلفه خنک‌کننده

خنک‌کننده در این سیستم به این صورت کار می‌کند که اطلاعات فشار و دما را از حسگرهای حرارتی و فشار می‌گیرد، سپس دمای محیط را با توجه به این مقادیر تنظیم می‌کند. درصورتی‌که هر دو مقدار ورودی به این مؤلفه نادرست باشد خنک‌کننده نمی‌تواند درست کار کند و به اصطلاح دچار خرابی می‌شود. درصورتی‌که یکی از این دو مقدار ناصحیح باشد در ۹۵ درصد مواقع خنک‌کننده درست کار می‌کند و در ۵ درصد مواقع اشتباه کار می‌کند. مدل SAN این مؤلفه در شکل ۱۶ نشان داده شده است.




**شکل ۱۶: مدل SAN خنک‌کننده**

مشخصات مکان‌های مدل این مؤلفه:

- **CoolerWorkingPlace:** نشان‌دهنده سالمی خنک‌کننده است.
- **CoolerFailedPlace:** نشان دهنده خرابی خنک‌کننده است.
- **CoolerSensorInput:** این مکان به دلیل وابستگی داده‌ای خنک‌کننده با حسگر حرارتی، بین آن‌ها به اشتراک گذاشته است و مقدار دمایی که حسگر حرارتی گزارش می‌دهد از این مکان برمی‌دارد.
- **CoolerPresureInput:** این مکان به دلیل وابستگی داده‌ای خنک‌کننده با حسگر فشار، بین آن به اشتراک گذاشته است و مقدار فشاری که حسگر فشار گزارش می‌دهد از این مکان برمی‌دارد.

جدول مربوط به دروازه‌های خروجی و ورودی در جدول ۶ و ۷ نشان داده شده‌است. همچنین نرخ خرابی فعالیت‌های این مدل در جدول ۸ نشان داده شده است. تمامی نرخ‌ها از توزیع نمایی پیروی می‌کنند. فعالیت Time_Activity1 با توجه به عبارت‌های احتمالی خنک‌کننده که در بخش قبل توضیح داده شد، دارای دو حالت احتمالی با احتمالات ۰.۰۵ و ۰.۹۵ است.

**جدول ۶: جدول دروازه‌های خروجی مدل مؤلفه خنک‌کننده**

| | تابع خروجی |
|---|---|
| **Output_Gate1** | CoolerFailedPlace->Mark()=1;CoolerWorkingPlace->Mark()=0; |

**جدول ۷: جدول دروازه‌های ورودی مدل مؤلفه خنک‌کننده**

| | عبارت شرطی |
|---|---|
| **Input_Gate1** | CoolerWorkingPlace->Mark()==1 && CoolerSensorInput->IOvalue->Mark()==2 && CoolerPresureInput->IOvalue->Mark()==2 |
| | **تابع ورودی** |
| | CoolerWorkingPlace->Mark()=0 |
| | **عبارت شرطی** |
| **Input_Gate2** | CoolerWorkingPlace->Mark()==1 &&((CoolerSensorInput->IOvalue->Mark()==2 && CoolerPresureInput->IOvalue->Mark()==1) \|\| (CoolerSensorInput->IOvalue->Mark()==1 && CoolerPresureInput->IOvalue->Mark()==2)) |

**جدول ۸: نرخ فعالیت‌های مدل مؤلفه خنک‌کننده**

| نام فعالیت | نرخ شلیک |
|---|---|
| Time_Activity1 | 10 |
| CoolerFailureRate | CoolerFailureRate |

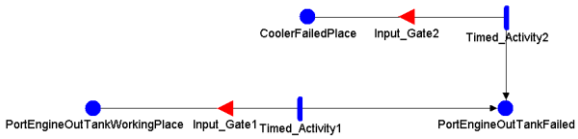
### طراحی مؤلفه مخزن خارجی

مدل SAN این مؤلفه در شکل ۱۷ نشان داده شده است. مشخصات مکان‌ها این مدل به‌صورت زیر است:

- **CoolerFailedPlace:** با توجه به اینکه مخزن خارجی با خنک‌کننده وابستگی غیرداده‌ای دارد. حالت خرابی خنک‌کننده، داخل مدل مخزن خارجی قرار داده شده است.
- **PortEngineOutTankFailed:** حالت خرابی مخزن
- **PortEngineOutTankWorking:** حالت سالم بودن مخزن

نرخ خرابی فعالیت‌های این مدل در جدول ۹ نشان داده شده است. تمامی نرخ‌ها از توزیع نمایی پیروی می‌کنند.

جداول مربوط به دروازه‌های ورودی در جدول ۱۰ آورده شده است.




**شکل ۱۷: مدل SAN مخزن خارجی**

**جدول ۹: نرخ فعالیت‌های مدل مؤلفه مخزن خارجی**

| نام فعالیت | نرخ شلیک |
|---|---|
| Time_Activity1 | PortEngineTankFailureRate |
| Time_Activity2 | PortEngineTankFailureRate + 4* PortEngineTankFailureRate |

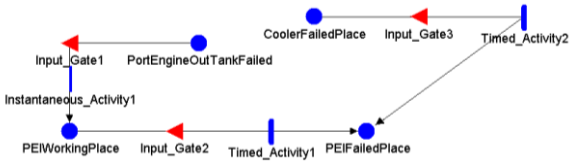
**جدول ۱۰: جدول دروازه‌های ورودی مدل مؤلفه مخزن خارجی**

| | |
|---|---|
| **Input_Gate1** | **عبارت شرطی** |
| | CoolerFailedPlace->Mark()==0 && PortEngineOutTankFailed->Mark()==0 && PortEngineOutTankWorkingPlace->Mark()==1 |
| **Input_Gate2** | **عبارت شرطی** |
| | CoolerFailedPlace->Mark()==1 && PortEngineOutTankFailed->Mark()==0 && PortEngineOutTankWorkingPlace->Mark()==1 |

### ❑ طراحی مؤلفه مخزن داخلی

مدل SAN این مؤلفه در شکل ۱۸ نشان داده شده است. مکان‌های استفاده شده در این مدل عبارت‌اند از:

- **CoolerFailedPlace**: با توجه به اینکه مخزن داخلی به خنک‌کننده وابستگی غیردادهای دارد، بنابراین مکانی که نشان‌دهنده خرابی خنک‌کننده است باید در این مدل قرار داده شود.
- **PortEngineOutTankFailed**: این مکان نشان‌دهنده خرابی مخزن خارجی است. با توجه به گراف ترتیب اجرایی، فعالیت مخزن داخلی بعد از خرابی مخزن خارجی شروع می‌شود، بنابراین مکانی که نشان‌دهنده خرابی مخزن خارجی است باید در مدل قرار داده شود.
- **PEIFailedPlace**: این مکان نشان‌دهنده خرابی مخزن داخلی است.
- **PEIWorkingPlace**: این مکان نشان‌دهنده آماده به کار بودن مخزن است.




**شکل ۱۸: مدل SAN مخزن داخلی**

جداول مربوط به دروازه‌های ورودی در جدول ۱۲ آورده شده است.

**جدول ۱۲: جدول دروازه‌های ورودی مدل مؤلفه داخلی**

| | |
|---|---|
| **Input_Gate1** | **عبارت شرطی** |
| | PortEngineOutTankFailed->Mark()==1 && PEIWorkingPlace->Mark()==0 && PEIFailedPlace->Mark()==0 |
| | **تابع ورودی** |
| | PortEngineOutTankFailed->Mark()=1; |
| **Input_Gate2** | **عبارت شرطی** |
| | CoolerFailedPlace->Mark()==0 && PEIWorkingPlace->Mark()==1 && PEIFailedPlace->Mark()==0 |
| | **تابع ورودی** |
| | PEIWorkingPlace->Mark()=0; |
| **Input_Gate3** | **عبارت شرطی** |
| | CoolerFailedPlace->Mark()==1 && PEIFailedPlace->Mark()==0 && PEIWorkingPlace->Mark()==1 |

همچنین نرخ خرابی فعالیت‌های این مدل در ۱۱ نشان داده شده است. تمامی نرخ‌ها از توزیع نمایی پیروی می‌کند.

**جدول ۱۱: نرخ فعالیت‌های مدل مؤلفه داخلی**

| نام فعالیت | نرخ شلیک |
|---|---|
| Time_Activity1 | PInnerTankFailureRate |
| Time_Activity2 | PInnerTankFailureRate+ 4* PInnerTankFailureRate |

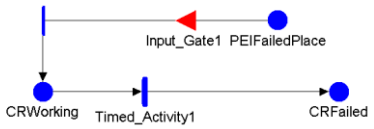
### ❑ طراحی مؤلفه مخزن رزرو شده

مدل SAN این مؤلفه در شکل ۱۹ نشان داده شده است. مکان‌های استفاده شده در این مدل عبارت‌اند از:

- **CRWorking**: این مکان نشان‌دهنده انجام عمل سوخت‌رسانی توسط مخزن رزرو شده است. با توجه به اینکه مخزن رزرو شده وابستگی غیردادهای به مخزن داخلی دارد، مکانی که نشان‌دهنده خرابی مخزن داخلی است باید در مدل قرار داده شود.
- **PEIFailedPlace**: این مکان نشان‌دهنده خرابی مخزن داخلی است. به دلیل دارا بودن ترتیب اجرایی با مخزن داخلی، این مکان در مدل قرار داده شده است.
- **CRFailed**: هنگامی‌که مخزن رزرو شده دچار خرابی شود یک نشانه داخل این مکان قرار داده می‌شود.

جدول مربوط به دروازه ورودی در جدول ۱۳ نشان داده شده است. همچنین مشخصات دروازه‌ی ورودی این مدل به‌صورت زیر است:

- **Input_Gate1**: این دروازه هنگامی اجازه شلیک به فعالیت Instantaneous_Activity1 را می‌دهد که مخزن داخلی دچار خرابی شود.




**شکل ۱۹: مدل SAN مخزن رزرو شده**

نرخ خرابی فعالیت‌های این مدل در جدول ۱۴ نشان داده شده است. تمامی نرخ‌ها از توزیع نمایی پیروی می‌کند:

**جدول ۱۳: جدول دروازه‌های ورودی مدل مؤلفه رزرو شده**

| | |
|---|---|
| **Input_Gate1** | **عبارت شرطی** |
| | PEIFailedPlace->Mark()==1 && CRWorking->Mark()==0 && CRFailed->Mark()== |

**جدول ۱۴: نرخ فعالیت‌های مدل مؤلفه رزرو شده**

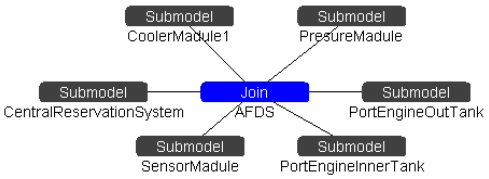
| نام فعالیت | نرخ شلیک |
|---|---|
| Time_Activity1 | CRFailureRate |

### ۵-۴- الحاق مؤلفه‌های سیستم سوخت‌رسان

با توجه به گراف‌های سه‌گانه‌ی سیستم سوخت‌رسان که در بخش قبل توضیح داده شد، برای الحاق مؤلفه‌ها با یکدیگر باید مکان‌ها به‌صورت زیر با یکدیگر به اشتراک گذاشته شوند.

- مکان‌های CoolerFailedPlace در مدل مؤلفه‌های حسگر حرارتی، حسگر فشار، خنک‌کننده، مخزن خارجی و مخزن داخلی (این اشتراک به دلیل وجود وابستگی غیرداده‌ای است).
- مکان‌های PEIFailedPlace در مدل مؤلفه‌های مخزن داخلی و مخزن رزرو شده (این اشتراک به خاطر حضور این دو مؤلفه در گراف ترتیب اجرایی است).
- مکان‌های PortEngineOutTankFailed در مدل مؤلفه‌های مخزن داخلی و مخزن خارجی (این اشتراک به خاطر حضور این دو مؤلفه در گراف ترتیب اجرایی است).
- مکان‌های CoolerPresureInput و PresureOutput در مدل مؤلفه‌های خنک‌کننده و حسگر فشار (این اشتراک به دلیل داشتن وابستگی داده‌ای بین این دو مؤلفه است).
- مکان‌های CoolerSensorInput و TSensorOutput در مدل مؤلفه‌های خنک‌کننده و حسگر حرارتی (این اشتراک به دلیل داشتن وابستگی داده‌ای بین این دو مؤلفه است).

تصویر مربوط به الحاق مؤلفه‌ها در شکل ۲۰ نشان داده شده است.




**شکل ۲۰: مدل تکرار-الحاق سیستم سوخت‌رسان**

### ۵-۵- ارزیابی مطالعه موردی

برای مدل‌سازی و شبیه‌سازی از نرم افزار Mobius در محیط ویندوز ۱۰ استفاده شده است.

پس از الحاق مؤلفه‌های سیستم، به منظور اجرای شبیه‌سازی ابتدا سیستم را بدون حضور هیچ خطایی اجرا می‌کنیم ، سپس با تزریق خطایی که منجر به خرابی حسگر حرارتی شود شبیه‌سازی را اجرا می‌کنیم و نتایج این دو حالت را باهم مقایسه می‌کنیم. مقادیری که برای متغیرهای این مدل در نظر گرفته شده است در جدول ۱۵ نشان داده شده است. در هر آزمایش، تعداد اجرای مدل تا زمان همگرا[۱۵] شدن نتایج به یک عدد ادامه پیدا می‌کند. در واقع از آن نقطه به بعد، افزایش تعداد دفعات اجرای مدل در خروجی مدل تغییری ایجاد نمی‌کند.

**جدول ۱۵: مقادیر درنظر گرفته‌شده برای متغیرهای موجود در مدل**

| نام متغیر | مقدار |
|---|---|
| CRFailureRate | 1.0E-3 |
| CoolerFailureRate | 1.0E-3 |
| PInnerTankFailureRate | 1.0E-3 |
| PortEngineTankFailureRate | 1.0E-3 |
| PresureFailureRate | 1.0E-4 |
| TSensorFailureRate | 1.0E-4 |

#### ◘ تحلیل رفتار خنک‌کننده در حضور و عدم حضور خطا

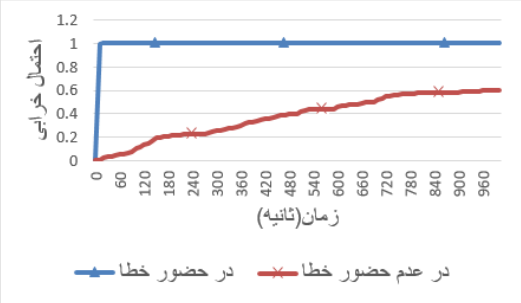
با توجه به تحلیل نمودار شکل ۲۱ اگر خنک‌کننده را با عبور از میانگین ۰.۷۵ خراب فرض کنیم. میانگین زمان لازم برای خرابی خنک‌کننده به‌صورت زیر است:

میانگین زمانی تا خرابی خنک‌کننده در حضور خطا:

MTTF = 10s

میانگین زمانی تا خرابی خنک‌کننده در عدم حضور خطا:

MTTF = 1330s




**شکل ۲۱: رفتار خرابی خنک‌کننده در حضور و عدم حضور خطا**

هنگامی‌که حسگر خراب باشد در ۷۵ درصد از آزمایش‌های انجام شده خنک‌کننده در ثانیه ۱۰ خراب شده است و هنگامی‌که حسگر خراب نباشد در ۷۵ درصد آزمایش از ثانیه ۱۳۳۰ به بعد خنک‌کننده دچار خرابی شده است.

#### ◘ تحلیل رفتار مخزن خارجی در حضور و عدم حضور خطا

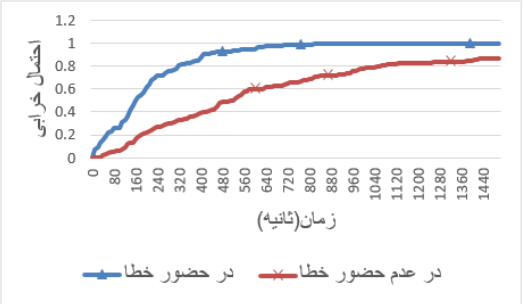
اولین مخزنی که عملیات سوخت‌رسانی را انجام می‌دهد مخزن خارجی است. با توجه به تحلیل نمودار شکل ۲۲ این مخزن را اگر پس از گذراندن میانگین ۰.۷۵ خراب فرض کنیم میانگین زمان لازم تا اولین خرابی به‌صورت زیر است:

میانگین زمانی تا خرابی مخزن خارجی در حضور خطا:

MTTF = 280s

میانگین زمانی تا خرابی مخزن خارجی در عدم حضور خطا:

MTTF = 960s




**شکل ۲۲: رفتار خرابی مخزن خارجی در حضور و عدم حضور خطا**

با وجود خطا در حسگر حرارتی، مخزن خارجی در ثانیه ۲۸۰ام دچار خرابی شده و موتور سیستم از این به بعد از مخزن داخلی برای سوخت‌رسانی استفاده می‌کند. اما در حالتی که خطایی وجود ندارد این اتفاق در ثانیه ۹۶۰ام اتفاق می‌افتد. بنابراین درصورتی‌که مدت زمان عملیات سیستم کمتر از ۹۶۰ باشد این مخزن در طول زمان عملیات دچار مشکلی نخواهد شد.

### تحلیل رفتار مخزن داخلی در حضور و عدم حضور خطا

نمودار رفتار این مؤلفه در شکل ۲۳ نشان داده شده است. بنابراین این مؤلفه تا زمان خرابی مخزن خارجی بیکار است. میانگین مدت‌زمان بیکاری این مخزن در حضور و عدم حضور خطا به‌صورت زیر است:

میانگین زمانی بیکاری مخزن داخلی در حضور خطا:

$F(T_1) = 50s$

میانگین زمانی بیکاری مخزن داخلی در عدم حضور خطا:

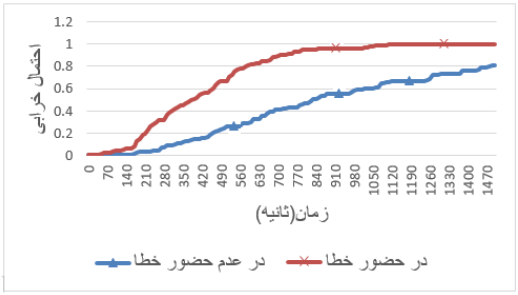
$F(T_1) = 170s$

میانگین زمانی تا خرابی مخزن داخلی در حضور خطا:

$MTTF = 540s$

میانگین زمانی تا خرابی مخزن داخلی در عدم حضور خطا:

$MTTF = 1380s$




**شکل ۲۳: رفتار خرابی مخزن داخلی در حضور و عدم حضور خطا**

### تحلیل رفتار حسگر فشار در حضور و عدم حضور خطا

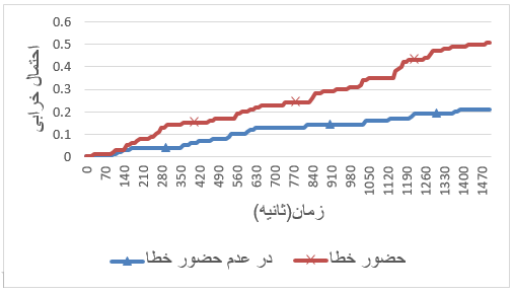
با توجه به نمودار شکل ۲۴ احتمال خرابی حسگر در ثانیه ۱۵۰۰ام به‌صورت زیر است:

احتمال خرابی حسگر فشار در ثانیه ۱۵۰۰ در حضور خطا:

$P(F) = 0.51$

احتمال خرابی حسگر فشار در ثانیه ۱۵۰۰ در عدم حضور خطا:

$P(F) = 0.21$




**شکل ۲۴: رفتار خرابی حسگر فشار در حضور و عدم حضور خطا**

### تحلیل رفتار مخزن رزرو شده در حضور و عدم حضور خطا

هنگامی مخزن رزرو شده فعالیت سوخت‌رسانی خود را آغاز می‌کند که مخزن داخلی دچار مشکل شده باشد. نمودار تحلیل نحوه خرابی این مؤلفه در شکل ۲۵ آمده است. با توجه به تحلیل این نمودار:

میانگین زمان تا شروع سوخت‌رسانی در حضور خطا:

$T = 130s$

میانگین زمان تا شروع سوخت‌رسانی در عدم حضور خطا:

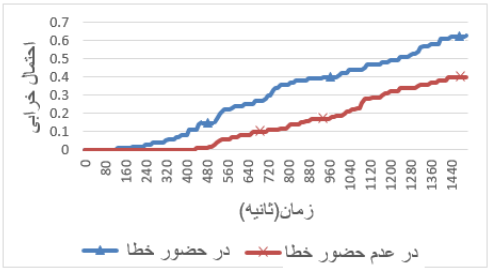
$T = 440s$

احتمال خرابی کامل سیستم در حضور خطا:

$P(F) = 0.63$

احتمال خرابی کامل سیستم در عدم حضور خطا:

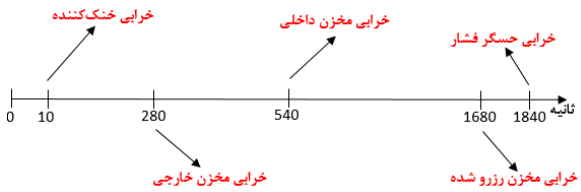
$P(F) = 0.4$




**شکل ۲۵: رفتار مخزن رزرو شده در حضور و عدم حضور خطا**

### نحوه انتشار خطا از حسگر حرارتی به سایر مؤلفه‌ها

با توجه به مشاهده رفتار هر یک از مؤلفه‌ها که در بخش قبل توضیح داده شده، انتشار خطا از مؤلفه حسگر حرارتی به سایر مؤلفه‌ها در شکل ۲۶ نشان داده شده است.




**شکل ۲۶: چگونگی انتشار خطای حسگر حرارتی در سایر مؤلفه‌ها**

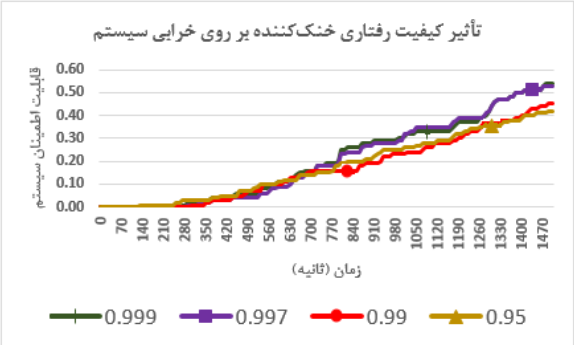
### ۵-۶- تحلیل حساسیت ورودی‌های مختلف بر نتایج مدل‌سازی

در این بخش به تحلیل حساسیت استفاده از خنک‌کننده‌ها و حسگرهای حرارتی مختلف در نتایج ارزیابی پرداخته شده است. در ادامه جزییات این دو آزمایش ذکر شده است.

#### ◘ تأثیر استفاده از خنک‌کننده‌های با دقت‌های مختلف بر قابلیت اطمینان سیستم

در این آزمایش از خنک‌کننده‌های مختلف با دقت‌های متفاوت استفاده شده است. قابلیت اطمینان کل سیستم با خنک‌کننده‌های متفاوت در شکل ۲۷ نشان داده شده است.

همانطور که در شکل ۲۷ نشان داده شده است. در صورتی که بتوانیم رفتار خنک‌کننده در صورت خراب بودن یکی از حسگرها از ۰.۹۵ به ۰.۹۹۷ افزایش دهیم، قابلیت اطمینان سیستم از ۰.۴۱ به ۰.۵۴ افزایش پیدا خواهد کرد. همین نمودار نشان می‌دهد که استفاده کردن از خنک‌کننده با دقت ۰.۹۹۹، تفاوت چندانی با خنک‌کننده با دقت ۰.۹۹۷ ندارد. از این نمودار می‌توان نتیجه گرفت که بهترین خنک‌کننده برای این سیستم خنک‌کننده‌ای با دقت ۰.۹۹۷ است.

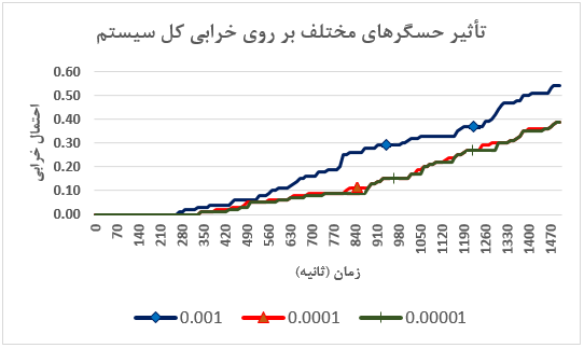



**شکل ۲۷: تأثیر کیفیت رفتاری خنک‌کننده بر روی خرابی سیستم**

#### ◘ تأثیر استفاده از حسگرهای حرارتی متفاوت بر قابلیت اطمینان سیستم

در این آزمایش از حسگرهای با نرخ خرابی متفاوت استفاده شده است. این آزمایش مشخص می‌کند که با استفاده از حسگرهای حرارتی مختلف تا چه اندازه‌ای می‌توان قابلیت اطمینان سیستم را بالا برد. نمودار مربوط به این آزمایش در شکل ۲۸ نشان داده شده است.

همانطور که در شکل ۲۸ نشان داده شده است. در صورت استفاده از حسگر با نرخ خرابی ۰.۰۰۰۱ می‌توان قابلیت اطمینان سیستم را تا حدود ۰.۶ بالا برد. استفاده کردن از حسگر بهتر مانند E-5 دیگر تأثیر چندانی بر قابلیت اطمینان کل سیستم نخواهد گذاشت.




**شکل ۲۸: تأثیر حسگرهای مختلف بر روی خرابی کل سیستم**

### ۵-۷- مقایسه روش پیشنهادی و نتیجه‌گیری

در این بخش روش‌پیشنهادی این پژوهش با مقاله‌های [۱۵, ۱۸] مقایسه شده است. در جدول ۱۶ جزییات این مقایسه را نشان می‌دهد. در ادامه مقایسه روش پیشنهادی از منظر پارامترهای مختلف با سایر روش‌ها آورده شده است.

**جدول ۱۶: مقایسه روش پیشنهادی با دو روش دیگر**

| | روش پیشنهادی | [۱۸] | [۱۵] |
|---|---|---|---|
| در نظر گرفتن تفاوت مؤلفه‌ها | ✓ | ✓ | ✓ |
| در نظر گرفتن ترتیب اجرایی | ✓ | × | × |
| در نظر گرفتن وابستگی غیرداده‌ای | ✓ | × | × |
| نیاز به وارد کردن جزییات سیستم | × | × | ✓ |
| نیاز به پیش‌بینی مسیرهای خطا | × | ✓ | × |
| سادگی مدل | ✓ | ✓ | × |
| زمان‌بر بودن ارزیابی | × | × | ✓ |
| سختی به‌دست آوردن پارامترها | ✓ | ✓ | ✓ |

#### ۵-۷-۱- سختی مدل‌سازی

در این بخش سختی مدل‌سازی روش پیشنهادی با دو روش دیگر بررسی شده است. در اینجا منظور از سختی مدل‌سازی چگونگی ساخت مدل و آماده کردن آن جهت ارزیابی است.

در مقاله [۱۸] که مثال اول از این مقاله گرفته شده بود، برای مدل‌سازی از شبکه پتری و شبکه بیزی استفاده کرده است. این مدل‌ها با توجه به ساده بودن و امکانات کم آن‌ها فرایند مدل‌سازی را سخت و پیچیده می‌کند. این امکانات شامل مکان‌های بسط‌یافته، فعالیت‌های دارای چند حالت احتمالی و غیره است، که در مدل SAN وجود دارد. بنابراین درصورتی‌که یک سیستم را با این روش مدل‌سازی شود به دلیل قابلیت‌های کم مدل، مدل بسیار بزرگ و فرایند مدل‌سازی زمان‌بر و پیچیده می‌شود. همچنین در این روش باید قبل از مدل‌سازی همه‌ی خطاها را پیش‌بینی کرد و نحوه‌ی

ترکیب آن‌ها برای انتشار در سایر مؤلفه‌ها را نیز در نظر گرفت. در روش مقاله [۱۵]، از آتاماتای هیبریدی استفاده شده بود. برای مدل‌سازی با این روش باید تمامی جزییات مدل را در نظر گرفت. این جزییات شامل فرمول‌های مربوط به هر یک از مؤلفه‌ها نیز می‌شود. به‌طور مثال برای پر شدن مخزن آب باید دقیقاً فرمول نحوه‌ی پر شدن مدل را قرار داد و عملاً باید سیستم به‌صورت واقعی مدل شود. که این روش فرایند مدل‌سازی را بسیار پیچیده می‌کند. در روش پیشنهادی این مقاله نیز به دست آوردن گراف وابستگی غیردادهای می‌تواند بسیار زمان‌بر باشد. درواقع مدل‌ساز به‌منظور مدل‌سازی سیستم‌های بزرگ، باید زمان زیادی را صرف شناسایی ارتباطات غیرمستقیم بین مؤلفه‌های سیستم کند.

**۵-۷-۲- پارامترهای در نظر گرفته شده در مدل**

در روش‌های مقاله [۱۵, ۱۸] به ارتباط مستقیم مؤلفه‌ها و نحوه‌ی ترکیب خطاها باهمدیگر پرداخته شده است. درحالی‌که پارامترهای دیگری در فرایند انتشار خطا می‌توانند مؤثر باشند. در روش پیشنهادی این مقاله علاوه بر در نظر گرفتن ارتباط مستقیم مؤلفه‌ها، پارامترهای دیگری در نظر گرفته‌شده‌اند که این پارامترها عبارت‌اند از ترکیب احتمالی خطاها و وابستگی غیردادهای که بین مؤلفه‌ها وجود دارد. همچنین در این روش ترتیب اجرایی مؤلفه‌ها در نظر گرفته شده است. یک مؤلفه ممکن است به‌طور مداوم در حال کار نباشد بلکه بعد از به اتمام رسیدن فعالیت یک مؤلفه دیگر، کار آن شروع شود. البته در مقاله [۱۶] احتمال فعالیت مؤلفه‌ها بعد از اتمام کار هر مؤلفه در نظر گرفته شده است.

**۵-۷-۳- نتایج به دست آمده از مدل**

در روش پیشنهادی این مقاله می‌توان نتایج متنوعی از اجرای مدل به دست آورد که این نتایج عبارت‌اند از:

- تأثیر فعال شدن یک خطا در یک مؤلفه خاص بر رفتار خرابی سایر مؤلفه‌ها
- ارزیابی قابلیت اطمینان کل سیستم
- احتمال خرابی هر مؤلفه در هرلحظه
- مشاهده مقادیر ورودی و خروجی مؤلفه‌ها در هرلحظه
- ترتیب خرابی مؤلفه‌ها و نحوه انتشار تأثیرات خطا بر سایر مؤلفه‌ها
- پیدا کردن مؤلفه‌های حیاتی سیستم
- مشخص کردن مؤلفه‌های مناسب به منظور طراحی سیستم

در سایر روش‌های پیشنهاد شده در این حوزه مشاهده نحوه تأثیر یک خطا بر ورودی و خروجی سایر مؤلفه‌ها در هرلحظه و همچنین مشاهده رفتار خرابی تمامی مؤلفه‌ها یا غیرممکن است یا باید مدل بسیار پیچیده‌تر شود تا این پارامترها را مشاهده کرد.

**۵-۷-۴- سختی ارزیابی و به دست آوردن نتایج**

در روش پیشنهادی مقاله [۱۵] برای هر بار اجرای مدل و مشاهده نحوه‌ی انتشار خطا، حدود ۲۲ ساعت زمان لازم است که برای مدل‌سازی و به دست آوردن نتایج زمان بسیار زیادی است. درحالی‌که روش ارائه شده در این مقاله پس از ساخت مدل، حدود ۲ دقیقه زمان بر روی سیستم یاد شده جهت اجرای مدل و به دست آوردن نتایج زمان لازم دارد.

**۵-۷-۵- جامعیت و مقیاس‌پذیری**

با توجه به رابط‌های ارائه شده برای مؤلفه‌های سیستم، هر مؤلفه‌ای شامل تعدادی ورودی و تعدادی خروجی است و نحوه نگاشت ورودی‌ها به خروجی‌ها را نیز می‌توان با عبارت‌های احتمالی مشخص کرد. مشخص است که هر مؤلفه‌ای در این ساختار می‌تواند قرار بگیرد.

با توجه به اینکه ارتباط تمامی مؤلفه‌ها در مدل ارائه شده این مقاله بر اساس ورودی و خروجی و ارتباط غیردادهای است، درصورتی‌که چندین سیستم وجود داشته باشند که باهمدیگر ارتباط داشته باشند و بتوان آن‌ها را کنار هم قرار داد، به‌راحتی می‌توان ارتباط بین آن‌ها را با توجه به قوانین گفته‌شده فراهم کرد و مدل سیستم ترکیبی، متشکل از چند زیرسیستم را ایجاد نمود. بنابراین می‌توان نشان داد که مدل ارائه شده مقیاس‌پذیر نیز است.

**۵-۷-۶- جمع‌بندی مقایسه روش پیشنهادی**

درنهایت همانطور که در بخش‌های قبل توضیح داده شد، می‌توان گفت مدل پیشنهادی در این مقاله دارای روش مدل‌سازی ساده‌تری نسبت به دو مدل دیگر است. همچنین، پارامترهایی مانند وابستگی غیردادهای در مدل پیشنهادی در نظر گرفته شده است که در مدل‌های قبلی دیده نشده بود. زمان اجرای مدل پیشنهادی در مقایسه با روش مقاله [۱۵] بسیار پایین‌تر است که در بخش قبلی مقایسه شدند. ویژگی مهم مدل پیشنهادی مقیاس‌پذیری آن است که می‌توان هریک مؤلفه‌ها و حتی سیستم‌ها را به صورت جداگانه مدل‌سازی کرد و سپس مدل‌ها را به راحتی با استفاده از مدل تکرار-الحاق کنار هم قرارداد و به ارزیابی رفتاری آن‌ها در کنار یکدیگر پرداخت. همچنین همانطور که در بخش قبل توضیح داده شد با استفاده از این روش می‌توان نتایج متنوع‌تری را به دست آورد.

**۵-۸- کارهای آینده**

به‌منظور بهبود در روش پیشنهادی، موضوعات و عناوین زیر برای تحقیق‌های آتی و کارهای آینده پیشنهاد می‌شود:

- روشی برای خودکارسازی فرایند ساخت مدل به‌صورتی که با ارائه گراف‌های سه‌گانه بتوان مدل مربوط به انتشار خطا سیستم را استخراج کرد.
- بهبود روش پیشنهادی با در نظر گرفتن پارامترهایی همچون احتمال کارکرد مؤلفه‌های مختلف سیستم در زمان‌های مختلف.
- آزمایش روش پیشنهادی در دنیای واقعی و بررسی نتایج آن با داده‌های واقعی به منظور اعتبارسنجی مدل و بهبود روش پیشنهادی.

## مراجع


[1] R. Alur, Principles of Cyber-Physical Systems, Massachusetts: MIT Press, 2015.

[2] S. Seshia and E. Lee, Introduction to Embedded Systems - A Cyber-Physical Systems Approach, MIT Press, 2017.

[3] M. Fan, Z. Zeng, E. Zio, R. Kang and Y. Chen, "A stochastic hybrid systems model of common-cause failures of degrading components," *Reliability Engineering & System Safety*, vol. 172, pp. 159-170, 2018.

[4] R. Kang and Z. Li, "Strategy for reliability testing and evaluation of cyber physical systems," in *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, Dec 2015.

[5] G. Simko, T. Levendovszky, M. Maroti and J. Sztipanovits, "Towards a theory for cyber-physical systems modeling," in *Proceedings of the 4th ACM SIGBED International Workshop on Design, Modeling, and Evaluation of Cyber-Physical Systems*, Berlin, April 2014.

[6] R. Michael and P. Liggesmeyer, "Modeling and analysis of safety-critical cyber physical systems using state/event fault trees," in *International Conference on Computer Safety, Reliability and Security*, Toulouse, Sep 2013.

[7] A. Avizienis, J.-C. Laprie, B. Randell and C. Landwehr, "Basic Concepts and Taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11-33, 2004.

[8] W. H. Sanders and J. F. Meyer, "Stochastic activity networks: formal definitions and concepts," in *Lectures on formal methods and performance analysis*, New York, Springer, 2001, pp. 315 - 343.

[9] M. Rahnamay Naeini and M. M. Hayat, "Cascading Failures in Interdependent Infrastructures: An Interdependent Markov-Chain Approach," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 1997-2006, 2016.

[10] R. A. Shuvro, Z. Wangt , P. Das, M. R. Naeini and M. M. Hayat, "Modeling cascading-failures in power grids including communication and human operator impacts," in *IEEE Green Energy and Smart Systems Conference*, Long Beach, Nov 2017.

[11] S. V. Buldyrev, R. Parshani, G. Paul, H. Stanley and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, pp. 1025-1028, 2010.

[12] Z. Zuyuan, W. An and S. Fangming, "Cascading Failures on Reliability in Cyber-Physical System," *IEEE Reliability Society*, vol. 65, no. 4, pp. 1745 - 1754, 2016.

[13] Z. Huang and C. Wang, "Characterization of Cascading Failures in Interdependent Cyber-Physical Systems," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2158-2168, 2015.

[14] H. Peng, Z. Kan, D. Zhao, J. Han, J. Lu and Z. Hu, "Reliability analysis in interdependent smart grid systems," *Physica A: Statistical Mechanics and its Applications*, vol. 500, pp. 50-59, 2018.

[15] C. Heracleous, M. M.Polycarpou, G. Ellinas, C. G.Panayiotou and P. Kolios, "Hybrid systems modeling for critical infrastructures interdependency analysis," *Reliability Engineering & System Safety*, vol. 165, pp. 89-101, 2017.

[16] A. Morozov and K. Janschek, "Probabilistic error propagation model for mechatronic systems," *Mechatronics*, vol. 24, no. 8, pp. 1189-1202, 2014.

[17] A. Morozov and K. Janschek, "Dual Graph Error Propagation Model for Mechatronic System Analysis," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 9893-9898, 2011.

[18] S. Kabir, M. Walker and Y. Papadopoulos, "Dynamic system safety analysis in HiP-HOPS with Petri Nets and Bayesian Networks," *Safety Science*, vol. 105, pp. 55-70, 2018.

[19] S. Kabir, Y. Papadopoulos, M. Walker, D. Parker, J. Ignacio Aizpurua, J. Lampe and E. Rüde, "A Model-Based Extension to HiP-HOPS for Dynamic Fault Propagation Studies," in *5th International Symposium on Model-Based Safety and Assessment*, Aug 2017.

[20] M. Walker, L. Bottaci and Y. Papadopoulos, "Compositional Temporal Fault Tree Analysis," in *International Conference on Computer Safety, Reliability, and Security*, 2007.

[21] E. Edifor, M. Walker, N. Gordon and Y. Papadopoulos, "Using simulation to evaluate dynamic systems with weibull or lognormal distributions," in *Proceedings of the Ninth International Conference on Dependability and Complex Systems*, Brunow, June 2014.

[22] L. Grunske and B. Kaiser, "Automatic generation of analyzable failure propagation models from component-level failure annotations," in *Fifth International Conference on Quality Software*, Melbourne, Sep 2005.

[23] C. Zhou, X. Huang, X. Naixue, Y. Qin and S. Huang, "A class of general transient faults propagation analysis for networked control systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 4, pp. 647 - 661, 2015.

[24] Y. Liu, D. Lu, L. Deng, T. Bai, K. Hou and Y. Zeng, "Risk assessment for the cascading failure of electric cyber-physical system considering multiple information factors," *IET Cyber-Physical Systems: Theory & Applications*, vol. 2, no. 4, pp. 155 - 160, 2017.

[25] X. Ge, R. F. Paige and J. A. McDermid, "Probabilistic Failure Propagation and Transformation Analysis," in *28th International Conference on Computer Safety, Reliability, and Security*, Berlin, 2009.


## زیرنویس‌ها:

[1] Hybrid Systems
[2] Fault
[3] Propagation
[4] Reliability
[5] Scalable
[6] Topology
[7] Stochastic Activity Networks
[8] Replication/Join
[9] Pathology
[10] Marcov chain
[11] Cyber-Physical Systems
[12] Strategy
[13] failure propagation and transformation analysis
[14] Extended Place
[15] Converge