

## مدلسازی منطقی کامپیوتر پرواز برای تشخیص عملکرد ناصحیح عملگرهای برقی - مکانیکی پرنده‌های بدون سرنشین با استفاده از شبکه‌های پتری

علیرضا آهنگرانی فراهانی<sup>۱</sup>، عباس دیدبان<sup>۲</sup>

۱ دانشجوی دکتری، دانشکده برق و کامپیوتر، دانشگاه سمنان، سمنان

۲ دانشیار، دانشکده برق و کامپیوتر، دانشگاه سمنان، سمنان، adideban@semnan.ac.ir

تاریخ دریافت: ۱۳۹۶/۰۷/۱۱

تاریخ پذیرش: ۱۳۹۷/۰۶/۱۳

### چکیده

در این مقاله با استفاده از ابزار شبکه‌های پتری روشی برای مدلسازی عملیات منطقی، کشف خطا، طراحی کنترل‌کننده نظارتی و در نتیجه افزایش قابلیت اطمینان برای زیرسیستم عملگر برقی - مکانیکی پرنده‌های بدون سرنشین ارائه شده است. در پرنده‌های بدون سرنشین عملگرها برای کنترل سطوح کنترلی استفاده می‌شود، لذا ایجاد روشی که بتواند رفتار منطقی و خطاهای زیرسیستم‌ها را مدل کند و رفتار مناسبی در مقابل خطاهای احتمالی سیستم داشته باشد، بسیار مفید خواهد بود. در روش ارائه شده در این مقاله، جهت طراحی کنترل‌کننده نظارتی بر مبنای شبکه‌های پتری سه گام اصلی برداشته شده است: در اولین گام، ضوابط و خصوصیات اصلی بخش‌های مختلف با استفاده از شبکه پتری به صورت جداگانه مدل شده است، در گام دوم سنکرون‌سازی بین مدل‌ها انجام شده و در گام آخر با استفاده از کنترل نظارتی، کل رویه کنترل جهت حصول اطمینان از بروز وضعیت ناامن در سیستم ضمانت می‌گردد. در این پژوهش جهت جلوگیری از ورود در شرایط ناامن و همچنین کنترل فرایند از روش افزونگی تحلیلی استفاده شده است. نتایج نشان می‌دهد که با استفاده از این روش احتمال خرابی کاهش و قابلیت اطمینان سیستم افزایش خواهد یافت.

### واژگان کلیدی

شبکه‌های پتری، کنترل نظارتی، هواپیمای بدون سرنشین، عملگرهای برقی - مکانیکی، تشخیص خطا

### ۱. مقدمه

ایمنی انجام شود. برای دستیابی به این هدف، در فرایند طراحی باید پارامترهایی چون وزن، قابلیت اطمینان، استحکام، دقت و سرعت در کنترل مورد نظر قرار گیرد. یکی از مهمترین بخش‌های پرنده‌های بدون سرنشین، کامپیوتر پرواز<sup>۱</sup> است که وظیفه نظارت

امروزه استفاده از هواپیماهای بدون سرنشین برای عملیات غیرنظامی در حوزه‌هایی چون کشاورزی، نقشه‌برداری و عملیات نظامی مثل تجسس و شناسایی در حال گسترش است. بنابراین عملیات مورد نظر باید مطمئن، راحت، دقیق، ارزان و در کمال

جریان مدلسازی جلوگیری می‌شود. از اینرو، در حال حاضر این روش به یکی از مهمترین ابزارها برای مدلسازی و کنترل سیستم‌های گسسته - پیشامد تبدیل شده است [۶]. شبکه‌های پتری با در اختیار قرار دادن ابزار قدرتمند گرافیکی و ریاضی به طراح اجازه می‌دهد تا بدون در نظر گرفتن رفتار و معادلات پیچیده حاکم بر زیرسیستم‌ها به‌طور دقیق بر روند عملکرد نظارت کند. سپس با داشتن نظارت کافی بر عملکرد سیستم تصمیمات و سناریوهای مختلف را اتخاذ نماید. بنابراین با توجه به مزیت‌های ابزار شبکه‌های پتری بسیاری از محققان به‌سمت این نوع مدلسازی گرایش پیدا کرده‌اند.

در مسائل هوایی برای فرمان‌دهی به سطوح کنترلی از عملگرهای متفاوتی از جمله عملگرهای هیدرولیکی، پنوماتیکی یا الکترومکانیکی استفاده می‌شود. هرچند قابلیت عملکردی این نوع عملگرها اثبات شده، اما امروزه با توجه به اهمیت وزن حداقلی وسیله، قابلیت ساده برنامه‌نویسی، دقت و کارایی و افزایش قابلیت اطمینان استفاده از سرو موتورهای الکتریکی متداول شده است. یکی از چالش‌های اصلی در هواپیماهای بدون سرنشین، مدلسازی رفتار منطقی و تشخیص خطاها در عملگرهای برقی - مکانیکی و مقابله با آنهاست. در عملگرهای برقی - مکانیکی، خطا بنا به دلایل نرم‌افزاری مانند عدم دریافت داده مناسب از کامپیوتر پرواز یا سخت‌افزاری مانند آسیب‌دیدگی سنسورها یا راه‌اندازها رخ می‌دهد [۷]. در سال‌های اخیر اغلب تحقیقات صورت‌گرفته جهت تشخیص و جلوگیری از خطا در عملگرها، براساس مقایسه داده‌های پروازی و مدل دینامیکی آنها بوده است. بنابراین، واحد کامپیوتر پرواز همواره پس از وقوع خطا یا نقص به مقابله با آن می‌پردازد [۸-۹]. معمولاً در این روش‌ها شناسایی خطا براساس روش‌های تخمین پارامترها، فیلتر کالمن یا روش‌های هوشمندی مانند شبکه‌های عصبی صورت می‌گیرد [۱۰]. در این شرایط پیکربندی مجدد کنترل‌کننده برای شرایط غیرایمن یا رفتن به حالت عادی یا خروج از عملیات مدنظر برای سیستم دشوار است [۱۱]. ضمن آنکه در اکثر تحقیقات صورت‌گرفته، تشخیص خطا مبتنی بر ارزیابی سخت‌افزار در مسائل هوایی است [۱۲-۱۳].

پس از تشخیص خطا، نیاز به مقابله با خطا وجود دارد که در برخی موارد با ایجاد افزونگی سخت‌افزاری صورت پذیرفته است. ایجاد سخت‌افزار معمولاً سبب افزایش هزینه‌های طراحی و ساخت می‌شود [۹]. حال آنکه ابزار شبکه پتری کنترل نظارتی یا

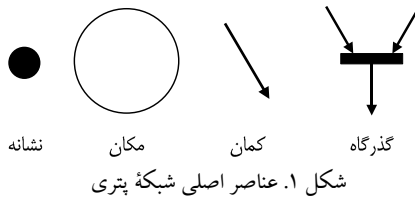
و کنترل ایمنی وسیله را برعهده دارد. به‌طور کلی کامپیوتر پرواز دارای دو وظیفه اصلی مدیریت پرواز و مدیریت مأموریت است. بخش مدیریت پرواز پیاده‌سازی عملیات‌های هدایت، کنترل، داده‌برداری و وظایفی از این قبیل و بخش مدیریت مأموریت، وظیفه مدیریت و نظارت بر سلامت زیرسیستم‌ها را برعهده دارند. با افزایش سطح خودکاری و هوشمندی واحد مدیریت مأموریت، عملیات‌های نظارتی و تصمیم‌سازی بر عملکرد کل سیستم، در این واحد صورت می‌گیرد. در نتیجه، با افزودن الگوریتم‌های مناسب مقابله با خطا به واحد مدیریت مأموریت، قابلیت عملیاتی وسیله در شرایط بروز عدم قطعیت و نقص، ارتقا می‌یابد [۱].

یکی از مباحث مهم در طراحی واحد مدیریت مأموریت تشخیص خطا و نحوه عملکرد سیستم در هنگام وقوع خطاست. خطا رویدادی است که سبب تغییر در رفتار مطلوب سیستم شده و در قسمت‌های مختلف سیستم رخ می‌دهد. سه خطای اصلی در پرنده خطای عملگر، خطای سنسور و خطای اجزا در فرایند است. جهت کشف و مقابله با خطا، کامپیوتر پرواز از عملیات‌های منطقی ناظر بر رفتار پیشامدی زیرسیستم‌ها و نحوه تعامل آنها با هم استفاده می‌کند. بنابراین مدلسازی عملیات منطقی و خطاهای زیرسیستم‌ها قبل از پیاده‌سازی در کامپیوتر پرواز سبب شناخت دقیق از آنها شده و به طراح اجازه طراحی کنترل‌کننده مناسب را می‌دهد. برای مدلسازی این رفتارها می‌توان از روش‌های گسسته - پیشامد<sup>۲</sup> استفاده کرد [۲]. استفاده از ابزارهای گسسته - پیشامد برای مدلسازی سیستم‌ها با خواص گسسته در حال گسترش است. در تعریف سیستم‌های گسسته - پیشامد می‌توان گفت سیستم‌هایی که دارای حالت‌های گسسته بوده و با وقوع هر پیشامد از حالتی به حالت دیگر تغییر وضعیت می‌دهند [۳]. دینامیک این سیستم‌ها از تغییر حالت‌ها و وقوع پیشامدها تشکیل می‌شود. با در اختیار داشتن مدل سیستم گسسته - پیشامد و ضوابط رفتاری آن، می‌توان یک ناظر مناسب جهت کنترل حلقه بسته ایجاد نمود.

در منابع علمی، ابزارهای متنوعی برای مدلسازی عملیات منطقی کامپیوتر پرواز با استفاده از روش گسسته - پیشامد مانند دیاگرام حالت، اتوماتا<sup>۳</sup> و شبکه‌های پتری<sup>۴</sup> ارائه شده است [۴]. از بین روش‌های یادشده، روش اتوماتا و دیاگرام حالت با افزایش تعداد متغیرهای سیستم، دچار انفجار حالت می‌شوند که این مسئله منجر به پیچیده‌شدن مدلسازی سیستم می‌گردد [۵]. اما ساختار شبکه‌های پتری به‌گونه‌ای است که از وقوع چنین پدیده‌ای در

آورده شده است [۱۵]. شبکه پتری یک مجموعه پنج‌عضوی است که به صورت زیر تعریف می‌شود [۱۶]:

$$N = \{P, T, W^-, W^+, M_0\} \quad (1)$$



در مجموعه بالا  $P$  مجموعه متناهی و غیرتهی از مکان‌ها،  $T$  مجموعه متناهی و غیرتهی از گذرگاه‌ها،  $W^-$  تابع ورودی بیان‌کننده مکان‌های ورودی به گذرگاه‌ها،  $W^+$  تابع خروجی بیان‌کننده مکان‌های خروجی از گذرگاه‌ها و  $M_0$  حالت اولیه سیستم و بیانگر تعداد نشانه‌های اولیه در مکان‌هاست. اگر تعداد مکان‌ها برابر با  $|P|$  و تعداد گذرگاه‌ها برابر با  $|T|$  در نظر گرفته شود، توابع ورودی و خروجی که ارتباط بین مکان‌ها و گذرگاه‌ها را مشخص می‌کند، ماتریس‌های  $|T| \times |P|$  است. اگر  $i$  مکان ورودی به گذرگاه  $j$  باشد، در این صورت آرایه سطر  $i$  ام و ستون  $j$  ام ماتریس  $W^-$  برابر با وزن کمان رابط مکان  $i$  و گذرگاه  $j$  است. به طریق مشابه اگر مکان  $i$  یک مکان خروجی از گذرگاه  $j$  باشد، آرایه سطر  $i$  ام و ستون  $j$  ام ماتریس  $W^+$  برابر با وزن کمان رابط گذرگاه  $j$  و مکان  $i$  است.

یکی دیگر از مفاهیم مهم در شبکه‌های پتری ماتریس تلافی<sup>۹</sup> است که به صورت  $W = W^+ - W^-$  تعریف می‌شود و تمام عملیات ریاضی روی این ماتریس صورت می‌گیرد [۱۷]. در شبکه پتری گسسته معادله حالت یا معادله اساسی عبارت است از [۱۸]:

$$M_k = M_0 + W \cdot U \quad (2)$$

$$U = \{u_0, u_1, u_2, u_3, \dots, u_k\} \quad (3)$$

که در آن برداری با ابعاد  $n \times 1$  و نشان دهنده وضعیت در لحظه  $k$  ام است. در اینجا  $n$  مشخص‌کننده تعداد حالت‌ها یا مکان‌های سیستم است. در معادله ۲ بردار  $U$  بردار  $m \times 1$  نشان‌دهنده آتش‌شدن گذرگاه‌هاست. همچنین  $m$  برابر با تعداد گذرگاه‌هاست.

## ۲-۲. مدل ضوابط

در سیستم‌های واقعی جهت ارتباط بین اجزا و زیرسیستم‌ها ضوابط و شرایطی حاکم است. گاهی مدلسازی با شبکه‌های پتری ضوابط و قواعد مورد نیاز بین اجزا و زیرسیستم‌های یک سیستم را بسیار

کنترل پروسه‌ای را در اختیار کامپیوتر پرواز قرار می‌دهد. کنترل نظارتی باعث می‌شود تا پیش از وقوع خطا آن را کشف کرده و از تبدیل شدن خطاهای محلی به خرابی‌ها و در نتیجه توقف فرایند جلوگیری کند [۱۴].

هدف این مقاله آن است که با استفاده از ابزار شبکه‌های پتری، بخشی از رفتار منطقی زیرسیستم‌ها (عملگرها) در کامپیوتر پرواز مدلسازی شود. در این مدلسازی بدون در نظر گرفتن رفتارهای دینامیکی پیچیده زیرسیستم، رفتار منطقی، حالت‌ها و خطاهای آن مدل می‌شود. سپس با در اختیار داشتن حالت سیستم و اعمال کنترل مناسب می‌تواند وضعیت غیرایمن را قبل از ورود عملگر به آن ناحیه تشخیص دهد. بدین ترتیب، سیستم کامپیوتر پرواز فرصت دارد تا با این وضعیت مقابله نماید و در نتیجه قابلیت اطمینان مجموعه افزایش پیدا می‌کند. روشی که در این پژوهش برای افزایش بهبود احتمال خرابی عملگرها به صورت نرم‌افزاری استفاده شده، روش افزونگی تحلیلی است. برای نشان‌دادن قابلیت روش پیشنهادی، مدلسازی براساس داده‌های یک عملگر فرضی با استفاده از ابزار شبکه‌های پتری انجام شده‌است. بررسی نتایج حاصل نشان می‌دهد که در صورت اضافه کردن الگوریتم‌های کنترلی به مدل اولیه، احتمال کشف خرابی افزایش می‌یابد. روند مقاله در بخش‌های بعدی بدین شرح است که بخش دوم مفاهیم پایه مدلسازی شبکه‌های پتری و کنترل نظارتی آورده شده‌است. در بخش سوم مدلسازی دقیقی از بخش‌های مختلف عملگر ارائه و سپس مدل ترکیب شده نهایی عملگر ارائه شده است. در ادامه در بخش چهارم مدلسازی نهایی عملگر ارائه و کنترل‌کننده مناسب طراحی شده است. بخش پنجم مربوط به ارائه نتایج حاصل از اجرای الگوریتم مورد نظر و مقایسه با حالت رایج است. در انتها نتیجه‌گیری و پیشنهاداتی برای ادامه کار بیان خواهد شد.

## ۲. مفاهیم پایه

در ابتدا برای طراحی و پیاده‌سازی رفتار منطقی سروعملگرها باید ابزار مدلسازی؛ یعنی شبکه‌های پتری و تعاریف آن معرفی شوند.

### ۲-۱. شبکه‌های پتری

یکی از ابزارهای بسیار قدرتمند ریاضی و گرافیکی جهت مدلسازی سیستم‌ها با رفتار گسسته - پیشامد، شبکه‌های پتری است. شبکه‌های پتری چهار عنصر اصلی دارند که عبارت‌اند از مکان<sup>۵</sup>، گذرگاه<sup>۶</sup>، نشانه<sup>۷</sup> و کمان<sup>۸</sup> که در شکل ۱ نمایشی از این عناصر

فیزیکی از طریق افزایش قطعات و ادوات، تحمل‌پذیری سیستم را افزایش می‌دهد. در حالی‌که افزونگی تحلیلی به‌معنای استفاده از روابط تابعی جهت بالا بردن قابلیت اطمینان سیستم است [۲۱]. در ادامه عملگر و افزونگی تحلیلی در نظر گرفته شده، با استفاده از ابزار شبکه پتری به‌طور کامل مدلسازی می‌شوند.

### ۳. مدل عملگر الکترومکانیکی

در این بخش با استفاده از ابزار شبکه پتری مدلی از رفتار منطقی عملگر ارائه خواهد شد. در اینجا عملکرد منطقی و خطاهای عملگر بدون در نظر گرفتن رفتارهای پیچیده دینامیکی مدلسازی می‌شوند. از دیدگاه کلی عملگرهای الکترومکانیکی دارای چهار بخش اصلی موتور سروی الکتریکی، راه‌انداز که با کامپیوتر پرواز مرتبط است، بخش پردازشگر و بافر ورودی - خروجی است. در شکل ۴ نمایی از آن مشاهده می‌شود [۲۲]. در اینجا بافرهای ورودی - خروجی به‌عنوان ضوابط در نظر گرفته شده است و همچنین موتور، راه‌انداز و پردازشگر مدل اصلی هستند. با مدلسازی این اجزا و ترکیب این مدل‌ها، مدل نهایی استخراج خواهد شد.

### ۳-۱. مدل پتری سروموتور الکتریکی

مدلسازی رفتار منطقی سروموتور الکتریکی به‌راحتی با استفاده از ابزار شبکه پتری صورت می‌پذیرد. این رفتار منطقی را می‌توان با سه حالت بی‌کار ( $p_1$ )، راستگرد ( $p_2$ ) یا چپگرد ( $p_3$ ) مدلسازی کرد. به‌عبارتی یا موتور بیکار است یا به یک جهت در حال چرخش است. در اینجا موتور الکتریکی دستور ارسال شده از کامپیوتر پرواز را اجرا می‌نماید. معمولاً در پرنده‌های بدون سرنشین عملگرها در یک محدوده از پیش تعیین‌شده قادر به حرکت‌اند که این محدوده را می‌توان با یک میکروسوئیچ به‌صورت سخت‌افزاری محدود نمود. شکل ۵ مدل پتری رفتار منطقی سروموتور را نشان می‌دهد.

### ۳-۲. مدل پتری راه‌انداز

راه‌انداز یک بخش بسیار مهم در عملگر محسوب است که دو عمل اصلی ارتباط با کامپیوتر پرواز جهت دریافت و ارسال داده‌ها و کنترل عملگر را برعهده دارد. راه‌انداز، داده‌ها را از کامپیوتر پرواز دریافت و صحت آنها را با استفاده از الگوریتم‌های صحت‌سنجی بررسی می‌کند ( $p_5$ ). اگر داده‌ها نامعتبر بود، راه‌انداز آن را به کامپیوتر پرواز اطلاع می‌دهد ( $p_6$ )، اما اگر داده دریافت‌شده معتبر باشد تصدیق دریافت داده را

گویاتر و خلاصه‌تر بیان می‌نماید. به‌صورت ویژه وقتی توالی وقوع پیشامدها توسط زیرسیستم‌ها وجود دارد، مدلسازی ضوابط با استفاده از شبکه‌های پتری بسیار کارساز است و بیان آن را بسیار ساده و گویا می‌نماید [۱۵].

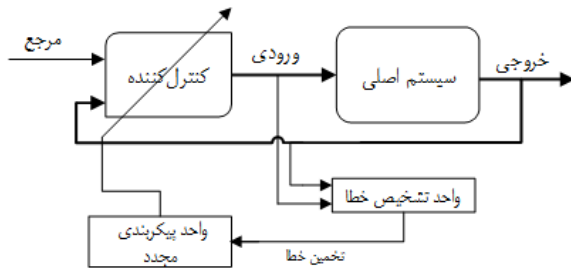
### ۲-۳. کنترل نظارتی<sup>۱۰</sup>

شرایط و ضوابط که در سیستم‌های گسسته - پیشامد وجود دارد، باعث محدودشدن حالت‌ها و قابلیت‌های سیستم می‌شود. بنابراین باید در مدلسازی و طراحی کنترل‌کننده مورد توجه قرار گیرد. نظریه کنترل نظارتی برای طراحی کنترل‌کننده در سیستم‌های گسسته - پیشامد ابداع شد. هدف نهایی از آن، طراحی کنترل‌کننده‌ای است که رفتار سیستم را محدود کرده و در چارچوب ضوابط مورد نظر (مطلوب) قرار دهد. با اعمال طراحی کنترل نظارتی بر سیستم گسسته - پیشامد می‌توان اطمینان داشت که رفتار سیستم کنترل شده با حداکثر درجه آزادی ممکن به رفتار مورد نظر و مطلوب نزدیک است. برای طراحی کنترل‌کننده مناسب باید هر یک از اجزا و ضوابط بین آنها به‌صورت جداگانه مدلسازی و با هم ترکیب شوند. در تئوری کنترل نظارتی نقش اصلی کنترل برعهده ناظر است. همان‌گونه که در شکل ۲ مشاهده می‌شود، ناظر خروجی‌ها را از سیستم دریافت کرده و با توجه به ضوابط تعریف‌شده باید تصمیم بگیرد که چه کاری در آینده مجاز یا غیرمجاز است. در اصل در اینجا ناظر تعیین‌کننده این است که کدام گذرگاه‌ها می‌توانند فعال شوند [۱۹]. استفاده از روش کنترل نظارتی خطای انسانی را به‌صورت چشمگیری کاهش داده و در نتیجه باعث دستیابی به رفتار مطلوب می‌شود.

### ۲-۴. روش فعال در بالابردن قابلیت اطمینان

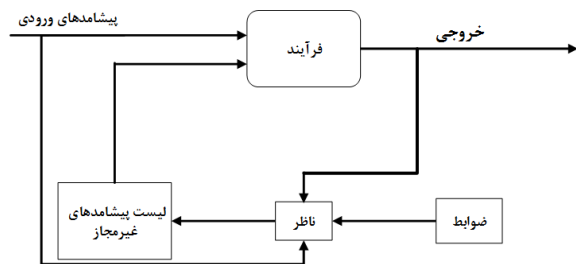
اجزای اصلی سیستم تحمل‌پذیر خطا در شکل ۳ آورده شده است [۲۰]. در این روش واحد تشخیص خطا بر مبنای ورودی‌ها و خروجی‌های سیستم، خطا را تشخیص داده و سپس به واحد پیکربندی مجدد گزارش می‌دهد. واحد پیکربندی مجدد کارایی کنترل‌کننده را پس از به‌وجود آمدن خطا در سطح قابل قبولی قرار می‌دهد. در نهایت کنترل‌کننده با اعمال کنترل مناسب، سیستم را در شرایط ایمن حفظ می‌نماید. یکی از راه‌های مقابله با خطاهای احتمالی ایجاد افزونگی در سیستم است. افزونگی به دو دسته تقسیم می‌شود: افزونگی فیزیکی و افزونگی تحلیلی. افزونگی

خطاست. بدین ترتیب راهانداز و کامپیوتر پرواز به راحتی از خطا مطلع شده و می‌توانند سناریوهای جدید را اجرا کنند ( $p_{12}$  نشان‌دهنده حالتی است که موتور باید ساعتگرد حرکت می‌کرده و زمان از محدوده مورد نظر تجاوز کرده و  $p_{16}$  نشان‌دهنده حالتی است که موتور باید پادساعتگرد حرکت می‌کرده و زمان از محدوده مورد نظر تجاوز کرده است). در اینجا حالت‌های  $p_{12}$  و  $p_{16}$  نشان‌دهنده افزونگی تحلیلی است که به مدل اضافه شده تا بتواند خرابی را تشخیص داده و اعمال سیگنال کنترلی لازم را داشته باشد. معمولاً راهاندازها مجهز به سنسور اندازه‌گیری جریان اثر هال هستند.

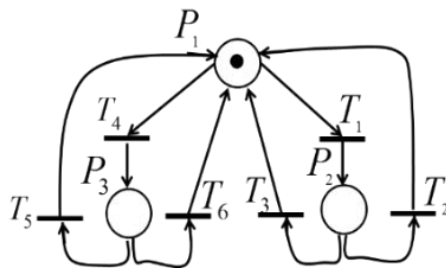


شکل ۳. بلوک دیاگرام کلی روش فعال در بالابردن قابلیت اطمینان

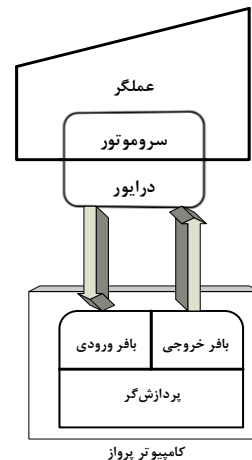
به کامپیوتر پرواز ارسال می‌کند ( $p_7$ ). پس از اطمینان از صحت داده، راهانداز دستور ارسال شده را تحلیل کرده که خروجی تحلیل دستور گردش محور موتور است ( $p_8$ ). سپس دستور حرکت به موتور ارسال شده و به‌طور همزمان زمان سنج را هم راهاندازی می‌کند (در اینجا  $p_9$  نشان‌دهنده وضعیت حرکت ساعتگرد و آغاز به‌کار زمان سنج و  $p_{13}$  نشان‌دهنده حرکت پادساعتگرد و آغاز به‌کار زمان سنج است). آغاز به‌کار زمان سنج یکی از راه‌های تشخیص خطاست؛ زیرا اگر حرکت دورانی محور موتور در طی مدت زمان محدود انجام نگیرد، به معنای این است که موتور وظیفه خود را به خوبی انجام نداده، بنابراین نشان‌دهنده



شکل ۲. بلوک دیاگرام تئوری کنترل نظارتی



شکل ۵. مدل منطقی سروموتور با استفاده از شبکه پتری



شکل ۴. شماتیک ارتباط عملگر با کامپیوتر پرواز

کامپیوتر پرواز آن را به راحتی تشخیص می‌دهند ( $p_{20}$ ). شکل ۶ مدلسازی راهانداز با استفاده از شبکه پتری را نشان می‌دهد. توصیف دقیق حالت‌های مدل پتری راهانداز در جدول ۳ آورده شده است. در ادامه تمامی گذرگاه‌ها که نشان‌دهنده پیشامد است برای مدل منطقی راهانداز در جدول ۴ آمده است. با وارد کردن مدل اندازه‌گیری جریان و اندازه‌گیری زمان به مدل اصلی و ارائه کنترل‌کننده نظارتی مناسب در صورت بروز خطا، احتمال ورود عملگر به شرایط غیرایمن کم می‌شود.

استفاده از این اندازه‌گیری می‌تواند خطاهای احتمالی را تشخیص داده و با طراحی کنترل‌کننده مناسب با آنها مقابله نماید. بنابراین یکی دیگر از افزونگی‌های تحلیلی جهت بالابردن قابلیت اطمینان سیستم، اندازه‌گیری جریان موتورها و طراحی کنترل نظارتی مناسب است. در اینجا اندازه‌گیری جریان راهاندازی با مکان ( $p_{17}$ ) نشان داده شده که در صورت تجاوز از مقدار مجاز خطا آشکار می‌شود و این خطا برای راهانداز و کامپیوتر ارسال می‌شود ( $p_{18}$ ). حال آنکه اگر خطایی در جریان راهاندازی وجود نداشت این سنسور به‌طور مداوم جریان موتور را اندازه گرفته ( $p_{19}$ ) و در حالی که در خطایی در جریان نامی رخ دهد، راهانداز و

جدول ۱. توصیف مکان‌های شکل ۵

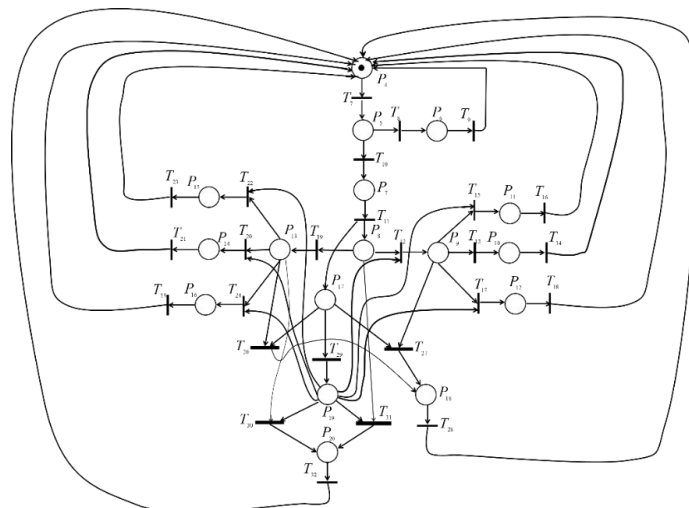
مکان	توصیف مکان
$p_1$	موتور در حال بی‌کار قرار دارد
$p_2$	موتور در حال گردش ساعتگرد
$p_3$	موتور در حال گردش پادساعتگرد

جدول ۲. توصیف گذرگاه‌های شکل ۵

گذرگاه	توصیف گذرگاه
$t_1$	شروع حرکت ساعتگرد
$t_2$	ایست کردن
$t_3$	عملگر به پایان حرکت ساعتگرد رسید
$t_4$	شروع حرکت پاد ساعتگرد
$t_5$	ایست کردن
$t_6$	عملگر به پایان حرکت پاد ساعتگرد رسید

جدول ۳. توصیف مکان‌های مدل منطقی راه‌انداز با استفاده از شبکه پتری

مکان	توصیف مکان
$p_4$	راه‌انداز بیکار است
$p_5$	راه‌انداز در حالت قرائت و دیکد کردن داده است
$p_6$	در حالت ارسال پیام نامعتبر بودن داده به کامپیوتر پرواز
$p_7$	در حالت ارسال تصدیق پیام دریافتی به کامپیوتر پرواز
$p_8$	در حالت تحلیل داده برای ساعتگرد یا پادساعتگرد بودن دوران موتور
$p_9$	در حالت اجرای دستور ساعتگرد و روشن بودن ساعت
$p_{10}$	حالت ارسال خاتمه حرکت ساعتگرد به کامپیوتر پرواز
$p_{11}$	ارسال خطا به کامپیوتر پرواز به دلیل رسیدن به سوئیچ حد مجاز سمت ساعتگرد
$p_{12}$	حالت ارسال خطا به کامپیوتر پرواز به دلیل تجاوز از حد زمانی
$p_{13}$	در حالت اجرای دستور پادساعتگرد و روشن بودن ساعت
$p_{14}$	حالت ارسال خاتمه حرکت پادساعتگرد به کامپیوتر
$p_{15}$	ارسال خطا به کامپیوتر پرواز به دلیل رسیدن به سوئیچ حد مجاز سمت پادساعتگرد
$p_{16}$	حالت ارسال خطا به کامپیوتر پرواز به دلیل تجاوز از حد زمانی
$p_{17}$	سنسور جریان در حال اندازه‌گیری جریان راه‌اندازی
$p_{18}$	ارسال خطا به کامپیوتر پرواز به دلیل تجاوز از حد جریان راه‌اندازی
$p_{19}$	سنسور در حال اندازه‌گیری جریان نامی
$p_{20}$	ارسال خطا به کامپیوتر پرواز به دلیل تجاوز از حد جریان نامی



شکل ۶. مدل منطقی راه‌انداز با استفاده از ابزار پتری

جدول ۴. توصیف گذرگاه‌های مدل راه‌انداز با استفاده از شبکه پتری

گذرگاه	توصیف گذرگاه
$t_7$	داده در بافر ورودی موجود است
$t_8$	داده نامعتبر است
$t_9$	اعلام نامعتبر بودن داده‌ها ارسال شد
$t_{10}$	داده معتبر است
$t_{11}$	اعلام معتبر بودن داده‌ها ارسال شد
$t_{12}$	شروع دوران موتور به‌صورت ساعتگرد
$t_{13}$	پایان اجرای فرمان ساعتگرد موتور
$t_{14}$	ارسال اعلام اجرای فرمان ساعتگرد موتور به کامپیوتر پرواز انجام شد
$t_{15}$	عملگر دچار خطای رسیدن به حد نهایی شد
$t_{16}$	ارسال خطای تجاوز از حد نهایی موقعیت ساعتگرد به کامپیوتر پرواز انجام شد
$t_{17}$	زمان‌سنج به حد نهایی خود رسید
$t_{18}$	ارسال خطای تجاوز از حد نهایی زمان‌سنج در حرکت ساعتگرد به کامپیوتر پرواز انجام شد
$t_{19}$	شروع دوران موتور به‌صورت پادساعتگرد
$t_{20}$	پایان اجرای فرمان پادساعتگرد موتور
$t_{21}$	ارسال اعلام اجرای فرمان پادساعتگرد موتور به کامپیوتر پرواز انجام شد
$t_{22}$	عملگر دچار خطای رسیدن به حد نهایی شد
$t_{23}$	ارسال خطای تجاوز از حد نهایی در حرکت پادساعتگرد به کامپیوتر پرواز انجام شد
$t_{24}$	زمان‌سنج به حد نهایی خود رسید
$t_{25}$	ارسال خطای تجاوز از حد نهایی زمان‌سنج در حرکت پادساعتگرد به کامپیوتر پرواز انجام شد
$t_{26}$	شروع اندازه‌گیری جریان راه‌اندازی در حرکت ساعتگرد
$t_{27}$	شروع اندازه‌گیری جریان راه‌اندازی در حرکت پادساعتگرد
$t_{28}$	ارسال خطای تجاوز از حد جریان راه‌اندازی به کامپیوتر پرواز
$t_{29}$	زمان راه‌اندازی موتور پایان یافت
$t_{30}$	شروع اندازه‌گیری جریان نامی در حرکت ساعتگرد
$t_{31}$	شروع اندازه‌گیری جریان نامی در حرکت پادساعتگرد
$t_{32}$	ارسال خطای تجاوز از حد جریان نامی به کامپیوتر پرواز

### ۳-۳. مدل پتری رفتار منطقی بافرها

بافرهای ورودی - خروجی جزئی از پردازشگر محسوب شده که به‌عنوان ضوابط باید مدل شوند. پردازشگر محاسبات اولیه را انجام داده و در بافر خروجی قرار می‌دهد. از طرفی داده‌های ارسال شده از کامپیوتر پرواز در بافر ورودی قرار گرفته و سپس برای تحلیل و استفاده به پردازشگر فرستاده می‌شود. در شکل ۷ مکان  $p_{24}$  نماینده حالت بافر خروجی و  $p_{25}$  نماینده حالت بافر ورودی است. توصیف مکان‌ها و گذرگاه‌های موجود در شکل ۷ به ترتیب در جدول‌های ۵ و ۶ آمده است.

### ۳-۴. مدلسازی منطقی کامپیوتر پرواز با شبکه پتری

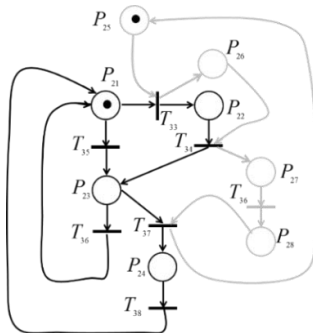
کامپیوتر پرواز در رابطه با عملگر بدین صورت عمل می‌کند که داده را از بافر ورودی دریافت کرده ( $p_{20}$ ) و شروع به پردازش داده دریافت شده می‌نماید. سرانجام دستورات خروجی لازم به بافر خروجی منتقل شده و از بافر خروجی به عملگر می‌رسد. مدلسازی

رفتار منطقی کامپیوتر پرواز با ابزار پتری در شکل ۸ آمده است. توصیف دقیق مکان‌ها و گذرگاه‌ها برای بخش مربوط به پردازشگر نیز به‌ترتیب در جدول‌های ۷ و ۸ آمده است.

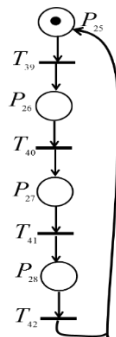
### ۴. مدلسازی نهایی

برای دستیابی به مدل نهایی دو گام باید برداشت: گام اول مدلسازی اجزا و ضوابط است که در بخش قبلی صورت پذیرفت. گام دوم ترکیب و سنکرون‌سازی مدل‌هاست. سنکرون‌سازی مدل‌های اصلی و ضوابط کار مهم و اثرگذاری است؛ زیرا در سنکرون‌سازی باید گذرگاه‌های مشترک بین ضوابط و اجزا مشخص و با هم ترکیب شوند. مدلسازی نهایی عملگر شامل دو بخش است: بخش اول شامل سنکرون‌سازی پردازشگر به‌عنوان بخش اصلی و بافرهای ورودی - خروجی به‌عنوان مدل ضوابط است که در شکل ۹ نتیجه آن مشاهده می‌شود. بخش دوم شامل سنکرون‌سازی موتور و راه‌انداز است که در شکل ۱۰ مدلسازی آن قابل مشاهده است.

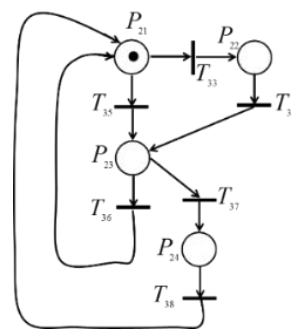




شکل ۹. نمایی از مدل سنکرون شده کامپیوتر پرواز و بافرها



شکل ۸. نمایی از مدل سازی رفتار منطقی کامپیوتر پرواز



شکل ۷. مدل سازی رفتار منطقی بافر ورودی خروجی با ابزار پتری

جدول ۵. توصیف مکان‌های مدل منطقی بافرها

توصیف مکان	مکان
در حال بررسی بافر ورودی	$p_{21}$
در حال دریافت داده در بافر ورودی	$p_{22}$
در حال بررسی بافر خروجی	$p_{23}$
در حال ارسال داده از بافر خروجی	$p_{24}$

جدول ۶. توصیف گذرگاه‌های مدل منطقی بافرها

توصیف گذرگاه	گذرگاه
بافر ورودی خالی نیست	$t_{33}$
داده در بافر ورودی دریافت شد	$t_{34}$
بافر ورودی خالی است	$t_{35}$
داده برای ارسال وجود ندارد	$t_{36}$
بافر خروجی اشغال است	$t_{37}$
داده در بافر خروجی ارسال شد	$t_{38}$

جدول ۷. توصیف مکان‌های مدل پتری کامپیوتر پرواز

توصیف مکان	مکان
کامپیوتر پرواز بیکار است	$p_{25}$
کامپیوتر پرواز در حال دریافت داده از بافر ورودی	$p_{26}$
در حال پردازش داده	$p_{27}$
کامپیوتر پرواز در حال ارسال داده به بافر خروجی	$p_{28}$

جدول ۸. توصیف گذرگاه‌های مدل پتری کامپیوتر پرواز

توصیف گذرگاه	گذرگاه
داده در بافر ورودی موجود است	$t_{39}$
داده از بافر خروجی به پردازشگر ارسال و پردازش شروع شد	$t_{40}$
پایان پردازش	$t_{41}$
دستور از پردازشگر به بافر خروجی ارسال شد	$t_{42}$



طرفی، می‌دانیم احتمال آتش شدن  $t_{10}$  پنج برابر  $t_8$  می‌باشد  
 $(\lambda_{10} = 5\lambda_8)$ . حال با توجه به زمان متوسط و رابطه ۵ داریم:

$$T_5 = \frac{1}{\lambda_8 + \lambda_{10}} = \frac{1}{\lambda_8 + 5\lambda_8} = \frac{1}{6\lambda_8} \Rightarrow \lambda_8 = \frac{1}{6T_5} \\ = \frac{1}{6T_5} \text{ \& } \lambda_{10} = \frac{5}{6T_5}$$

۳. پس از تعیین  $\lambda_i$  ها، توالی مختلف گذرگاه‌ها که به خطای مورد نظر در سیستم ختم می‌شود، محاسبه می‌شود. برای هر کدام از توالی‌ها،  $\omega$  و  $\tau$  های مرتبط با همان گذرگاه به دست آورده می‌شود.  $\omega$  نرمالیزه شده نرخ آتش گذرگاه‌های توالی مورد نظر و  $\tau$  حاصلضرب  $\omega$  در مدت زمان انجام آزمایش است. فرض بر این است که مدت زمان انجام آزمایش حدود ۱۰۰ ساعت باشد. جدول ۱۰ لیست خرابی‌ها به همراه توالی آنها را نشان می‌دهد.

۴. در این مرحله، با استفاده از رابطه ۶ احتمال تغییر از یک حالت به حالت دیگر محاسبه می‌شود.

$$Q_{M_i M_j}(\tau) = \int_0^\tau \lambda_L e^{-(\lambda_L + \lambda_k + \dots + \lambda_n)x} dx \quad (6)$$

۵. احتمال کل هر توالی خرابی از حاصلضرب هر کدام از تغییر حالت‌ها به دست آورده می‌شود:

$$f_{ij} = \prod_{i \in E} Q_{M_i M_j} \quad (7)$$

در مرحله آخر، با حاصل جمع تمام احتمالات به دست آمده (برای یک خطای مشخص)، احتمال خرابی کل به دست می‌آید.

$$F_{ij} = \sum_{n=1}^h f_{ij} \quad (8)$$

که در آن  $h$ ، تعداد کل توالی‌ها برای رسیدن به یک خطای مشخص است. در نهایت، در جدول ۱۱ احتمال کشف خرابی هر حالت آورده شده است.

همان‌گونه که در جدول ۱۱ مشاهده می‌شود، احتمال هر خرابی با در نظر گرفتن افزونگی تحلیلی به دست آورده شده است. برای بررسی عملکرد افزونگی تحلیلی، بخش زمان‌سنج از مدل اصلی حذف شده ( $p_{16}$  و  $p_{12}$ ) و احتمال خرابی مجدداً محاسبه می‌شود. در صورت نبود افزونگی تحلیلی برای زمان در نظر گرفته شده، احتمال کشف خرابی  $3/38 \times 10^{-4}$  کاهش پیدا می‌کند. همان‌طور که نشان داده شد، با استفاده از شبکه پتری می‌توان به راحتی رفتار مطلوب و منطقی مورد انتظار از هر زیرسیستم را

با توجه مدل‌سازی ارائه شده معادله حالت مدل سنکرون شده پردازشگر و بافرها در زیر ادامه آمده است.

$$W = W^+ - W^- \\ = \begin{bmatrix} -1 & 0 & -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$M(k) = M_0 + W \cdot U \\ = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot U \quad (4)$$

## ۵. روش محاسباتی احتمال خرابی

در این بخش کارایی و قابلیت اطمینان الگوریتم کنترل نظارتی طراحی شده بررسی می‌شود. از اینرو در این بخش احتمال خرابی برای مدل پتری عملگر بدون کنترل کننده و مدل پتری عملگر به همراه افزونگی محاسبه و مقایسه خواهند شد. هرچه میزان احتمال خرابی کاهش یابد، بدین معناست که قابلیت اطمینان کنترل کننده نظارتی طراحی شده بهتر است. قابلیت اطمینان احتمال انجام وظیفه سیستم بدون وقوع خرابی است، پس اگر احتمال خرابی سیستم بهبود یابد در نتیجه عملکرد سیستم بهبود یافته است. بنابراین این پارامتر یکی از معیارهای مهم برای بیان کارایی هر مجموعه می‌باشد. برای محاسبه احتمال خرابی، از الگوریتم زیر در ۷ مرحله استفاده می‌شود [۲۳].

۱. سیستم، مانند آنچه در بخش قبلی صورت گرفت، به همراه خطاهای آن مدل می‌شود

۲. نرخ آتش تمام گذرگاه‌ها،  $\lambda_i$  مشخص می‌شود. مقادیر نرخ آتش گذرگاه‌ها با استفاده از رابطه ۵ به دست می‌آید:

$$T_{avr} = \frac{1}{\sum \lambda_i} \quad (5)$$

جدول ۹ مقادیر نرخ آتش گذرگاه‌ها را برای یک عملگر فرضی نشان می‌دهد. برای محاسبه نرخ آتش گذرگاه‌ها فرض کنید که سیستم در حالت  $p_5$  (یعنی در وضعیت قرائت داده) به مدت  $T_5$  ثانیه (زمان متوسط حضور در این حالت) قرار دارد. از

استفاده از افزونگی تحلیلی، هزینه‌های طراحی و ریسک مواجهه با خطا را به شدت کاهش داده و در نتیجه احتمال خطای کل سیستم را بهبود می‌بخشد. هرچند استفاده از افزونگی تحلیلی هزینه سیستم تحمل‌پذیر خطا را کاهش می‌دهد، اما در بعضی مواقع نمی‌تواند حفظ سیستم در حالت امن را تضمین نماید. بنابراین در برخی شرایط در کنار افزونگی تحلیلی نیاز به افزونگی سخت‌افزاری برای سیستم وجود دارد.

بدون در نظر گرفتن دینامیک‌های پیچیده ریاضی، مدل و پیاده‌سازی نمود [۲۴]. مدل‌سازی با شبکه پتری ابزار بسیار قدرتمند گرافیکی و ریاضی در اختیار طراح قرار می‌دهد. خاصیت گرافیکی این ابزار، برای طراح اصلی امکان مشاهده روند سیستم و خطاهای احتمالی را به صورت دیاگرام ارائه می‌دهد. سپس به راحتی با ابزار ریاضی که در اختیار می‌گذارد، امکان مدل‌سازی روش حل یا سناریوهای ممکن بعد از رخ دادن خطا را ایجاد می‌نماید [۲۵].

جدول ۹. نرخ آتش گذرگاه‌ها

نرخ آتش		نام گذرگاه
۰/۸	$\lambda_7$	$t_7$
۰/۰۵	$\lambda_8$	$t_8$
۰/۰۰۶	$\lambda_9$	$t_9$
۰/۲۵	$\lambda_{10}$	$t_{10}$
۰/۲	$\lambda_{11}$	$t_{11}$
۰/۴	$\lambda_{12}$	$t_{12}$
۰/۱	$\lambda_{13}$	$t_{13}$
۰/۰۹	$\lambda_{14}$	$t_{14}$
۰/۰۰۳	$\lambda_{15}$	$t_{15}$
۰/۰۰۱	$\lambda_{16}$	$t_{16}$
۰/۰۰۵	$\lambda_{17}$	$t_{17}$
۰/۰۰۳	$\lambda_{18}$	$t_{18}$
۰/۴	$\lambda_{19}$	$t_{19}$
۰/۱	$\lambda_{20}$	$t_{20}$
۰/۰۹	$\lambda_{21}$	$t_{21}$
۰/۰۳	$\lambda_{22}$	$t_{22}$
۰/۰۰۱	$\lambda_{23}$	$t_{23}$
۰/۰۰۵	$\lambda_{24}$	$t_{24}$
۰/۰۰۳	$\lambda_{25}$	$t_{25}$
۰/۳	$\lambda_{26}$	$t_{26}$
۰/۰۳	$\lambda_{27}$	$t_{27}$
۰/۰۰۰۱	$\lambda_{28}$	$t_{28}$
۰/۰۱	$\lambda_{29}$	$t_{29}$
۰/۳	$\lambda_{30}$	$t_{30}$
۰/۰۳	$\lambda_{31}$	$t_{31}$
۰/۰۰۰۱	$\lambda_{32}$	$t_{32}$

جدول ۱۰. لیست خرابی‌ها و توالی آن‌ها

عنوان خرابی		توالی خرابی		نرخ آتش نرمال شده هر توالی خطا		حاصلضرب نرخ آتش توالی خطا در زمان آزمایش	
داده نامعتبر		$t_7, t_8$	$\omega_7$	۰/۱۱۱	$\tau_7$	۱۱/۱	
			$\omega_8$	۰/۸۸۸	$\tau_8$	۸۸/۸	
انبعاث حرکت در حالت ساعتگرد		$t_7, t_{10}, t_{11}, t_{12}, t_{15}$	$\omega_7$	۰/۰۴۶۹	$\tau_7$	۴/۶۹	
			$\omega_{10}$	۰/۳۷۵۵	$\tau_{10}$	۳۷/۵۵	
			$\omega_{11}$	۰/۳۷۵۵	$\tau_{11}$	۳۷/۵۵	
			$\omega_{12}$	۰/۱۸۷۸	$\tau_{12}$	۱۸/۷۸	
			$\omega_{15}$	۰/۰۱۴۱	$\tau_{15}$	۱/۴۱	

۴/۷۵	$\tau_7$	۰/۰۴۷۵	$\omega_7$	$t_7, t_{10}, t_{11}, t_{12}, t_{17}$	اشباع زمانی در حالت حرکت ساعتگرد
۳۸	$\tau_{10}$	۰/۳۸	$\omega_{10}$		
۳۸	$\tau_{11}$	۰/۳۸	$\omega_{11}$		
۱۹	$\tau_{12}$	۰/۱۹	$\omega_{12}$		
۰/۲۳۷	$\tau_{17}$	۰/۰۰۲۳	$\omega_{17}$		
۴/۹۹	$\tau_7$	۰/۰۴۹۹	$\omega_7$	$t_7, t_{10}, t_{11}, t_{26}, t_{28}$	جریان راهاندازی بیش از حد در ساعتگرد
۳۹/۹	$\tau_{10}$	۰/۳۹۹	$\omega_{10}$		
۳۹/۹	$\tau_{11}$	۰/۳۹۹	$\omega_{11}$		
۱۴/۹۹۹	$\tau_{26}$	۰/۱۴۹۹	$\omega_{26}$		
$۴/۹۹ \times 10^{-3}$	$\tau_{28}$	$۴/۹۹ \times 10^{-3}$	$\omega_{28}$		
۴/۱۶۶	$\tau_7$	۰/۰۴۱۶	$\omega_7$	$t_7, t_{10}, t_{11}, t_{12}, t_{27}, t_{28}$	جریان نامی بیش از حد در ساعتگرد
۳۳/۳	$\tau_{10}$	۰/۳۳۳	$\omega_{10}$		
۳۳/۳	$\tau_{11}$	۰/۳۳۳	$\omega_{11}$		
۱۶/۶۶	$\tau_{12}$	۰/۱۶۶۶	$\omega_{12}$		
۱۲/۵	$\tau_{27}$	۰/۱۲۵	$\omega_{27}$		
$۴/۱۶ \times 10^{-3}$	$\tau_{28}$	$۴/۱۶ \times 10^{-3}$	$\omega_{28}$		
۴۶۹	$\tau_7$	۰/۰۴۶۹	$\omega_7$	$t_7, t_{10}, t_{11}, t_{19}, t_{22}$	اشباع حرکت در حالت پادساعتگرد
۳۷/۵۶	$\tau_{10}$	۰/۳۷۵۶	$\omega_{10}$		
۳۷/۵۶	$\tau_{11}$	۰/۳۷۵۶	$\omega_{11}$		
۱۸/۷۸	$\tau_{19}$	۰/۱۸۷۸	$\omega_{19}$		
۱/۴	$\tau_{22}$	۰/۰۱۴	$\omega_{22}$		
۴/۷۴	$\tau_7$	۰/۰۴۷۴	$\omega_7$	$t_7, t_{10}, t_{11}, t_{19}, t_{24}, t_{25}$	اشباع زمانی در حالت حرکت پادساعتگرد
۳۷/۹	$\tau_{10}$	۰/۳۷۹	$\omega_{10}$		
۳۷/۹	$\tau_{11}$	۰/۳۷۹	$\omega_{11}$		
۱۸/۹	$\tau_{19}$	۰/۱۸۹	$\omega_{19}$		
۰/۲۳۷	$\tau_{24}$	۰/۰۰۲۳۷	$\omega_{24}$		
۰/۱۴۳	$\tau_{25}$	۰/۰۰۱۴۳	$\omega_{25}$		
۴/۹۹	$\tau_7$	۰/۰۴۹۹	$\omega_7$	$t_7, t_{10}, t_{11}, t_{27}, t_{28}$	جریان راهاندازی بیش از حد در پادساعتگرد
۳۹/۹۹	$\tau_{10}$	۰/۳۹۹۹	$\omega_{10}$		
۳۹/۹۹	$\tau_{11}$	۰/۳۹۹۹	$\omega_{11}$		
۱۴/۹۹	$\tau_{27}$	۰/۱۴۹۹	$\omega_{27}$		
$۴/۹۹ \times 10^{-3}$	$\tau_{28}$	$۴/۹۹ \times 10^{-3}$	$\omega_{28}$		
۴/۱۶۶	$\tau_7$	۰/۰۴۱۶	$\omega_7$	$t_7, t_{10}, t_{11}, t_{19}, t_{31}, t_{32}$	جریان نامی بیش از حد در پادساعتگرد
۳۳/۳۳	$\tau_{10}$	۰/۳۳۳۳	$\omega_{10}$		
۳۳/۳۳	$\tau_{11}$	۰/۳۳۳۳	$\omega_{11}$		
۱۶/۶۶	$\tau_{19}$	۰/۱۶۶۶	$\omega_{19}$		
۱۲/۴۹۹	$\tau_{31}$	۰/۱۲۴۹	$\omega_{31}$		
$۴/۱۶ \times 10^{-3}$	$\tau_{32}$	$۴/۱۶ \times 10^{-3}$	$\omega_{32}$		

جدول ۱۱. احتمال کشف خرابی هر بخش

ردیف	خرابی	احتمال خرابی
۱	داده نامعتبر	۰/۰۷۳۴
۲	اشباع حرکت	۰/۰۱۰۶
۳	اشباع زمانی	$۳/۳۸ \times 10^{-4}$
۴	جریان راهاندازی بیش از حد	$۱/۷۳ \times 10^{-7}$
۵	جریان نامی بیش از حد	$۱/۷ \times 10^{-8}$

ارتباط بین طراح و مدل را ساده‌تر می‌نماید. با استفاده از این ابزار مدلسازی، رفتار و خطاهای عملگر بدون در نظر گرفتن دینامیک‌های پیچیده ریاضی مدل شد. سپس با افزودن تحلیلی، تدابیر و سناریوهای لازم، تحت عنوان کنترل نظارتی برای مقابله با خطاهای احتمالی یک سناریوی فرضی طراحی شد. با توجه به نتایج به‌دست آمده، تکنیک نرم‌افزاری افزودن تحلیلی باعث کاهش احتمال خطای سیستم گردید.

## ۶. نتیجه‌گیری

در این مقاله، روشی نوین برای مدلسازی رفتارهای منطقی عملگر برقی - مکانیکی، مدلسازی خطا و طراحی کنترل‌کننده نظارتی جهت کاهش احتمال خطا در پرنده‌های بدون سرنشین ارائه شده است. با استفاده از ابزار گرافیکی قدرتمند شبکه پتری به‌راحتی مدلی از رفتار منطقی عملگر ارائه شد. خاصیت گرافیکی شبکه‌های پتری و معادلات ریاضی که در اختیار قرار می‌دهد،

## ۷. مأخذ

- [1] Y. Zhou, C. Lin, Y. Liu, H. Xu, Analytical Study on the Reliability of Redundancy Architecture for Flight Control Computer Based on Homogeneous Markov Process, *IEEE Access*, Vol. 6, pp. 18290-18298, 2018.
- [2] A. Ahangarani Farahani and A. Dideban, A Novel Method to Model Electromechanical Actuator in UAV, *The 13<sup>th</sup> Conference of Iranian Aerospace Society*, Tehran, Iran, 2014. (in Persian)
- [3] B. Hruz and M. C. Zhou, Modeling, *Control of Discrete-event Dynamic Systems*, Springer, 2007.
- [4] O. González, M. Cerrad, Diagnosis of Controlled Discrete Event Systems: An Approach Based on Chronicles and Modular Analysis by Using Automata Models, *Revista Iberoamericana de Automática e Informática industria*, Vol. 11, No. 2, pp. 191-201, 2014.
- [5] R. David, H. Alla, Discrete, Continuous, and Hybrid Petri Nets, *Printed in Germany*, Springer, 2005.
- [6] L. Ghomri, H. Alla, Continuous Petri Nets and Hybrid Automata for the Analysis of Manufacturing Systems, *International Federation of Automatic Control*, Vol. 48, No. 3, pp. 1024-1029, 2015.
- [7] S. Thavamanz, Fault Tolerant Control of a Ship Propulsion, Ph.D Thesis, University State University of New York, USA, 2006.
- [8] F. Bateman, H. Noura, M. Ouladsine, Active Fault Diagnosis and Major Actuator Failure Accommodation: Application to a UAV, *InTech Advances in Flight Control Systems*, 2011.
- [9] F. Bateman, H. Noura, M. Ouladsine, Actuators Fault Diagnosis and Tolerant Control for an Unmanned Aerial Vehicle, *IEEE International Conference on Control Applications Part of IEEE Multi-conference on Systems and Control*, Singapore, 2007.
- [10] D. Lefebvre, On-Line Fault Diagnosis With Partially Observed Petri Nets, *IEEE Transactions on Automatic Control*, Vol. 59, No. 7, pp. 1919-1924, 2014.
- [11] J. Liu, K. Chen, Z. Wang, Fault Analysis for Flight Control System Using Weighted Fuzzy Petri Nets, *Journal of Convergence Information Technology*, Vol. 6, No. 3, 2011.
- [12] C. Wu, J. Qi, D. Song, X. Qi, J. Han, Simultaneous State and Parameter Estimation Based Actuator Fault Detection and Diagnosis for an Unmanned Helicopter, *International Journal of Applied Mathematics and Computer Science*, Vol. 25, No. 1, pp. 175-187, 2015.
- [13] M. A. Trigos, A. Barrientos, J. del Cerro, Unmanned Helicopter Faults Diagnosis based on Petri Nets, *Revista de Investigaciones*, Vol. 8, No. 2, pp. 91-103, 2016.
- [14] M. Blanke, C. W. Frei, F. Kraus, R. J. Patton M. Staroswiecki, *What is Fault-Tolerant Control*, *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, Budapest, Hungary, 2000.
- [15] A. Ahangarani Farahani, A. Dideban, Continuous-Time Delay-Petri Nets as A New Tool to Design State Space Controller, *Information Technology and Control*, Vol. 45, No. 4, pp. 401-411, 2016.

- [16] A. Ahanarani Farahani, A. Dideban, E. Najafgholi, Modeling Continuous Systems by Petri Nets Using Speed Control Arcs. the 4<sup>th</sup> International Conference on Control, Instrumentation and Automation. Iran, 2016.
- [17] M. Zareiee, A. Dideban, A New Idea for Having a Simplified Controller in Discrete Event Systems, The First Iran International Conference on Industrial Automation, Iran, 2009. (in Persian)
- [18] B. Gudino-Mendoza, E. López-Mellado, H. Alla, A Linear Characterization of the Switching Dynamic Behavior of Timed Continuous Petri Nets with Structural Conflicts, *Nonlinear Analysis: Hybrid Systems*, Vol. 19, pp.38–59, 2016.
- [19] A. Aybar, A. Iftar, Supervisory Controller Design to Enforce Basic Properties in Timed-Place Petri Nets, Proc.of the 6<sup>th</sup> IFAC Conference on Management and Control of Production and Logistic .Fortaleza, Brazil, 2013.
- [20] J. J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*, Marcel Dekker, New York, USA, 1998.
- [21] P. Jalote, *Fault Tolearnce in Distributed Systems*, Printice Hall, 1998.
- [22] M. Ferval, A. Lecanu, A. Maussion, *Flight Control Sstem for an Aircraft*, USA, 2011.
- [23] M. Hejrati, Analyzing the Reliability of the Automation System on Vessels Using Petri Nets and Providing a Way to Improve It, Master Thesis, Semnan University, Semnan, Iran, 2016. (in Persian)
- [24] L. Sun, W. Jiang, S. Yang M. Yu, Fault Diagnosis for Electromechanical System via Extended Analytical Redundancy Relations, *IEEE Transaction on Industrial Information*, 2018.
- [25] J. Ruiz-Leon, C. R. Vazquez and A. Ramirez-Trevino J. A. Fraustro, Structural Fault Diagnosis in Timed Continuous Petri Nets, in Proceedings of the 13<sup>th</sup> International Workshop on Discrete Event System, Xi'an, China, pp. 159-164, 2016.

پی‌نوشت

- 
1. flight computer
  2. discrete event
  3. automata
  4. Petri nets
  5. place
  6. transition
  7. marking
  8. arc
  9. incidence matrix
  10. supervisory control