

# بهینه‌سازی سیستم استنتاج فازی در حال تکامل، برای شناسایی و کنترل بهینه مولتی روتورها

محمد حسین علوی دوست<sup>۱</sup>، حسین صفری<sup>۲\*</sup>، فرزاد بهرامی<sup>۳</sup>

- ۱- دانشجوی دکتری مدیریت صنعتی، دانشکده مدیریت دانشگاه تهران، تهران، ایران
- ۲- استاد گروه مدیریت صنعتی، دانشکده مدیریت دانشگاه تهران، تهران، ایران
- ۳- استادیار گروه مدیریت صنعتی، دانشکده علوم اداری و اقتصاد، اراک، ایران  
(دریافت مقاله: ۱۴۰۴/۰۲/۰۷ تاریخ پذیرش: ۱۴۰۴/۰۷/۱۱)

## چکیده

در سال‌های اخیر، وسیله نقلیه هوایی بدون سرنشین (UAV) در حوزه‌های نظامی و غیرنظامی گسترش چشمگیری داشته است. در این میان، پهپادهای پرده‌گردان (RUAV) و به طور خاص مولتی روتورها به دلیل توانایی برخاست و فرود عمودی و انجام مانورهای سریع، مورد توجه ویژه محققان در طراحی سیستم‌های کنترل خودمختار قرار گرفته‌اند. اغلب روش‌های سنتی کنترل مولتی روتورها مبتنی بر تکنیک اصل اولیه (FPT) بوده و در مواجهه با عدم قطعیت‌ها ناکارآمد هستند. اخیراً سیستم‌های فازی هوشمند جایگزین این روش‌ها شده‌اند که با وجود بهبود عملکرد، به دلیل ساختار ثابت، قابلیت انطباق با محیط‌های پویا را ندارند. سیستم‌های فازی در حال تکامل (EFS) با به‌روزرسانی ساختار و پارامترها، راهکاری مؤثر برای مدیریت تغییرات ناگهانی در پروازهای بلادرنگ مولتی روتورها ارائه می‌دهند. اگرچه نسخه‌های پیشرفته‌ای از EFS توسعه یافته‌اند، اما اغلب تمرکز آن‌ها بر شناسایی سیستم بوده و به بهینه‌سازی پارامترها توجه کافی نشده است. این پژوهش با معرفی EFS جدیدی به نام GODFIS (سیستم استنتاج فازی پویا مبتنی بر بهینه‌سازی سراسری) که از مفاهیم مرکز ثقل بازگشتی و بهینه‌سازی سراسری پارامترهای تالی بهره می‌برد، به رفع این شکاف می‌پردازد. همچنین، معیار جدید با عنوان «اصل حذف نویز» (NEP) به منظور حفظ کیفیت پاسخ‌ها و کاهش اثرات نویز ارائه شده است. آزمایشات نشان می‌دهد که روش GODFIS در حل مسائل معیار از دقت بالاتری نسبت به روش‌های موجود برخوردار است. با استفاده از مجموعه داده‌های متنوع، اثبات شده که این الگوریتم علاوه بر مدل‌سازی و کنترل مولتی روتورها، از قابلیت تعمیم‌پذیری و کاربرد گسترده در حوزه‌های مختلف کنترل و پیش‌بینی برخوردار است.

واژه‌های کلیدی: سیستم‌های فازی در حال تکامل، بهینگی، کنترل، مولتی روتورها

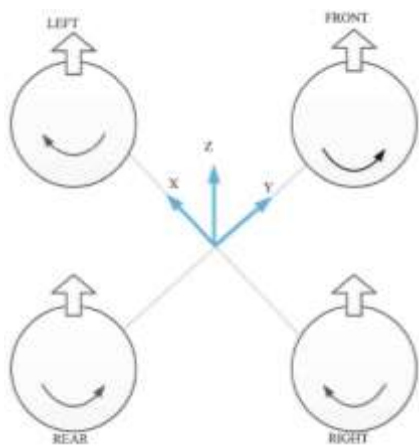
## Optimization of Evolving Fuzzy Inference Systems for Optimal Identification and Control of Multi-rotors

M. H. Alavi doost, H. Saffari, F. Bahrami

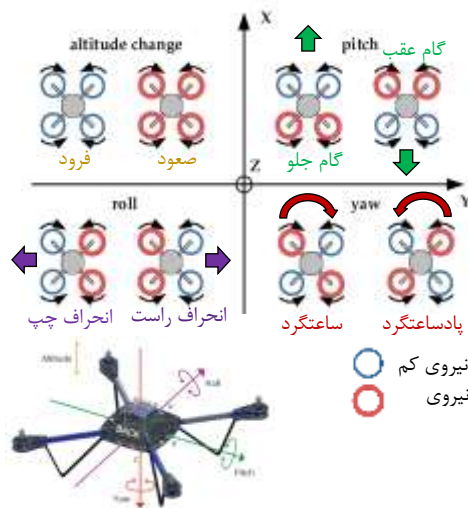
### Abstract

In recent years, unmanned aerial vehicles (UAVs) have seen significant growth in both military and civilian applications. Among them, rotary-wing UAVs (RUAVs), particularly multi-rotors, have gained attention from researchers for designing autonomous control systems due to their vertical takeoff and landing capabilities and rapid maneuvers. Most traditional multirotor control methods are based on First Principles Techniques (FPT), which face challenges in managing uncertainties. Recently, intelligent fuzzy systems have replaced these methods; however, despite improved performance, their fixed structure limits adaptability to dynamic environments. Evolving fuzzy systems (EFS), by updating both structure and parameters, provide an effective solution for managing sudden changes in real-time multirotor flights. Although advanced EFS versions have been developed, they often focus on system identification, neglecting sufficient parameter optimization. This research introduces a new EFS called GODFIS (Globally Optimized Dynamic Fuzzy Inference System), which utilizes recursive center-of-gravity and global optimization of consequent parameters to address this gap. Additionally, a new criterion, the "Noise Elimination Principle" (NEP), is proposed to maintain response quality and reduce noise effects. Experiments demonstrate that GODFIS achieves higher accuracy than existing methods in benchmark problems. Using diverse datasets, it has been proven that this algorithm not only excels in multi-rotor modeling and control but also offers high generalizability and broad applicability in various control and prediction domains.

**Key words:** Evolving Fuzzy Systems, Optimality, Control, Multi-rotors



شکل ۱. یک کوادکوپتر ساده با پیکربندی ضربدری



شکل ۲. نحوه ایجاد ۴ حرکت اصلی در کوادکوپترها

با توجه به کاربردهای گسترده مولتی روتور در حوزه‌های نظامی و غیرنظامی، تمایل تحقیقاتی برای افزایش هوشمندی این سیستم‌ها در حال رشد است.

اکثر روش‌های سنتی کنترل مولتی روتورها مبتنی بر تکنیک اصل اولیه (FPT)<sup>۸</sup> می‌باشند. FPT به روش‌های کنترلی مبتنی بر مدل‌های ریاضی دقیق سیستم، که از قوانین فیزیکی اساسی (مانند معادلات دینامیکی نیوتنی) مشتق می‌شوند، اشاره دارند. این روش‌ها شامل کنترلرهای کلاسیک مانند کنترلرهای تناسبی-انترگرالی-مشتقی (PID)، کنترلرهای بهینه خطی-مربعی (LQR)، و مدل‌های پیش‌بینی‌کننده (MPC) هستند که برای طراحی سیستم‌های کنترلی مولتی

## مقدمه

پیشرفت‌های فناوری موجب بهبود کارایی وسایل نقلیه هوایی بدون سرنشین<sup>۱</sup> (پهپادها) شده است [۱، ۲]. این وسایل، با سطوح مختلف خودمختاری، به دلیل مزایایی مانند انجام مأموریت‌های پرخطر، وزن کم، مصرف سوخت بهینه و تعمیرات آسان، کاربرد گسترده‌ای در حوزه‌های نظامی و غیرنظامی یافته‌اند [۳، ۴].

پهپادها به سه دسته اصلی تقسیم می‌شوند: پهپادهای بال ثابت، پره‌گردان<sup>۲</sup> (RUAV) و بال متحرک<sup>۳</sup>. در این میان، مولتی روتورها<sup>۴</sup> به عنوان زیرمجموعه‌ای از RUAVها، به دلیل قابلیت‌های منحصر به فردی مانند توانایی برخاست و فرود عمودی و انجام مانورهای سریع، توجه ویژه پژوهشگران را به خود جلب کرده‌اند. در میان انواع مولتی روتورها، کوادکوپترها به دلیل مزایایی مانند وزن سبک و هزینه کمتر، از محبوبیت و کاربرد گسترده‌تری برخوردارند. کوادکوپترها در دو پیکربندی ضربدری (×) و بعلاوه‌ای (+) ساخته می‌شوند (پیکربندی ضربدری در شکل ۱ نشان داده شده است) و حرکت آن‌ها از طریق تغییر سرعت چهار روتور کنترل می‌شود (شکل ۲ نحوه ایجاد ۴ حرکت اصلی را نمایش می‌دهد):

- حرکت عمودی (بالا/پایین Z): افزایش/کاهش سرعت یکسان همه روتورها.
- انحراف<sup>۵</sup> (چپ/راست θ): تغییر سرعت روتورهای چپ راست.
- گام<sup>۶</sup> (جلو/عقب φ): تنظیم سرعت روتورهای جلو یا عقب.
- چرخش<sup>۷</sup> (ψ): افزایش سرعت یک جفت روتور مورب و کاهش جفت مقابل.

<sup>1</sup> Unmanned Aerial Vehicle

<sup>2</sup> rotary wing

<sup>3</sup> flapping wing

<sup>4</sup> Multi-Rotor

<sup>5</sup> roll

<sup>6</sup> pitch

<sup>7</sup> yaw

<sup>8</sup> first-principle techniques

روتورها استفاده می‌شوند. در زمینه مولتی روتورها، FPTها معمولاً از معادلات ۶ درجه آزادی (۶-DoF) برای مدل‌سازی حرکات انتقالی (X, Y, Z) و چرخشی ( $\varphi, \theta, \psi$ ) استفاده می‌کنند. با این حال، این روش‌ها در مواجهه با عدم قطعیت‌های محیطی (مانند نویز حسگرها یا اختلالات باد) و دینامیک‌های غیرخطی پیچیده مولتی روتورها با چالش‌هایی مواجه هستند. به عنوان مثال، کنترلرهای PID به تنظیم دقیق پارامترها وابسته‌اند و در شرایط پویا ممکن است عملکرد ضعیفی داشته باشند [۵]. به عبارت دیگر، اگرچه تکنیک‌های کنترل کلاسیک (FPT) به دلیل سادگی ساختار با موفقیت در کنترل انواع مولتی روتورها به کار گرفته شده‌اند، اما ناتوانی این روش‌ها در مدل‌سازی سیستم‌های دینامیکی ذاتاً غیرخطی، موجب توجه پژوهشگران به روش‌های کنترل غیرخطی شده است. نقطه ضعف اصلی کنترلرهای خطی و غیرخطی، وابستگی آن‌ها به مدل دقیق ریاضی سیستم و چالش‌های موجود در مدل‌سازی عدم قطعیت‌ها (مانند نویز سنسورها و اختلالات محیطی) است که محققان را به سمت روش‌های مبتنی بر دانش بدون نیاز به مدل سوق داده است.

در میان روش‌های متعدد بدون مدل، سیستم‌های منطقی فازی (FLS)<sup>۹</sup>، شبکه‌های عصبی مصنوعی (ANN)<sup>۱۰</sup> و روش‌های ترکیبی عصبی-فازی (NF)<sup>۱۱</sup> به عنوان پرکاربردترین روش‌ها در حوزه کنترل شناخته می‌شوند [۴]. چرا که توانایی مؤثری در مواجهه با سیستم‌های پیچیده غیرخطی و آشفته دارند. ANNها به طور موفقیت‌آمیزی در مدل‌سازی جعبه سیاه<sup>۱۲</sup> برای بسیاری از کاربردهای پیش‌بینی مورد استفاده قرار گرفته‌اند، زیرا دارای ساختارهای مدل‌سازی انعطاف‌پذیری هستند که با سیستم‌های متنوعی سازگار می‌شوند. با این حال، انتقاد اصلی به ANNها تولید مدل‌های غیرشفاف است که تفسیر دانش نهفته در آن‌ها را دشوار می‌سازد [۶].

برای مواجهه با این مشکل، برخی از محققان بر روی سیستم مدل‌سازی فازی مبتنی بر داده متمرکز شده‌اند که می‌توان آن‌ها را به صورت سیستم‌های عصبی-فازی نیز نمایش داد. (ترکیبی از سیستم‌های فازی با الگوریتم‌های یادگیری شبکه عصبی که برای جبران محدودیت‌های ذاتی هر دو روش

طراحی شده‌اند). این سیستم‌ها مدل‌های انعطاف‌پذیری با قوانین فازی IF-THEN تفسیرپذیر تولید می‌کنند که قادر به یادگیری از داده‌های تجربی بدون نیاز به دخالت انسان هستند. همچنین، سیستم‌های فازی توانایی بالاتری در مدیریت الگوهای داده نادقیق<sup>۱۳</sup> که از اطلاعات تاریخی متمایز هستند، از خود نشان می‌دهند. فرایند استنتاج فازی که شبیه به تصمیم‌گیری انسانی عمل می‌کند، امکان ارائه توضیحات شفاف برای هر تصمیم اتخاذ شده توسط مدل‌های مبتنی بر قوانین فازی را فراهم می‌سازد؛ به گونه‌ای که تصمیمات توسط متخصصان حوزه قابل بررسی و تأیید بوده و تعامل انسان-ماشین را تسهیل می‌نماید [۷].

انواع مختلفی از سیستم‌های عصبی-فازی در سال‌های اخیر توسعه یافته‌اند [۸-۱۳]. به طور خلاصه، در این میان تکنیک‌های مختلف مبتنی بر دانش، سیستم‌های استنتاج فازی (و عصبی-فازی) به عنوان یکی از برجسته‌ترین روش‌های مورد استفاده در کنترل مولتی روتورها مطرح هستند. این برتری نه تنها به دلیل انعطاف‌پذیری ذاتی این سیستم‌ها در حل مسائل متنوع، بلکه به خاطر ارائه راهکارهای شفاف و تفسیرپذیری است که به راحتی توسط اپراتورها قابل راستی‌آزمایی می‌باشند. همین ویژگی‌ها موجب جلب توجه گسترده محققان در کاربردهای مختلف کنترل شده است [۱۴-۱۶].

برای ارتقای عملکرد و مقابله با عدم قطعیت‌ها، توسعه از کنترلرهای فازی نوع ۱ به نوع ۲ صورت گرفته است [۱۷-۲۰]. علاوه بر این، امکان پیاده‌سازی این سیستم‌ها با تقلید از کنترلرهای FPT از طریق آموزش آفلاین وجود دارد. کنترل‌کننده‌های فازی با تکنیک‌های مرسوم مانند فازی-PID<sup>۱۴</sup>، فازی-PI، فازی-PD [۲۱-۲۶]، کنترل لغزشی فازی<sup>۱۵</sup> [۲۷]، کنترل پس‌گرد<sup>۱۶</sup> فازی [۲۸، ۲۹] ترکیب می‌شوند تا مدل‌سازی دقیق‌تری از کنترل مولتی روتور ارائه داده و عملکرد بهینه‌تری داشته باشند.

با این وجود، کنترل‌کننده‌های آموزش‌دیده مذکور دارای محدودیت‌های قابل توجهی هستند:

(۱) معماری ثابت: این سیستم‌ها دارای ساختارهای از پیش تعیین‌شده با تعداد ثابت نورون‌ها، توابع عضویت، قوانین و

<sup>13</sup> Imprecisely described data

<sup>14</sup> Proportional Integral Derivative

<sup>15</sup> fuzzy-sliding mode

<sup>16</sup> back-stepping

<sup>9</sup> Fuzzy Logic systems

<sup>10</sup> Artificial Neural Networks

<sup>11</sup> Neuro-Fuzzy

<sup>12</sup> Black-box

افزایشی آنلاین، ۳) ساختار باز و خودسازمانده، ۴) قابلیت ذخیره سازی اطلاعات، ۵) تعامل فعال، ۶) اکتساب دانش و خودبهبودی و ۷) یادگیری مکانی-زمانی [۳۶].

کنترلرهای انعطاف پذیر نه تنها امکان تنظیم پارامترها را فراهم می کنند، بلکه می توانند ساختار خود را با افزودن یا حذف لایه ها و قوانین به صورت خودتطبیقی تکامل بخشند [۳۸، ۳۹]. همچنین، ویژگی خودانطباقی به آن ها اجازه می دهد بدون نیاز به تنظیم مجدد یا آموزش از ابتدا، با تغییرات پویای سیستم سازگار شوند. به عبارت دیگر، این سیستم ها از سناریوی یادگیری تک گذری<sup>۲۴</sup> استفاده می کنند که برای نیازهای آنلاین در پلتفرم های با منابع محاسباتی محدود (مانند مولتی روتورها) مناسب است [۴۰-۴۲].

پژوهش ها در زمینه کنترلرهای فازی تکاملی از آغاز قرن ۲۱ با معرفی کنترلر خودسازمانده فازی-عصبی در مقاله [۴۳] آغاز شد و در مطالعاتی مانند [۳۸، ۴۸-۴۴] توسعه یافت. این روش ها به دلیل سرعت یادگیری بالا، انعطاف پذیری، قابلیت مدل سازی آنلاین و همچنین نیاز به منابع محاسباتی محدود، موفقیت چشمگیری در مسائل کنترل داشته اند.

با وجود موفقیت های چشمگیر EFS ها، یکی از چالش های اصلی این سیستم ها بهینه نبودن بخش های مقدم و تالی<sup>۲۵</sup> است که از جریان داده ها یاد می گیرند [۴۹]. این مشکل ناشی از استفاده از روش یادگیری تک گذری بدون مکانیزم بهینه سازی است که امکان جستجوی مکرر برای یافتن راه حل بهینه را محدود می کند.

با توجه به وجود این کاستی در روش های موجود، این مقاله نیز در راستای رفع آن، یک الگوریتم جدید را معرفی می کند. در این راستا، با الهام از روش های کلاسیک، یک چارچوب بهینه سازی آنلاین برای شناسایی سیستم های فازی در حال تکامل تاکاگی-سوگنو ارائه می شود که تحت عنوان GODFIS<sup>۲۶</sup> (سیستم استنتاج فازی پویا مبتنی بر بهینه سازی سراسری) شناخته می شود.

الگوریتم GODFIS با هدف پیش بینی سری های زمانی و کنترل سیستم های غیرخطی، با دقت بالا و سرعت آموزش مناسب طراحی شده و شامل استراتژی های مختلفی برای تعیین تعداد قوانین فازی (شامل افزودن، حذف و هرس

لایه ها هستند. این پیکربندی ایستا، توانایی تطابق با شرایط غیرثابت در طول زمان<sup>۱۷</sup> را از آن ها سلب می کند.

(۲) وابستگی به داده های ورودی: عملکرد این کنترل کننده ها به شدت متکی بر دانش کارشناسی یا داده های آموزشی است. هرگونه نقص یا نادرستی در این داده ها مستقیماً به کاهش کارایی سیستم منجر می شود.

(۳) محدود بودن به مسائل آفلاین: این روش ها فرض می کنند تمام داده های مورد نیاز در ابتدای فرایند آموزش در دسترس هستند، بنابراین صرفاً برای مسائل آفلاین مناسب بوده و قابلیت استفاده در محیط های پویا را ندارند.

(۴) نیاز به پردازش دسته ای<sup>۱۸</sup>: برای به روزرسانی مدل، این سیستم ها نیازمند بازبینی کامل مجموعه داده ها هستند که مستلزم یادگیری مجدد و صرف زمان قابل توجه است. این ویژگی باعث کاهش قابلیت مقیاس پذیری برای استقرار بلادرنگ<sup>۱۹</sup> (آنلاین) می شود.

(۵) عدم انعطاف پذیری ساختاری: منطق فازی سنتی، تکنیک های کنترل و روش های کنترل ترکیبی مورد استفاده در مدل سازی مولتی روتورها، فاقد توانایی تکامل ساختاری برای انطباق با تغییرات ناگهانی و اساسی در محیط عملیاتی هستند.

همچنین، مطالعات متعدد [۳۰-۳۴] نشان داده اند که رایج ترین معماری ها و الگوریتم های شبکه عصبی شامل پرسپترون های چندلایه<sup>۲۰</sup> (MLPs)، شبکه های پایه شعاعی<sup>۲۱</sup> (RBF) و نگاشت های خودسازمانده<sup>۲۲</sup> (SOM) برای یادگیری تطبیقی و آنلاین کارایی مناسبی ندارند.

برای رفع این محدودیت ها، می توان از کنترلرهای استنتاج فازی با ساختار انعطاف پذیر تحت عنوان سیستم های فازی تکاملی (EFSs)<sup>۲۳</sup> بهره برد [۳۵]. این سیستم ها باید قادر باشند در حین فعالیت، توسعه یابند، دانش خود را به روز کنند و مدل را از طریق تعامل با محیط اصلاح نمایند [۳۶، ۳۷].

سیستم های هوشمند تطبیقی آنلاین باید هفت ویژگی اساسی را دارا باشند: (۱) یادگیری سریع، (۲) یادگیری تطبیقی

<sup>17</sup> time-varying non-stationary

<sup>18</sup> Batch

<sup>19</sup> real-time

<sup>20</sup> Multi-layer Perceptron

<sup>21</sup> Radial Basis Function

<sup>22</sup> self-organizing maps

<sup>23</sup> Evolving Fuzzy systems

<sup>24</sup> One-pass

<sup>25</sup> Consequent

<sup>26</sup> Global Optimized Dynamic Fuzzy Inference System

مدل دقیق، عدم قطعیت‌ها و اغتشاشات مدل نشده، تأثیرات منفی قابل توجهی بر عملکرد آن‌ها می‌گذارند.

ماهیت غیرخطی سیستم‌های دینامیکی باعث شد که محققان به سمت کنترلرهای غیرخطی مختلفی مانند کنترلرهای پس‌گرد، لغزشی، خطی‌سازی بازخورد (FBL)<sup>۲۷</sup> و تکنیک‌های کنترل استوار  $H^\infty$  روی آوردند. با این حال، نقطه ضعف اصلی تمام این کنترلرهای خطی و غیرخطی، وابستگی شدید آن‌ها به مدل دقیق دینامیکی سیستم تحت کنترل است که یک محدودیت اساسی محسوب می‌شود، چرا که مدل‌سازی عدم قطعیت‌هایی مانند نویز سنسورها، اختلالات محیطی و سایر عوامل نامعین بسیار دشوار است. این چالش‌ها، محققان را به سمت تکنیک‌های کنترل مبتنی بر دانش بدون نیاز به مدل سوق داده است.

اولین تلاش‌ها در زمینه کنترل بدون مدل مولتی روتورها با استفاده از کنترلرهای FLS [۵۱]، ANN [۵۲]، و NF [۵۳] آغاز شد. اما نیاز به آموزش آفلاین، ساختار غیر قابل تغییر و وابستگی به دانش متخصصان، کاربرد این روش‌ها را در سیستم‌های پیچیده با محیط نامعین محدود ساخت. برای غلبه بر این محدودیت، پژوهشگران به توسعه کنترلرهای تطبیقی هوشمند مبتنی بر FLS، NN و NF روی آوردند [۵۴]. با این حال، این کنترلرها نیز فاقد قابلیت تغییر ساختار به صورت پویا بودند و اگرچه عملکرد آن‌ها در سیستم‌های پیچیده بهبود یافت اما همچنان با محدودیت‌های قابل توجهی در کنترل سیستم‌های پیچیده مواجه بودند.

برای حل این مسئله، کنترلرهای هوشمند تکاملی به عنوان راه‌حلی امیدبخش برای مدیریت سیستم‌های دینامیکی پیچیده با عدم قطعیت مطرح شدند. تحقیقات در این زمینه از اوایل قرن ۲۱ آغاز شد. در سال ۲۰۰۳، یک کنترلر خودسازمانده فازی-عصبی در [۴۳] معرفی شد که از معیارهای خطای سیستم، کامل بودن و نرخ کاهش خطا در مکانیزم تکامل قوانین استفاده می‌کرد. اما نیاز این کنترلر به ذخیره تمام داده‌های ورودی-خروجی قبلی و محاسبه ماتریس‌های بزرگ در هر مرحله، باعث افزایش قابل توجه بار محاسباتی و نیاز به منابع سخت‌افزاری قوی شد. اجرای بسیاری از این کنترل‌کننده‌ها در عمل با محدودیت‌هایی مواجه است، چرا که نیاز به حافظه بالا و توان پردازشی قابل توجه به دلیل حجم

قوانین، و روش‌های به‌روزرسانی برای پارامترهای مقدم و تالی (شامل به‌روزرسانی مراکز خوشه‌ها، شعاع خوشه‌ها، و ضرایب خطی در بخش تالی) از جریان داده ورودی می‌باشد. از منظر بهینه‌سازی، هدف اصلی این مقاله توسعه الگوریتمی است که بتواند تابع خطای کلی سیستم فازی را با در نظر گرفتن تغییرات ساختاری و پارامتری به حداقل برساند.

شایان ذکر است که تا کنون EFS‌های متعددی برای کاربردهای متنوع در دنیای واقعی توسعه یافته‌اند [۳۷]. با توجه به محدودیت فضا، تمرکز این مقاله صرفاً بر نمایندگان شاخص سیستم‌های فازی در حال تکامل و به‌ویژه رویکردهایی است که به جنبه‌های بهینه‌سازی این سیستم‌ها پرداخته‌اند.

## مرور ادبیات

این بخش به بررسی جامع مطالعات پیشین در حوزه مرتبط با پژوهش حاضر اختصاص دارد. با توجه به دوگانه بودن اهداف این مقاله که شامل "بررسی کاربرد EFS‌ها در کنترل مولتی روتورها" و "ارائه یک الگوریتم جدید EFS" می‌باشد. مرور ادبیات به صورت ساختاریافته در دو بخش مجزا انجام می‌گیرد:

- ۱) بررسی مطالعات مربوط به کاربرد EFS‌ها در کنترل مولتی روتورها

### ۲) بررسی مطالعات توسعه‌ای در حوزه الگوریتم‌های EFS

این تقسیم‌بندی امکان بررسی دقیق‌تر هر یک از جنبه‌های تحقیق را فراهم می‌سازد و جایگاه پژوهش حاضر را در میان مطالعات پیشین به وضوح مشخص می‌نماید

## کاربرد EFS‌ها در حوزه کنترل مولتی روتورها

در دهه‌های اخیر، کنترلرهای خطی مبتنی بر FPT مانند روش‌های PID به دلیل سادگی طراحی و پیاده‌سازی، به طور گسترده در کنترل مولتی روتورها مورد استفاده قرار گرفته‌اند [۴۸]. اگرچه این کنترلرها در محیط‌های خاص عملکرد رضایت‌بخشی دارند؛ اما وجود بهره ثابت در آن‌ها باعث می‌شود در برابر عدم قطعیت‌های محیطی مانند وزش باد مقاوم نباشند. از دیگر روش‌های کنترل خطی پرکاربرد می‌توان به تکنیک LQ (تکنیک خطی درجه دوم) اشاره کرد [۵۰]. این کنترلرهای خطی، به دلیل ماهیت غیرخطی و پیچیده سیستم‌های دینامیکی، اغلب قادر به دستیابی به عملکرد مطلوب نیستند. علاوه بر این، به دلیل وابستگی این کنترلرها به

<sup>27</sup> Feedback Linearization

محاسباتی طراحی شده است. سپس، در مقاله [۴۸] الگوریتم PALM با افزودن یک ماژول ساده تطبیق قوانین و مکانیزم تکامل قوانین مبتنی بر "اهمیت شبکه"، به الگوریتم PAC توسعه یافت. به طور خلاصه، مقایسه‌ای میان کنترلرها موجود در جدول ۱ ارائه شده است.

جدول ۱. مقایسه بین کنترلرها موجود در ادبیات

نوع کنترلر ویژگی	PID	بدون مدل (FLS, NN, NF)	تطبیقی هوشمند	EFS
نیاز به مدل سیستم	دارد (خطی ساده)	ندارد (داده یا قوانین تجربی)	تا حدی / تطبیق آنلاین	ندارد / آنلاین و خودسازمانده
نوع کنترلر	خطی و ثابت (برای سیستم‌های خطی ساده مناسب)	غیرخطی و هوشمند (سیستم‌های پیچیده و غیرخطی)	تطبیقی و هوشمند (پارامترها تغییر می‌کنند)	فازی و در حال تکامل (ساختار و پارامترها خودکار تغییر می‌کنند)
غیرخطی / نامطمئن	ضعیف	خوب	خوب	بسیار خوب
قابلیت یادگیری	ندارد	دارد (ساختار ثابت)	دارد (پارامتری تطبیقی)	دارد (پارامتری + ساختاری)
انعطاف پذیری	پایین (پارامترها ثابت؛ نیاز به تنظیم مجدد)	متوسط (یادگیری از داده‌ها، اما ساختار ثابت)	بالا (تطبیق پارامترها)	بسیار بالا (خود- توسعه‌دهنده؛ قوانین و ساختار تکامل می‌یابند)
پیچیدگی محاسباتی	کم	متوسط تا زیاد	زیاد	زیاد
قابلیت آنلاین	ندارد	ندارد	ندارد	دارد
نیاز به داده- های تاریخی	-	دارد	دارد	ندارد
کاربرد صنعتی	بسیار رایج	محدود به پروژه‌های خاص	پیشرفته / پروژه‌های هوشمند	سیستم‌های پیچیده و پویا

به زبان ساده می‌توان عملکرد کنترل‌کننده‌های مختلف را به راننده‌ای تشبیه کرد که ویژگی‌های زیر را دارند:  
✓ PID مثل یک راننده با تجربه در جاده صاف است: سریع و ساده، ولی در تغییرات ناگهانی (مثل پیچ‌های تند) ضعیف.

زیاد محاسبات دارند. این مسئله به‌ویژه در سیستم‌هایی که نیازمند پاسخ سریع و آنی از کنترل‌کننده هستند (برای پیروی از دستورات مطلوب)، مشکل‌ساز می‌شود.  
یکی از نمونه‌های کنترل‌کننده‌های در حال تکامل در مطالعه [۳۸] ارائه شده که بر پایه مدل فازی در حال تکامل تاکاگی-سوگنو (eTS) [۳۵] توسعه یافته است. هرچند این کنترلر از ماهیت در حال تکامل برخوردار است، اما با چندین محدودیت مهم روبروست. نخست، در سازوکار تکامل ساختاری آن، قوانین فازی تنها افزوده یا جایگزین می‌شوند و فرایند هرس قوانین در نظر گرفته نشده است. علاوه‌براین، کنترلر eTS نیازمند نگهداری داده‌های پیشین است، که این امر منجر به افزایش بار محاسباتی و ناکارآمدی آن در کاربردهای واکنشی سریع شده است. در مطالعه [۴۴]، یک کنترل‌کننده ترکیبی با ترکیب سیستم فازی TS در حال تکامل و نسخه‌ای ایستا از آن معرفی شد. اگرچه طراحی آن ساده بود، اما نیازمند دانستن برخی از پارامترهای دستگاه بود؛ پارامترهایی که در بسیاری از موارد، در حین عملیات کنترل، در دسترس نیستند. در مطالعه [۴۵] نیز یک کنترل‌کننده در حال تکامل با استفاده از تکنیک کنترل پیش‌بینی مدل ارائه شد. با این حال، وابستگی آن‌ها به مدل دینامیکی دستگاه، کاربرد آن‌ها را در سیستم‌های پیچیده غیرخطی محدود می‌کند.

از دیگر مقالات می‌توان به مقاله [۵۵] اشاره کرد که یک ساختار NF نوع ۲ (IT2) برای کنترل یک مولتی روتور ارائه شد. اما در این ساختار، سیستم IT2NF صرفاً نقش ناظر اغتشاشات و عدم قطعیت را ایفا می‌کرد، در حالی که کنترل وضعیت و موقعیت پرنده با یک کنترل‌کننده PD انجام می‌گرفت. در مطالعه [۵۶]، نیز یک کنترل‌کننده فازی در حال تکامل برای ردیابی مسیر یک نیروگاه مولتی روتور شبیه‌سازی شده، معرفی شد. اگرچه این کنترل‌کننده موفق به تولید هرس قوانین فازی در حین پرواز با دقت قابل قبول شد، تنها بر اساس یک مدل دینامیکی ساده‌شده از مولتی روتور عمل می‌کرد. در مقاله [۴۷] ساختار خوشه‌بندی بیضوی به جای خوشه‌بندی ابرکراهی در طراحی کنترل‌کننده فازی در حال تکامل به کار گرفته شد تا با استفاده از خوشه‌بندی مبتنی بر داده، تعداد پارامترهای کنترلی کاهش یابد. در همین راستا، در مقاله [۴۶] الگوریتم PALM را معرفی کرد که از خوشه‌بندی ابرصفحه‌ای استفاده می‌کند و با هدف کاهش پیچیدگی

[۵۷] پیشنهاد می‌شود. این مقاله به بررسی کاربرد الگوریتم‌های فازی در سیستم‌های خودران و مولتی روتورها پرداخته است. همچنین، مرور مطالعات [۴، ۵۸] نیز برای تکمیل دیدگاه مفید خواهد بود.

با وجود موفقیت‌های چشمگیر EFSها در محیط‌های عملیاتی، این سیستم‌ها علاوه بر بار محاسباتی، با چالش اساسی دیگری نیز مواجه‌اند. بخش‌های «اگر» (مقدم<sup>۲۸</sup>) و «آنگاه» (تالی<sup>۲۹</sup>) این سیستم‌ها که از جریان داده‌ها آموخته می‌شوند، معمولاً بهینه نیستند. دلیل اصلی این مسئله، استفاده از فرایند یادگیری تک‌گذری و نبود سازوکارهای بهینه‌سازی در مرحله‌ی آموزش است. در نتیجه این روش‌ها قادر به جستجوی تکرارشونده برای یافتن راه حل بهینه نیستند [۴۹]. این محدودیت، افت محسوسی در عملکرد EFSها ایجاد کرده و در سال‌های اخیر توجه پژوهشگران بسیاری را به خود جلب کرده است. بر این اساس، پژوهش حاضر با هدف رفع این محدودیت، الگوریتم جدیدی را معرفی می‌کند. در ادامه به مرور مطالعات پیشین در حوزه EFSها می‌پردازیم و نقاط قوت و ضعف آن‌ها مورد تحلیل قرار خواهد داد.

### الگوریتم‌های در حال تکامل

با مرور ادبیات می‌توان متوجه شد که رویکردهای مختلفی برای مدل‌سازی سیستم‌های عصبی-فازی و طراحی پایگاه قوانین فازی در حال تکامل ارائه شده‌اند که اغلب بر توسعه‌ای بر مدل‌های تاکاگی-سوگنو (TS) می‌باشند. اولین الگوریتم‌های آنلاین برای شناسایی EFSها در حدود سال ۲۰۰۰ مطرح شدند. از جمله آن‌ها می‌توان به شبکه‌های عصبی فازی در حال تکامل (EFuNN) اشاره کرد که توسط [۵۹] ارائه شد. این الگوریتم، مبتنی بر یادگیری محلی است و قابلیت یادگیری سریع از داده‌ها را دارد.

در ادامه، مطالعه [60] رویکردی غیرتکراری برای شناسایی سیستم‌های فازی در حال تکامل پیشنهاد کرد که در آن با استفاده از یادگیری افزایشی بدون نظارت، امکان جایگزینی قوانین قدیمی با قوانین جدید بر پایه پتانسیل اطلاعاتی فراهم گردید. سیستم استنتاج فازی عصبی در حال تکامل پویا

✓ کنترل‌کننده بدون مدل مثل راننده‌ای است که نقشه راه را تقریباً بلد است: عملکرد بهتری از PID دارد، ولی دقت آن همواره تضمین شده نیست.

✓ کنترل‌کننده هوشمند تطبیقی مثل راننده‌ای است که حین رانندگی تجربه کسب می‌کند: عملکرد پایدارتر و دقیق‌تر نسبت به کنترل‌کننده بدون مدل

✓ EFS مانند راننده‌ای است که نه تنها تجربه کسب می‌کند، بلکه خودرو خود را نیز در طول مسیر ارتقا می‌دهد: عالی‌ترین سطح از نظر دقت و توانایی در تطبیق ولی از نظر محاسباتی پرهزینه‌ترین روش

با توجه به موارد ذکر شده، EFSها مزایای قابل‌توجهی در حل چالش‌های شناسایی و کنترل مولتی روتورها ارائه می‌دهند که عبارت‌اند از:

✓ عدم نیاز به مدل دینامیکی سیستم: این سیستم‌ها بدون نیاز به مدل‌سازی دقیق دینامیک سیستم عمل می‌کنند که ویژگی قابل توجهی برای سیستم‌های ناشناخته می‌باشد.

✓ کارایی بالا در مواجهه با سیستم‌های غیرخطی و نامطمئن: این قابلیت در مدیریت سیستم‌های پیچیده و غیر قابل پیش‌بینی بسیار حائز اهمیت است.

✓ انعطاف‌پذیری بالا در شرایط مختلف: این روش‌ها قابلیت تطبیق با سناریوهای متنوع عملیاتی را دارا هستند.

✓ یادگیری آنلاین و عدم نیاز به داده‌های تاریخی: این ویژگی برای سیستم‌های خودران با منابع محاسباتی محدود، بسیار حیاتی است.

✓ کاربری در محیط‌های پویا و پیچیده: توانایی عملکرد در شرایط پویا و غیر ایده‌آل، EFSها را به گزینه‌ای منحصربه‌فرد برای کنترل و شناسایی مولتی روتورها تبدیل می‌کند.

این ویژگی‌ها EFSها را به یک راهکار کارآمد و سازگار برای چالش‌های کنترلی در مولتی روتورها و سیستم‌های مشابه تبدیل کرده است. اگرچه چالش‌هایی نظیر بار محاسباتی سنگین از معایب این روش محسوب می‌شود، اما پژوهش حاضر با معرفی یک الگوریتم پیشرفته EFS، راهکارهای عملی برای کاهش این محدودیت‌ها پیشنهاد می‌کند. برای آشنایی بیشتر با پیشرفت‌های حوزه کنترل‌های فازی، مطالعه مقاله‌ی مروری

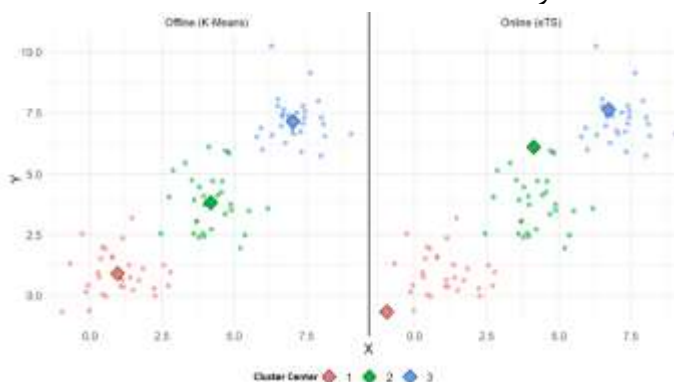
<sup>28</sup> Antecedent

<sup>29</sup> Consequent

محافظه کارانه برای افزودن قوانین فازی با آستانه‌های خودسازگار بهره می‌گیرد.

از دیگر تحقیقات قابل ذکر اخیر می‌توان به [۸۸-۹۷] اشاره کرد. برای مطالعه بیشتر، منابع جامعی از رویکردهای تکاملی را می‌توان در مقالات [۳۷، ۱۰۲-۹۸] یافت.

بررسی جامع این مطالعات نشان می‌دهد که اکثر الگوریتم‌های EFS مبتنی بر یادگیری تک‌گذری هستند و ساختار و پارامترهای خود را به صورت پویا و بر اساس بازخوردهای پیش‌بینی حاصل از جریان داده‌ها به‌روزرسانی می‌کنند. معمولاً تأکید اصلی EFS‌های موجود بر توانایی پاسخگویی سریع به تغییرات و انحرافات در الگوهای داده‌های جریانی است. این قابلیت به دلیل ماهیت غیرخطی و غیرثابت<sup>۳۸</sup> جریان‌های داده در کاربردهای واقعی، نقش کلیدی در موفقیت EFS‌ها ایفا می‌کند. با این وجود، بسیاری از این سیستم‌ها به دلیل پدیده «اثر بی‌آموزی»<sup>۳۹</sup> از دقت پیش‌بینی سراسری پایینی رنج می‌برند هستند [۱۰۳]. این مسئله ناشی از عدم بهینگی پارامترهای مقدم و تالی در EFS‌ها است، چرا که ساختار و پارامترهای سیستم عمدتاً به شیوه‌ای اکتشافی و خودکار توسعه می‌یابند و عدم وجود فرایند بهینه‌سازی پارامترها در فرایند یادگیری EFS‌ها، معمولاً منجر به از دست رفتن اطلاعات ارزشمند موجود در داده‌های آموزشی می‌گردد. در شکل ۳ می‌توان تفاوت یک الگوریتم آفلاین که مکانیزم بهینه‌سازی دارد و یک الگوریتم آنلاین که این مکانیزم را ندارد مشاهده کرد.



شکل ۳. مقایسه میان مراکز خوشه‌ها در روش آفلاین و آنلاین

(DENFIS)<sup>۳۰</sup>، معرفی شده توسط [36]، نیز از خوشه‌بندی بازگشتی و حداقل مربعات وزنی با عامل فراموشی برای به‌روزرسانی ساختار و پارامترهای پایگاه قوانین بهره می‌برد. بر مبنای این آثار، مدل eTS<sup>۳۱</sup> (تاکاگی-سوگنو در حال تکامل) توسط [۳۵] توسعه یافت که ترکیبی از یادگیری نظارت‌شده و بدون نظارت است و امکان افزودن و به‌روزرسانی قوانین را فراهم می‌سازد. توسعه‌های متعددی بر eTS صورت گرفته، از جمله Simpl\_eTS [۶۱]، eXtended eTS (xTS) [۶۲]، eTS+ [۶۳]، طبقه‌بندی‌کننده در حال تکامل مبتنی بر مدل eTS [۶۴] و مدل DeTS [۶۵] اشاره کرد. که همگی از معیارهایی مانند پتانسیل، چگالی و پراکندگی برای خوشه‌بندی داده‌ها استفاده می‌کنند. این مدل‌ها با استفاده از معیار پتانسیل/تراکم و پراکندگی به خوشه‌بندی فضای داده‌های ورودی می‌پردازند. از دیگر الگوریتم‌های مهم می‌توان به موارد زیر اشاره کرد:

FLEXFIS<sup>۳۲</sup> [۳۹]، SOFMLS [۶۶]، eMG [۶۷]، McFIS<sup>۳۳</sup> [۶۸]، SAFIS [۶۹]، ESAFIS [۷۰]، PANFIS [۷۱]، GENEFIS [۷۲]، GSETSK [۷۳]، Gen-Smart- [۷۴]، EFS [۷۴]، rFCM<sup>۳۴</sup> [۷۵]، rGK<sup>۳۵</sup> [۷۶]، eFuMo<sup>۳۶</sup> [۷۷]، ePL<sup>۳۷</sup> [۷۸]، ePL+ [۷۹]، rPFM [۸۰]، ePFM [۸۱].

در سال‌های اخیر، مدل‌های پیشرفته‌تری توسط محققین توسعه یافته‌اند. به‌عنوان نمونه، مدل eT2RFNN [۸۲] که برای بهبود استحکام در مواجهه با عدم قطعیت پیشنهاد شد. سیستم عصبی فازی در حال تکامل مبتنی بر همسان‌تروپی (CEFNS) [۸۳] و نسخه پیشرفته‌تر آن RMCEFS [۸۴]، از معیارهای خطای مکمل بهره می‌برند. همچنین، الگوریتم SPATFIS [۸۵] برای مدل‌سازی فازی زمانی-مکانی توسعه یافته که از بیشینه عضویت و شاخص شباهت برای تولید و ادغام قوانین استفاده می‌کند. الگوریتم SEFS [۸۶] نیز با تنظیم پویای آستانه بر اساس خطاهای آموزش، فرایند تولید قوانین را مدیریت می‌کند. مدل EFS-SLAT [۸۷] از استراتژی

<sup>30</sup> Dynamic evolving neural-fuzzy inference system

<sup>31</sup> Evolving Takagi-Sugeno

<sup>32</sup> Flexible fuzzy inference systems

<sup>33</sup> Meta-cognitive neuro-fuzzy inference system

<sup>34</sup> Recursive fuzzy c-means

<sup>35</sup> Recursive Gustafson-Kessel

<sup>36</sup> Evolving Fuzzy Model

<sup>37</sup> Evolving Participatory Learning

<sup>38</sup> None stationary

<sup>39</sup> unlearning effect

بنابراین، با اعمال روش‌های بهینه‌سازی مناسب، می‌توان شاهد بهبود چشمگیر در دقت پیش‌بینی این سیستم‌ها بود [۴۹، ۱۰۴]. این بهبود به ویژه در کاربردهای عملی که نیازمند دقت بالا هستند، حائز اهمیت است.

اخیراً توجه فزاینده‌ای به تحلیل بهینگی و پایداری EFS‌ها شده است [۱۰۵]. پایداری بخش تالی EFS‌ها در مطالعه [۱۰۶] مورد بررسی قرار گرفته است. به طور خاص، در [۱۰۳] یک رویکرد بهینه‌سازی خطای محلی برای شناسایی سیستم فازی پیشنهاد شده است. در [۱۰۷]، بهینه بودن EFS‌ها به طور سیستماتیک مورد مطالعه قرار گرفته و دو الگوریتم به منظور بهینه‌سازی قسمت‌های مقدم و تالی معرفی شده‌اند. با این وجود، دقت پیش‌بینی تنها در برخی از مسائل پس از بهینه‌سازی سیستم به طور جزئی بهبود می‌یابد. در مقاله [۴۹]، از الگوریتم بهینه‌سازی ازدحام ذرات (PSO)<sup>۴۰</sup> به منظور بهینه‌سازی یک EFS با نام ALMMo که در مقاله [۱۰۸] ارائه شده، استفاده کردند. الگوریتم آن‌ها (PSO-ALMMo) به لطف فرایند تکراری<sup>۴۱</sup> بهینه‌سازی پارامترها، در مقایسه با ALMMo، پیش‌بینی‌های دقیق‌تری ارائه می‌دهد.

با این حال، فرایند بهینه‌سازی پارامترها، به دلیل استفاده از محاسبات تکاملی، ممکن است با توجه به پیچیدگی مسئله و ابعاد فضای جستجو، زمان قابل‌توجهی تا همگرایی به یک جواب بهینه محلی نیاز داشته باشد. از این رو، این فرایند باید به صورت آفلاین انجام شود. به عبارت دیگر، علی‌رغم وجود دقت بیشتر، PSO-ALMMo تنها به سناریوهای آفلاین محدود می‌شود و در کاربری‌های آفلاین قابل استفاده نیست. در مطالعه‌ای دیگر [۱۰۴]، یک EFS مبتنی بر بهینه‌سازی ازدحام ذرات با استفاده از اطلاعات چند جمعیتی (MIIPSO-EFS) پیشنهاد شده است. این مدل دارای دو مرحله یادگیری است: (۱) یادگیری ساختاری: ایجاد پایگاه قوانین فازی، (۲) به‌روزرسانی پارامترها: تنظیم دقیق مدل با توجه به جریان داده. این الگوریتم نیز به دلیل استفاده از محاسبات تکاملی، محدود به سناریوهای آفلاین بوده و برای همگرایی، با توجه به ابعاد مسئله، به زمان زیادی نیاز دارد.

همان‌طور که مشاهده می‌شود، علی‌رغم وجود تعدد زیاد EFS‌ها، تحقیقات بسیار کمی به بهینه‌سازی EFS‌ها پرداخته-

اند. بیشتر این پژوهش‌ها از محاسبات تکاملی برای بهینه‌سازی استفاده کرده‌اند که باعث شده EFS‌ها محدود به سناریوهای آفلاین شوند و نیازمند زمان زیاد برای رسیدن به همگرایی باشند. در تنها مقاله‌ای که رویکرد محاسبات تکاملی ندارد [۱۰۷] نیز بهبود دقت پیش‌بینی تنها در برخی مسائل و به صورت جزئی مشاهده شده است.

در این پژوهش، برای نخستین بار یک سیستم استنتاج فازی در حال تکامل با قابلیت بهینه‌سازی سراسری هم‌زمان بخش مقدم و تالی در محیط آنلاین ارائه شده است. هسته اصلی این سیستم، الگوریتم خوشه‌بندی بازگشتی GOECA<sup>۴۲</sup> است که با الهام از روش‌های کلاسیک بهینه‌سازی و بر اساس مفهوم مرکز ثقل بازگشتی، پارامترهای مقدم را به صورت آنلاین و بدون نیاز به داده‌های تاریخی، بهینه می‌کند. در بخش تالی، برخلاف اغلب EFS‌های موجود که از به‌روزرسانی محلی بهره می‌برند، یک رویکرد بهینه‌سازی سراسری مبتنی بر RLS به کار گرفته شده که باعث افزایش معنادار دقت پیش‌بینی می‌شود، و بهینگی کلی ساختار سیستم را حفظ می‌کند. علاوه بر این، معیار جدیدی تحت عنوان «اصل حذف نویز»<sup>۴۳</sup> (NEP) به عنوان مکانیزم هرس<sup>۴۴</sup> و ساده‌سازی قوانین ارائه می‌شود. این مکانیزم با ترکیب دو شرط «پایین بودن کیفیت خوشه» و «کم بودن تعداد داده‌های تشکیل دهنده» به شناسایی قوانین نویزی می‌پردازد و ضمن حفظ بهینگی سیستم، اثر منفی نقاط نویزی را خنثی کرده و از پیچیدگی غیرضروری ساختار جلوگیری می‌کند. این ترکیب از نوآوری‌ها منجر به سیستمی می‌شود که در عین انعطاف‌پذیری ساختاری و سرعت پردازش بالا، دقت و پایداری چشمگیری در کاربردهای آنلاین ارائه می‌دهد. علاوه بر تمام موارد گفته شده، GODFIS به طور ویژه برای شناسایی و کنترل دینامیک‌های غیرخطی مولتی روتورها طراحی و بهینه‌سازی شده است و توانایی آن در تطبیق با تغییرات ناگهانی در رفتار پرواز، و همچنین شناسایی داده‌های نویزی به طوری که کیفیت شناسایی و کنترل دینامیک پرواز به طور بهینه مدیریت شود، یک نوآوری کلیدی است همچنین با به کارگیری بهینه‌سازی سراسری در تنظیم پارامترهای تالی، ساختار بهینه کلی سیستم را بهبود می‌بخشد.

<sup>42</sup> Global Optimization-based Evolving Clustering Algorithm

<sup>43</sup> Noise Elimination Principle

<sup>44</sup> Pruning Mechanism

<sup>40</sup> Particle Swarm Optimization

<sup>41</sup> Iteration based process

مهم ترین دستاوردها و مشارکت های کلیدی این پژوهش عبارتند از:

- (۱) ارائه الگوریتم خوشه بندی بازگشتی نوین با نام GOECA برای به روزرسانی پارامترهای مقدم که بر پایه مفاهیم روش های بهینه سازی کلاسیک استوار است. این الگوریتم به شکل محسوسی در یافتن نقاط بهینه پارامترهای خوشه ها مؤثر عمل می کند. بدین ترتیب بدون نیاز به داده های تاریخی یا محاسبات تکاملی زمان بر و همچنین با حفظ قابلیت کاربرد آنلاین مدل، می توان به راه حل های نزدیک به بهینه دست یافت.
- (۲) پژوهش حاضر معیار جدیدی با عنوان «اصل حذف نویز» برای هرس قوانین معرفی می کند که از یک سو بهینه بودن سراسری سیستم را حفظ کرده و از سوی دیگر اثرات نامطلوب نقاط نویزی بر ساختار سیستم را خنثی کرده و از پیچیدگی ساختار جلوگیری می نماید.
- (۳) تنظیم بهینه هم زمان مان های تعیین مقدم و تالی، به طوری که منجر به تولید نتایج برجسته تری در مقایسه با دیگر روش های آنلاین موجود در ادبیات پژوهشی شده است.
- (۴) GODFIS به طور خاص برای شناسایی و کنترل دینامیک های غیرخطی مولتی روتورها طراحی و بهینه سازی شده است. این روش برای اولین بار برای مدیریت بهینه دینامیک پرواز ارائه می شود با قابلیت سازگاری با تغییرات ناگهانی در رفتار پرواز و شناسایی دقیق داده های نویزی، کیفیت برتر شناسایی و کنترل دینامیک پرواز را تضمین می کند.

### بیان مسئله

یک سیستم فازی استاندارد که بر اساس قوانین فازی کار می کند، از چهار بخش اصلی تشکیل شده است [۳۷، ۹۹] (شکل ۴):

- (۱) **فازی ساز**<sup>۴۶</sup>: این بخش ورودی واضح عددی را به ورودی های فازی نگاشت می کند.

علاوه بر این، معیار جدید به نام «اصل حذف نویز» به عنوان مکانیزم هرس<sup>۴۵</sup> و ساده سازی قوانین ارائه می شود که ضمن حفظ بهینگی سیستم، تأثیر منفی نقاط نویزی را خنثی کرده و از پیچیدگی ساختار جلوگیری می کند. تحلیل های نظری و نتایج تجربی حاصل از پیاده سازی الگوریتم GODFIS بر روی مسائل معیار، نشان دهنده برتری قابل توجه این الگوریتم از نظر دقت پیش بینی و همچنین سرعت نسبتاً مناسب آن در مقایسه با روش های موجود است. خلاصه مقایسه بین EFS های موجود و الگوریتم پیشنهادی در جدول ۲ ارائه شده است.

جدول ۲. خلاصه مقایسه بین EFS های موجود و الگوریتم GODFIS

روش	EFS کلاسیک (مثل [35]eTS، [78]ePL، [81]ePFM)	EFS بهینه سازی آفلاین (مثل PSO-ALMMo [49])	روش پیشنهادی GODFIS
بهینه سازی بخش مقدم	خوشه بندی اکتشافی، بدون بهینه سازی دقیق مراکز و شعاع ها	بهینه سازی آفلاین، نیازمند داده های تاریخی کامل	بهینه سازی آفلاین و بدون نیاز به داده تاریخی
بهینه سازی بخش تالی	به روزرسانی محلی (wRLS) ← دقت متوسط	بهینه سازی تکاملی فقط در حالت آفلاین و زمان محاسباتی خیلی زیاد	بهینه سازی سراسری آنلاین با RLS، هماهنگ با مقدم بهینه شده
هرس قوانین	ندارند (مثل [35]eTS) دارند (مثل [63]eTS+)	معمولاً فاقد مکانیزم هرس قوانین نویزی یا کم کاربرد	اصل حذف نویز: حذف قوانین نویزی یا کم کاربرد
قابلیت کاربرد آنلاین	بله، ولی دقت و پایداری کمتر	خیر، فقط محدود به سناریو آفلاین	بله، حفظ هم زمان دقت، پایداری و انعطاف پذیری ساختاری
نیاز به داده های تاریخی کامل	خیر/بله	بله	خیر
پیچیدگی محاسباتی	پایین، ولی به بهای کاهش دقت	بالا، به دلیل محاسبات تکاملی	بهینه و سبک، به دلیل استفاده از روابط بازگشتی در مقدم و تالی

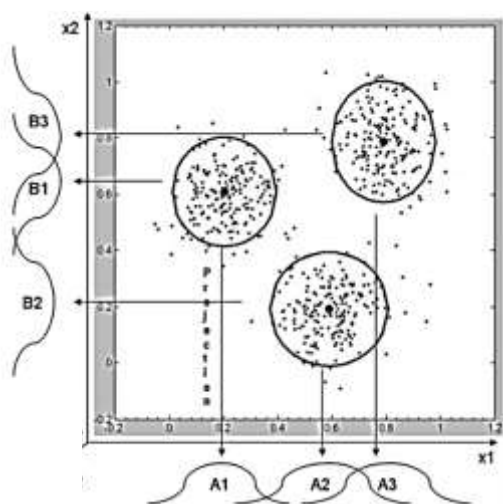
<sup>46</sup> fuzzifier

<sup>45</sup> Pruning Mechanism

$$y = \sum_{i=1}^r \lambda_i * y_i = \sum_{i=1}^r \lambda_i * \mathbf{x}_e^T \theta_i \quad (3)$$

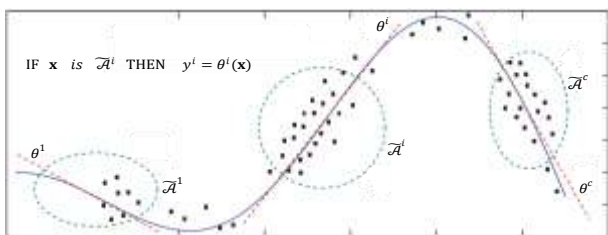
$$\lambda_i = \frac{\gamma_i(\mathbf{x})}{\sum_{j=1}^r \gamma_j(\mathbf{x})}; \quad \mathbf{x}_e^T = [1 \ \mathbf{x}^T] \quad (4)$$

پارامترهای بخش شرطی قوانین از طریق خوشه‌بندی فضای ورودی تعیین می‌شوند. شکل ۵ یک مثال سه بعدی (شامل دو ورودی  $x_1$  و  $x_2$  و یک خروجی  $y$ ) را نشان می‌دهد: سه خوشه به دو محور ( $x_1$  و  $x_2$ ) تصویر شده‌اند که فضای ورودی و قسمت مقدم قوانین را تشکیل می‌دهند [۳۹].



شکل ۵. تصویر افقی سه خوشه بر روی محورهای ورودی  $x_1$  و  $x_2$

همان‌طور که پیش‌تر اشاره شد، مدل TS از نواحی فازی پارامتری شده استفاده می‌کند و هر ناحیه را به یک مدل محلی مرتبط می‌سازد. رفتار غیرخطی این مدل در واقع از ترکیب هوشمندانه و وزن‌دار چندین مدل محلی به دست می‌آید. نکته مهم این است که میزان تأثیر هر مدل محلی در خروجی نهایی، مستقیماً به میزان فعال شدن آن مدل بستگی دارد. برای درک بهتر این مکانیزم، می‌توانید به شکل ۶ مراجعه کنید که چگونگی تقریب توابع توسط این مدل را به تصویر کشیده است.

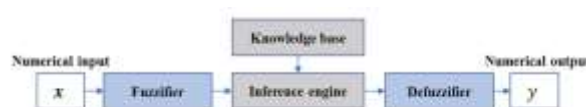


شکل ۶. مدل‌سازی TS برای تقریب تابع [81]

(۲) **موتور استنتاج**<sup>۴۷</sup>: در این بخش، با استفاده از قوانین فازی ذخیره‌شده در سیستم، مقادیر فازی ورودی به مقادیر فازی خروجی تبدیل می‌شوند.

(۳) **پایگاه دانش**<sup>۴۸</sup>: این بخش شامل تمامی قوانین فازی است که سیستم برای پردازش اطلاعات از آن‌ها استفاده می‌کند.

(۴) **غیرفازی‌ساز**<sup>۴۹</sup>: این بخش مقادیر فازی خروجی را به مقادیر عددی و دقیق تبدیل می‌کند.



شکل ۴. یک سیستم استاندارد مبتنی بر قوانین فازی

### مدل فازی تکاگی-سوگنو (TS)

در این پژوهش از مدل فازی TS نوع یک استفاده شده است و ساختار قانون فازی  $\tilde{A}_i$  به صورت فرمول (۱) تعریف می‌شود:

$$\begin{aligned} R_i: \text{IF } \mathbf{x} \text{ is } \tilde{A}_i \text{ THEN} \\ y_i = \theta_i(\mathbf{x}) = \mathbf{x}_e^T \theta_i = \\ \theta_{i0} + \theta_{i1}x_1 + \dots + \theta_{in}x_n \end{aligned} \quad (1)$$

که در این رابطه،  $R_i$  آمین قانون فازی  $i = 1, 2, \dots, r$  تعداد قوانین فازی سیستم،  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$  ورودی به سیستم،  $\tilde{A}_i = [\tilde{A}_{i1}, \tilde{A}_{i2}, \dots, \tilde{A}_{in}]$  مجموعه فازی بخش مقدم قانون فازی  $\tilde{A}_i$  با توابع عضویت  $\mu_{ij}(x_j)$ ،  $y_i \in \mathbb{R}$  خروجی قانون  $\tilde{A}_i$  و  $\theta_{i0}, \theta_{i1}, \dots, \theta_{in}$  پارامترهای تالی قانون  $\tilde{A}_i$  هستند.

فرمول قدرت فعال‌سازی (درجه فعال‌سازی نرمال) هر قانون فازی  $(\gamma_i)$  با فرمول (۲) محاسبه می‌شود و خروجی نهایی از ترکیب وزنی خروجی‌های تمام قوانین به دست می‌آید (فرمول-های (۳) و (۴)).

$$\gamma_i(\mathbf{x}) = \prod_{j=1}^n \mu_{ij}(x_j) \quad (2)$$

<sup>47</sup> Inference Engine  
<sup>48</sup> Knowledge Base  
<sup>49</sup> Defuzzifier

قابلیت درک آسان و تفسیرپذیری بالا در حوزه‌های مختلف مهندسی مورد توجه محققان قرار گرفت [۶، ۱۱۲-۱۰۹].  
ECM یک روش خوشه‌بندی آنلاین سریع است، که نیازی به تعداد خوشه‌های از پیش تعیین شده ندارد، زیرا آن‌ها به طور خودکار از جریان‌های داده بر اساس ویژگی‌هایشان تکامل می‌یابند. مراحل زیر جزئیات فرایند خوشه‌بندی ECM را در یک فضای دوبعدی را که در شکل ۷ آمده است شرح می‌دهد:

**گام ۱ - ایجاد اولین خوشه:** خوشه اول ( $C_1^0$ ) با استفاده از اولین نمونه داده‌ها ایجاد می‌شود و مرکز ( $C_1^0$ ) و شعاع آن ( $Ru_1^0 = 0$ ) تعیین می‌گردد (شکل ۷-ا).

**گام ۲ - بررسی نمونه جدید:** اگر تمام نمونه‌ها بررسی شده باشند، فرایند خوشه‌بندی به پایان می‌رسد. در غیر این صورت، فاصله‌ی نمونه فعلی با مراکز خوشه‌های موجود محاسبه می‌شود.

**گام ۳ - تخصیص نمونه به خوشه موجود:** اگر نمونه فعلی در محدوده یکی از خوشه‌های موجود واقع شود، خوشه جدیدی ایجاد نمی‌شود و خوشه‌ها نیز بروز نمی‌شوند (مانند نقاط  $x_4$  و  $x_6$  در شکل ۷-ب و شکل ۷-ج). در این حالت الگوریتم به گام ۲ باز می‌گردد، در غیر این صورت به گام ۴ می‌رود.

**گام ۴ - مقدار  $S_{ij}$  (فاصله نقطه جدید از خوشه‌ی  $Z_j$  + شعاع خوشه‌ی  $Z_j$ )** برای تمام خوشه‌ها محاسبه می‌گردد و با حد آستانه ( $D_{thr}$ ) از پیش تعریف شده مقایسه می‌شود.

○ **ایجاد خوشه جدید:** اگر  $S_{ia} = \min_j(S_{ij}) > 2D_{thr}$  یک خوشه جدید ایجاد می‌شود (مانند نقاط  $x_3$  و  $x_8$  در شکل ۷-ب و شکل ۷-ج). سپس الگوریتم به گام ۲ باز می‌گردد.

○ **تخصیص به خوشه مناسب:** اگر  $S_{ia} = \min_j(S_{ij}) < 2D_{thr}$  باشد، با حرکت دادن مرکز خوشه‌ی  $a$ ، و افزایش مقدار شعاع آن به میزان  $S_{ia}/2$  به‌روز می‌شود. (مانند نقاط  $x_2$ ،  $x_5$ ،  $x_7$  و  $x_9$  در شکل ۷-ب و شکل ۷-ج). سپس الگوریتم به گام ۲ باز می‌گردد.

در حقیقت این مسئله‌ای، شناسایی تعداد قوانین فازی  $r$ ، پارامترهای مقدم شامل مرکز خوشه  $Cc_i = (Cc_{i,1}, Cc_{i,2}, \dots, Cc_{i,r})$  و اندازه شعاع (میزان پراکندگی)  $\sigma_i = (\sigma_{i,1}, \sigma_{i,2}, \dots, \sigma_{i,r})$  و همچنین پارامترهای تالی  $\theta_i = (\theta_{i0}, \theta_{i1}, \theta_{i2}, \dots, \theta_{in})$  سیستم فازی TS، به منظور پیش‌بینی  $\gamma$  می‌باشند. بنابراین فعالیت‌های اصلی که معمولاً محققان روی آن‌ها و چگونگی پیاده‌سازی آن‌ها تمرکز دارند، عبارت‌اند از:

- ۱) شناسایی ساختار اولیه (شناسایی تعداد خوشه‌ها و قوانین) و یادگیری پارامترهای بخش مقدم (هم در مدل آفلاین و هم در مدل آنلاین)
- ۲) تخمین پارامترهای تالی‌های وابسته (هم در مدل آفلاین و هم در مدل آنلاین)
- ۳) اصلاح ساختار و به‌روزرسانی پارامترها در صورت نیاز (فقط در مدل آنلاین)

در ادامه روش یادگیری در حال تکامل برای بخش‌های مقدم و تالی را مورد بررسی قرار می‌دهیم و جزئیات الگوریتم یادگیری GODFIS به تفصیل توضیح خواهیم داد.

### خوشه‌بندی در حال تکامل

برای طراحی سیستم استنتاج فازی، نیاز به استخراج قواعد سیستم مورد مطالعه می‌باشد. قواعد سیستم‌های فازی توسط "متخصصان حوزه" یا "روش‌های خوشه‌بندی" و یا ترکیبی از آن‌ها، استخراج می‌شوند. روش اول روشی کیفی می‌باشد، از این رو قواعد از یک متخصص به متخصص دیگر تغییر می‌کند و همچنین همیشه دسترسی به متخصصان ممکن نیست. در مقابل، روش‌های مورد استفاده در این تحقیق، روش‌های کمی هستند که بر اساس داده‌های سیستم استخراج می‌شوند و نیازی به حضور متخصص ندارند. البته، مدل GODFIS به دلیل استفاده از ویژگی‌های مدل‌سازی TS، این امکان را به متخصصان می‌دهد که قوانین را حذف، اضافه یا اصلاح کنند.

در میان الگوریتم‌های خوشه‌بندی در حال تکامل، الگوریتم ECM<sup>۵۰</sup> که توسط [۳۶] برای خوشه‌بندی تک-گذری و آنلاین ارائه شده است، به دلیل سرعت بالا، سادگی، و همچنین

<sup>50</sup> Evolving Clustering Method

### راه حل پیشنهادی:

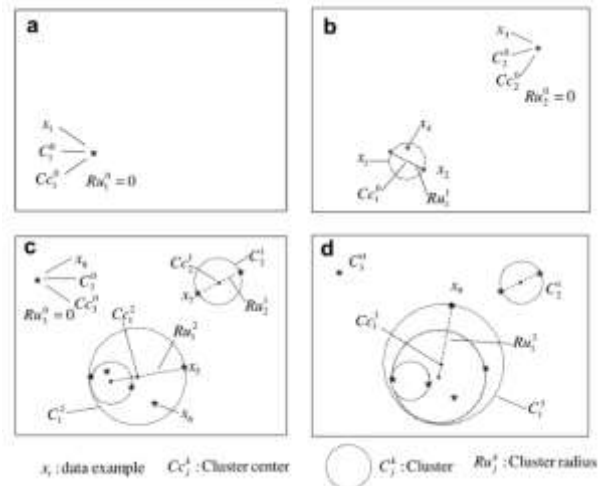
برای رفع این مشکلات، در این مقاله الگوریتم GOECA با ویژگی‌های زیر ارائه می‌شود:

- ✓ **بهینه‌سازی آنلاین:** ارائه یک روش بهینه‌سازی آنلاین با الهام از روش‌های کلاسیک خوشه‌بندی، به منظور افزایش عملکرد الگوریتم و کاهش خطای پیش‌بینی.
- ✓ **شعاع متغیر در ابعاد مختلف:** برای هر بعد شعاع متفاوتی در نظر گرفته می‌شود تا خوشه‌ها لزوماً کروی نباشند و امکان تشکیل هایپر بیضی هم وجود داشته باشد.
- ✓ **مکانیزم هرس قوانین:** ارائه یک معیار جدید برای حذف خوشه‌های منسوخ ارائه می‌شود که ضمن حفظ بهینگی باعث افزایش انعطاف‌پذیری، کاهش پیچیدگی و هزینه‌های محاسباتی، و همچنین حذف تأثیر داده‌های نویز می‌گردد.

### الگوریتم خوشه‌بندی GOECA

در اغلب الگوریتم‌های بهینه‌سازی آفلاین، مانند K-Means یا FCM، نقطه‌ای به‌عنوان مرکز خوشه در نظر گرفته می‌شود که میانگین وزنی تمام نقاط آن خوشه است. در واقع، این مرکز، نقطه‌ای است که مرکز ثقل خوشه محسوب می‌شود. البته، تعریف وزن در این الگوریتم‌ها ممکن است متفاوت باشد. در مقابل، بسیاری از الگوریتم‌های آنلاین خوشه‌بندی (مانند eTS، xTS، ePL، ePL+، ePFM) معمولاً خود نقاط داده را به‌عنوان مرکز خوشه در نظر می‌گیرند. این رویکرد با مفهوم مرکز ثقل در بهینه‌سازی متفاوت است، زیرا نقطه بهینه مرکز خوشه معمولاً دقیقاً روی داده‌های ورودی قرار ندارد. علاوه بر این، روش‌های آنلاین به‌صورت اکتشافی مراکز خوشه را به دست می‌آورند و و مرحله‌ای مجزا برای بهینه‌سازی خوشه‌ها ندارند که باعث کاهش دقت سراسری الگوریتم‌های آنلاین می‌شود.

هرچند برخی پژوهشگران سعی کرده‌اند با استفاده از روش‌های کلاسیک خوشه‌های بهینه را محاسبه کنند (مانند rFCM و rPFM)، اما از آنجاکه این الگوریتم‌ها فاقد مکانیزم تکامل هستند، در شرایط غیرثابت (پویا) عملکرد مناسبی ندارند. بنابراین، به روشی نیاز داریم که هم بتواند بهینگی مراکز خوشه‌ها را حفظ کند و هم برای محیط‌های آنلاین کاربرد داشته باشد.



شکل ۷. یک فرایند خوشه‌بندی پویا به استفاده الگوریتم ECM آنلاین در فضای ۲ بعدی

همان‌طور که مشاهده می‌شود، اگرچه الگوریتم اطلاعاتی از نمونه‌های قبلی را ذخیره نمی‌کند، اما حداکثر فاصله بین مرکز هر خوشه و دورترین نمونه متعلق به آن، از مقدار آستانه تجاوز نمی‌کند. این مقدار آستانه، پارامتری است که بر تعداد خوشه‌های تخمین زده‌شده تأثیر می‌گذارد. الگوریتم ECM سه ضعف عمده دارد:

- ✓ **عدم بهینگی:** الگوریتم ECM نیز مانند بسیاری از الگوریتم‌های خوشه‌بندی آنلاین، بهینه نمی‌باشد و این مسئله باعث کاهش عملکرد و دقت پیش‌بینی می‌شود. هرچند [۳۶] یک روش بهینه‌سازی مبتنی بر برنامه‌ریزی ریاضی (با محدودیت حداکثر فاصله) ارائه کردند؛ اما این روش به دلیل نیاز به ذخیره اطلاعات قبلی و اجرایی آفلاین، کارایی چندانی ندارد.
- ✓ **شعاع یکسان در تمام ابعاد:** در ECM شعاع گسترش هر خوشه در تمام ابعاد برابر است در نتیجه هر خوشه به شکل یک هایپر کره ایجاد می‌شود، در حالی که ممکن است شکل واقعی خوشه کروی نباشد.
- ✓ **عدم حذف خوشه‌های منسوخ:** نبود مکانیزم حذف خوشه‌های قدیمی باعث افزایش پیچیدگی و تأثیر نامطلوب داده‌های نویز بر خروجی سیستم می‌شود.

در ادامه میزان اثرگذاری نقاط بر مرکز خوشه و شعاع خوشه ارائه می‌شود.

### اصلاح مرکز خوشه به صورت بازگشتی

همان‌طور که گفته شد، در بیشتر الگوریتم‌های آفلاین، مرکز خوشه بر پایه میانگین وزنی نقاط آن خوشه محاسبه می‌شود که طبق فرمول (۷) محاسبه می‌شود.

$$CC_{ij}^k = \frac{\sum_{t=1, x^t \in C_i} x_j^t}{nPop_i^k}, \quad \forall i, j \quad (7)$$

که در آن  $nPop_i^k$  تعداد اعضای خوشه  $i$ ام در لحظه  $k$  و  $CC_{ij}^k$  مولفه  $j$ ام از مرکز خوشه  $i$ ام در لحظه  $k$  است. مشکلی که این روش دارد آن است که باید تمام نقاط خوشه در دسترس باشند، که در مسائل آنلاین و کلان‌داده‌ای ممکن است عملی نباشد. برای رفع این محدودیت، فرمول (۷) می‌بایست به صورت بازگشتی بازنویسی شود (فرمول (۸)-(۹)).

$$CC_{ij}^{k+1} = \frac{\sum_{t=1, x^t \in C_i} x_j^t}{nPop_i^{k+1}} = \frac{\sum_{t=1, x^t \in C_i} x_j^t + x_j^{k+1}}{nPop_i^k + 1} \\ = \frac{\sum_{t=1, x^t \in C_i} x_j^t}{nPop_i^k + 1} + \frac{x_j^{k+1}}{nPop_i^k + 1} \quad (8) \\ = CC_{ij}^k \times \frac{nPop_i^k}{nPop_i^k + 1} + \frac{x_j^{k+1}}{nPop_i^k + 1}$$

$$CC_{ij}^{k+1} = CC_{ij}^k \times \frac{nPop_i^k}{nPop_i^k + 1} + \frac{x_j^{k+1}}{nPop_i^k + 1} \quad (9)$$

هنگامی که خوشه تازه تشکیل می‌شود، مرکز خوشه ( $CC_{ij}^k$ ) همان اولین نقطه‌ای است که خوشه را ایجاد کرده است. از آن به بعد، هرگاه نقطه جدیدی به آن خوشه اضافه شود، مرکز خوشه طبق فرمول بازگشتی (۹) اصلاح می‌شود در غیر این صورت مرکز خوشه بدون تغییر باقی می‌ماند ( $CC_{ij}^{k+1} = CC_{ij}^k$ ). این معادله تضمین می‌کند که مرکز خوشه به مرکز ثقل واقعی آن همگرا شود و در عین حال سازگاری سیستم را با کاربرد آنلاین حفظ می‌کند.

این روش باعث می‌شود که الگوریتم از حالت بهینه خارج نشود و همواره مرکز ثقل نقاط را به عنوان مرکز خوشه در نظر بگیرد. به عبارتی دیگر، با وجود اینکه این روش مانند سایر روش‌های آنلاین یک فرایند اکتشافی است و به طور مستمر خودش را بهبود می‌دهد، ولی به صورت حدی مرکز خوشه‌ها با مرکز ثقل کل داده‌های مربوط به آن خوشه برابری می‌کند.

توابع عضویت درجه ارتباط قوانین با یک نمونه خاص را توصیف می‌کنند. در GODFIS، همانند بسیاری از پژوهش‌های مشابه در این حوزه، توابع عضویت گاوسی به کار رفته‌اند. دلیل این انتخاب نیز قابلیت تعمیم‌پذیری مناسب و پوشش کل فضای ویژگی‌هاست [۳۷، ۶۴]. تابع عضویت طبق فرمول (۵) تعریف می‌شود:

$$\mu_{ij}(x_j) = \exp \left( -\frac{(x_j - CC_{ij})^2}{2\sigma_{ij}^2} \right) \quad (5)$$

در این فرمول،  $CC_{ij}$  نشان‌دهنده مرکز خوشه،  $\sigma_{ij}$  شعاع یا منطقه نفوذ (وسعت تابع عضویت) آن خوشه است. با توجه به شباهت آن به توزیع نرمال، می‌توان شعاع را بر اساس انحراف معیار نیز بیان کرد.

### تصمیم‌گیری برای اصلاح خوشه‌ها

به روزرسانی خوشه‌ها در واقع شامل اصلاح مرکز خوشه و شعاع خوشه می‌باشد. برای سنجش تأثیر یک نقطه جدید بر پایگاه دانش، باید با استفاده از معیارهای مشخصی، اثر این نقطه را ارزیابی کرد. در الگوریتم GOECA (مانند ECA)، ساختار خوشه‌ها بر اساس معیار فاصله شکل می‌گیرد. اما تفاوت معیار فاصله در GOECA نسبت به ECA حذف اثر تعداد بعد، از معیار فاصله است که این ویژگی باعث عدم وابستگی حد آستانه به تعداد ابعاد می‌شود (فرمول (۶)).

$$D_i^k = \frac{\|CC_i^k - \mathbf{x}^k\|_d}{n}, \quad \forall i \quad (6)$$

در آن،  $\|\cdot\|_d$  نرم فاصله را نشان می‌دهد.

به طور کل برای اصلاح خوشه‌ها سه حالت وجود دارد:

- نقطه جدید درون محدوده یکی از خوشه‌ها قرار گیرد: تنها مرکز آن خوشه تغییر می‌کند و نیازی به تغییر شعاع نیست.
- نقطه جدید خارج خوشه‌ها باشد، اما به نزدیکتر خوشه تخصیص یابد: اگر  $Min(D_i^k) < D_{thr}$  باشد، هم شعاع و هم مرکز خوشه اصلاح می‌شوند.
- اگر نقطه جدید به هیچ خوشه‌ای نزدیک نباشد: اگر  $Min(D_i^k) > D_{thr}$  باشد، یک خوشه جدید تشکیل می‌شود که مرکز آن همان نقطه جدید است و شعاع اولیه آن صفر در نظر گرفته می‌شود.

## اصلاح شعاع کلاستر به صورت بازگشتی

زمانی که نقطه جدید خارج محدوده شعاع قرار گیرد و باعث به روزرسانی مرکز خوشه شود، شعاع آن نیز متناسب با شرایط جدید تغییر می کند. در مدل آفلاین ECM که در مقاله [۳۶] معرفی شده، شعاع خوشه برابر است با فاصله مرکز خوشه تا دورترین نقطه موجود در آن خوشه. در حالت آنلاین نیز، همین قاعده صادق است و طبق فرمول (۱۰) به صورت بازگشتی محاسبه می شود:

$$\begin{aligned} S_{ia} &= \min_j (S_{ij} = d_{ij} + Ru_j), \\ Ru_a &= S_{ia}/2 \end{aligned} \quad (10)$$

ضعف اصلی این فرمول در محاسبه شعاع، یکسان بودن شعاع برای تمام ابعاد و تولید خوشه های صرفاً کروی می باشد که برای داده های واقعی همیشه مناسب نیست. علاوه بر این، شعاع خوشه می بایست متناسب با روش مرکز ثقل بازگشتی به روزرسانی شود.

در روش پیشنهادی نیز فاصله مرکز خوشه تا دورترین نقطه در همان بعد به عنوان شعاع در نظر گرفته می گیرد. برای سادگی، ابتدا مرکز خوشه را طبق فرمول بازگشتی (۹) محاسبه می کنیم و سپس با توجه به قوانین بازگشتی می توان نحوه محاسبه شعاع را به صورت فرمول (۱۱) فرموله کرد. لازم به ذکر است، مقدار شعاع اولیه هر خوشه برای همه ابعاد برابر با صفر است.

$$\sigma_{aj}^{k+1} = \text{Max}(\sigma_{aj}^k, \|Cc_{aj}^{k+1} - x_j^{k+1}\|_d) \quad \forall j \quad (11)$$

این روش باعث می شود که خوشه بندی به طور پیوسته بر اساس مفهوم مرکز ثقل خود را با داده های جدید تطبیق دهد و همچنین محدودیت های روش های قبلی مانند تولید خوشه های صرفاً کروی و یا نیاز به ذخیره سازی داده های تاریخی را برطرف می کند.

## برآورد قسمت تالی مدل های در حال تکامل

به طور کلی دو روش اصلی برای محاسبه پارامترهای تالی وجود دارد: بهینه سازی محلی و بهینه سازی سراسری. بنابراین هر بار که خروجی های واقعی/هدف ارائه می شوند، پارامترهای تالی را می توان به صورت بازگشتی و به دو روش محلی یا سراسری به روزرسانی کرد. با مرور الگوریتم های پیشرفته قبلی، مشاهده می شود که پرکاربردترین روش های یادگیری تالی، یک

روش بهینه سازی محلی موسوم به wRLS است، نه نسخه سراسری آن (یعنی حداقل مربعات بازگشتی) [۱۰۳]. مقاله ای [۱۱۳]، این دو روش را در حالت آفلاین مقایسه کرده و نشان داده است که روش سراسری از دقت بالاتری نسبت به روش محلی برخوردار است. اگرچه این نتیجه در روش های آفلاین تأیید شده و انتظار می رود در روش های آنلاین نیز صدق کند، اما در عمل، در روش های آنلاین و بازگشتی این موضوع تأیید نشده است. برخی مقالات حتی گزارش داده اند که روش محلی در شرایط آنلاین دقت بیشتری نسبت به روش سراسری دارد [۶۳، ۳۹]. دلیل این امر می تواند ناشی از فرایند به روزرسانی پایگاه دانش در حالت آنلاین و روش تک-گذری باشد.

در مقاله [۱۰۳] اشاره شده است که عدم توجه به ارتباط بین یادگیری تالی و مقدم در روش های آنلاین می تواند باعث بروز پدیده «اثر بی آموزشی»<sup>۵۱</sup> گردد. این عبارت را می توان به این صورت نیز تفسیر کرد که خوشه بندی غیربهینه باعث ایجاد خطای مضاعف در پیش بینی می شود. از آنجاکه در روش های آنلاین معمولاً ابتدا بخش مقدم و سپس بخش تالی به روزرسانی می شوند، اگر تشخیص پارامترهای مقدم به درستی انجام نگیرد، این خطا به بخش تالی نیز منتقل شده و باعث تشدید خطای کلی می گردد.

تفاوت اصلی بین روش های سراسری و محلی در این است که در روش سراسری، تمام پارامترهای تالی به صورت یکپارچه به روز می شوند، در حالی که در روش محلی تنها پارامترهای مربوط به خوشه فعال تعدیل می گردد و توابع خطی به صورت مجزا و بدون در نظر گرفتن تأثیر متقابل قوانین مختلف، بهینه سازی می شوند. این ویژگی باعث می شود که در روش سراسری، خطای ناشی از تشخیص نادرست بخش مقدم، تمام ساختار مدل را تحت تأثیر قرار دهد و دقت پیش بینی را به میزان قابل توجهی کاهش دهد. بنابراین، یکی از دلایل عمده ای که روش های آنلاین ترجیح می دهند از به روزرسانی محلی برای بخش تالی استفاده کنند، همین مشکل تشخیص نادرست و غیربهینه پارامترهای مقدم است.

نتایج آزمایشات ما نشان می دهد که الگوریتم GODFIS به دلیل تشخیص دقیق خوشه ها مبتنی بر مفهوم مرکز ثقل و تلاش برای بهینگی خوشه ها، به طور معناداری با روش به روزرسانی سراسری تالی سازگاری دارد. این یافته با نتایج

<sup>51</sup> unlearning effect

$$\psi_k = [\lambda_1(x_k)x_{ek}^T, \lambda_2(x_k)x_{ek}^T, \dots, \lambda_r(x_k)x_{ek}^T]^T;$$

$$x_{ek} = [1, x_k^T]^T$$

به صورت جایگزین، فرم برداری فرمول (۱۳) را می توان به صورت فرمول (۱۴) بازنویسی کرد:

$$E_G = (Y - \Psi^T \theta)^T (Y - \Psi^T \theta) \quad (14)$$

که در آن  $Y$  یک ماتریس  $(m \times t)$ -بعدی تشکیل شده از  $y_k$  و  $\Psi$  یک ماتریس  $t \times (n+1) \times t$ -بعدی تشکیل شده از  $\psi_k^T$  است، و  $\theta$  یک ماتریس  $t \times (n+1) \times m$ -بعدی است که توسط مقادیر آنی تشکیل شد. بردار  $\theta$  ای که فرمول (۱۴) را به حداقل می رساند از شبه وارونگی محاسبه می شود (فرمول (۱۵)).

$$\theta = (\Psi^T \Psi)^{-1} \Psi^T Y \quad (15)$$

در روش بازگشتی که به صورت دوره به دوره می باشد، مقادیر ضرایب تالی را می توان از روش حداقل مربعات بازگشتی (که فیلتر کالمن نیز نامیده می شود) تخمین زد [۳۵، ۶۳]:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + C_k \psi_k (y_k - \psi_k^T \hat{\theta}_{k-1}) \quad (16)$$

$$C_k = C_{k-1} + \frac{C_{k-1} \psi_k \psi_k^T C_{k-1}}{1 + \psi_k^T C_{k-1} \psi_k}, \quad k = [1, t] \quad (17)$$

روابط (۱۶) و (۱۷)، به صورت بازگشتی و در هر دوره حساب می شوند. این الگوریتم با شرایط اولیه  $\hat{\theta}_0 = 0$  و  $C_0 = \Omega I$  آغاز می شود، که در آن  $\Omega$  یک عدد مثبت بزرگ است (معمولاً  $\Omega = 1000$ ) و  $I$  ماتریس همانی در نظر گرفته می شود.  $C_k$  یک ماتریس کوواریانس با ابعاد  $r(n+1) \times r(n+1)$  و  $\hat{\theta}_k$  برآورد ضرایب توابع با استفاده از  $k$  نمونه داده اولیه می باشد.

الگوریتم فیلتر کالمن بازگشتی می تواند به صورت آنلاین و کارآمد از داده های جدید یادگیری کند. این روش بدون نیاز به انجام محاسبات تکراری، اجرای فرایندهای بهینه سازی مجدد یا ذخیره سازی داده های تاریخی، قادر به به روزرسانی پارامترهای مدل است. اما این یادگیری با فرض ثابت بودن تعداد قوانین صورت می گیرد. با افزایش تعداد قوانین، تعداد معادلات و اندازه ماتریس کوواریانس نیز افزایش می یابد، از این رو

مقاله [۱۱۳] که برای شرایط بهینه و یادگیری آفلاین ارائه شده بود، همخوانی دارد. به عبارت دیگر، GODFIS با به کارگیری روش به روزرسانی سراسری تالی، نتایج بهتری نسبت به روش به روزرسانی محلی ارائه می دهد. اگرچه الگوریتم GODFIS اساساً برای به روزرسانی سراسری تالی (با هدف دستیابی به مدل بهینه) طراحی شده است، اما قابلیت استفاده از روش به روزرسانی محلی تالی را نیز دارا می باشد. در بخش های بعدی، هر دو مدل به روزرسانی سراسری و محلی تالی که توسط [۳۵] ارائه شده اند، مورد بررسی قرار خواهند گرفت.

این یافته ها حاکی از آن است که در صورت تشخیص صحیح خوشه ها و محاسبه دقیق پارامترهای مقدم، روش به روزرسانی سراسری می تواند عملکرد بهتری نسبت به روش محلی داشته باشد، که این موضوع اهمیت الگوریتم های خوشه بندی دقیق را در سیستم های استنتاج فازی برجسته می سازد.

برای مدل هایی با پارامترهای مقدم ثابت، مسئله تخمین پارامترهای مدل های خطی تالی را می توان به مسئله کمینه سازی مربعات خطا تبدیل کرد [۳۵]. این کار با جایگزینی عملیات جمع در رابطه (۳) با یک نمایش برداری معادل از خروجی ها ( $y$ ) طبق رابطه (۱۲) انجام می پذیرد.

$$y = \Psi^T \theta \quad (12)$$

که در آن  $\theta = [\theta_1^T, \theta_2^T, \dots, \theta_r^T]^T$  بردار متشکل از پارامترهای مدل خطی است.  $\psi = [\lambda_1 x_e^T, \lambda_2 x_e^T, \dots, \lambda_r x_e^T]^T$  بردار ورودی هایی است که با سطوح فعال ساری نرمال شده قوانین وزن دهی می شوند.

برای به روزرسانی تالی، می بایست به کمینه سازی تابع خطا پرداخت. برای یک مجموعه داده ورودی خروجی پرداخت. برای یک مجموعه داده ورودی خروجی  $(x_k^T, y_k), k = [1, t]$  بردار پارامترهای مدل خطی  $\theta$  که تابع هدف را به حداقل می رساند، از فرمول (۱۳) محاسبه می شود.

$$E_G = \sum_{k=1}^t (y_k - \psi_k^T \theta)^2 \quad (13)$$

به طوری که:

$$\theta_i = (X^T \Lambda_i X)^{-1} X^T \Lambda_i Y, \quad i = [1, r] \quad (20)$$

به صورت جایگزین، مجموعه‌ای از راه‌حل‌ها برای هر توابع هزینه  $E_{Li}$  (بردارها  $\theta_i$ ) را می‌توان به صورت بازگشتی از طریق الگوریتم RLS وزنی (wRLS) محاسبه کرد (فرمول‌های (۲۱) و (۲۲))، با شرط اولیه  $\hat{\theta}_0 = 0$  و  $c_{i0} = \Omega I$ .

$$\hat{\theta}_{ik} = \hat{\theta}_{ik-1} + c_{ik} x_{ek} \lambda_i(x_k) (y_k - x_{ek}^T \hat{\theta}_{ik-1}) \quad (21)$$

$$c_{ik} = c_{ik-1} + \frac{\lambda_i(x_k) c_{ik-1} x_{ek} x_{ek}^T c_{ik-1}}{1 + \lambda_i(x_k) x_{ek}^T c_{ik-1} x_{ek}}, k = [1, t] \quad (22)$$

همان‌طور که در روابط (۲۱) و (۲۲)، مشاهده می‌شود، رفتار الگوریتم wRLS به سطح فعال‌سازی قوانین وابسته است:

✓ زمانی که وزن فعال‌سازی نرمال شده یک قانون خاص  $i^*$  برابر با یک باشد ( $\lambda_i(x_k) = 1$ )، الگوریتم wRLS معادل RLS (فرمول‌های (۱۶) و (۱۷)) عمل می‌کند (تنها بر اساس این قانون  $i^*$  محاسبه می‌شود).

✓ برای قانونی ( $i^0$ ) که سطح فعال‌سازی نرمال شده برای یک مرحله زمانی معین برابر صفر باشد ( $\lambda_{i^0}(x_k) = 0$ )، پارامترها و ماتریس کوواریانس بدون تغییر می‌مانند.

✓ هنگامی که  $0 < \lambda_i(x_k) < 1$  به‌روزرسانی ماتریس کوواریانس و پارامترها با سطح فعال‌سازی نرمال شده وزن‌دهی می‌شوند. همچنین هم ماتریس کوواریانس و هم بردار پارامترها با توجه به میزان  $\lambda_i(x_k)$  تعدیل می‌شوند. این مکانیزم انعطاف‌پذیر باعث می‌شود الگوریتم بتواند به صورت هوشمندانه‌ای بین قوانین مختلف سیستم فازی تمایز قائل شود و به‌روزرسانی‌ها را متناسب با میزان مشارکت هر قانون در خروجی نهایی انجام دهد.

توجه به این نکته ضروری است که، کاربرد مستقیم RLS ممکن است منجر به عدم همگرایی شود زیرا الگوریتم اصلی RLS با فرض ساختار سیستم ثابت طراحی شده است [۶۳]. اما رویکرد پیشنهادی در این تحقیق دارای ویژگی‌های متمایزی است که این چالش را برطرف می‌سازد. برعکس بسیاری از روش‌های پیشین که ممکن است یک قانون فازی با قانون دیگری جایگزین شود و یا مراکز قوانین تغییر محسوسی با ورود داده جدید داشته باشند (مثلاً eTS+, xTS, ePL, ePL+، و...)، با ورود داده‌های جدید، ساختار به آرامی

تغییرات زیر را نیز می‌بایست در محاسبات فیلتر کالمن اعمال کرد:

✓ ضرایب تابع خطی قاعده جدید، از میانگین موزون ضرایب خطوط دیگر قواعد به دست می‌آید. لذا برای وزن‌دهی این ضرایب از درجه فعال‌سازی نرمال شده  $\lambda_i$  هر قاعده استفاده می‌شود و در این صورت خواهیم داشت (رابطه (۱۸)):

$$\hat{\theta}_k = \theta_i = (\hat{\theta}_{1,k-1}, \hat{\theta}_{2,k-1}, \dots, \hat{\theta}_{r,k-1}, \hat{\theta}_{r+1,k}), \quad (18)$$

$$\hat{\theta}_{r+1,k} = \sum_{i=1}^r \lambda_i \hat{\theta}_{i,k-1}$$

در این ماتریس  $\zeta_{ij}$ ، عنصر ماتریس کوواریانس  $\rho = (r^2 + 1)/r^2$  ضریب ثابت این عناصر است (برای اثبات این روابط به منبع [۳۵] مراجعه شود). در ماتریس اصلاح شده، بخشی از ماتریس که مرتبط با  $r+1$  امین قاعده می‌باشد  $n+1$  ستون‌ها و سطرهاى آخر در ماتریس همانند ماتریس کوواریانس اولیه  $C_0$ ، مقداردهی می‌شود. عدد بزرگ  $\Omega$  بر روی بخشی از قطر اصلی که از سطر و ستون  $(r(n+1) + 1) \times (r(n+1) + 1)$  شروع شده و تا سطر و ستون آخر ماتریس، قرار می‌گیرد. با این اصلاح، ماتریس کوواریانس بگونه‌ای تقریب زده می‌شود که گویی از ابتدا با  $r+1$  قاعده شکل گرفته است.

توابع هدف (۱۳)، (۱۴) در سطح سراسری بهینه هستند اما مدل‌های فرعی معنی‌دار محلی را می‌توان با استفاده از تابع هدف دارای وزن محلی یافت [۳۵، ۶۳]. یک راه حل تقریبی که تابع هزینه را به حداقل می‌رساند فرمول (۱۹) است که می‌توان با فرض استقلال و یا سطوح تعامل ضعیف بین زیرسیستم‌های خطی، تابع خطا را به مجموع توابع هزینه تبدیل کند:

$$E_L = \sum_{i=1}^r E_{Li}, \quad (19)$$

به طوری که در آن ماتریس  $X$  توسط  $x_{ek}^T$  تشکیل شده است ( $X \in R^{t \times (n+1)}$ ). ماتریس  $\Lambda_i$  یک ماتریس قطری است که  $\lambda_i(x_k)$  به عنوان عناصر آن در قطر اصلی است.

راه‌حل‌های  $\theta_i$  که مسائل حداقل مربعات وزنی بیان شده توسط توابع هدف  $E_{Li}$  را به حداقل می‌رسانند را می‌توان با اعمال یک شبه وارونگی وزنی به دست آورد (فرمول (۲۰)).

به طوری که  $\epsilon \in [0.03, 0.1]$  آستانه‌ای برای کنترل سودمندی هر خوشه است و  $r_k$  تعداد قوانین در پایگاه قانون در مرحله  $k$ ام است.

اصل سودمندی تأکید بر حفظ قوانین با کیفیت بالا و حذف قوانینی که منسوخ شده‌اند (در یک بازه‌ای استفاده شده‌اند و سپس استفاده چندان از آن‌ها نشده) دارد، اما این رویکرد ممکن است با بهینگی سراسری مدل در تضاد باشد. از طرفی ممکن است نقاط نویزی در داده‌ها، تشکیل خوشه جدید دهند و این امر نه تنها باعث پیچیدگی ساختار الگوریتم GODFIS می‌شود، بلکه باعث برآزش بیش از حد و تأثیر نامطلوب نقاط نویزی بر ساختار مدل می‌شود. به همین دلیل، برای دستیابی به بهینگی سراسری، لازم است اصل سودمندی بازنگری و اصلاح شود. به بیان دیگر برای حذف قوانین باید محتاط‌تر عمل کنیم تا از بهینگی سراسری دور نشویم. برای مقابله با نقاط نویزی، در این پژوهش "اصل حذف نويز" پیشنهاد می‌شود به این صورت که شرط جدیدی به معیار سودمندی افزوده می‌شود و قانون مطلوبیت مطابق با فرمول (۲۵) به‌عنوان اصل حذف نويز، اصلاح می‌گردد.

$$IF (U_{ik} < \epsilon) \text{ AND } \left( nPop_{ik} < \epsilon \times \frac{k}{r} \right) \quad (25)$$

$$THEN \text{ Prune } \mathcal{R}_i, \quad r_k \leftarrow r_k - 1$$

برای تشخیص اینکه یک خوشه از داده‌های نویزی تشکیل شده، باید دو شرط زیر به طور هم‌زمان برقرار باشد:

- (۱) کیفیت (مطلوبیت) خوشه پایین باشد
- (۲) تعداد داده‌هایی که یک خوشه را تشکیل می‌دهند از یک حدی  $\left( \epsilon \times \frac{k}{r} \right)$  کمتر باشد

در این رابطه،  $\frac{k}{r}$  نشان‌دهنده سهم هر خوشه از کل داده‌ها است (با فرض این که داده‌ها به طور یکنواخت بین خوشه‌ها تقسیم شوند). اگر تعداد داده‌های یک خوشه به طور قابل توجهی کمتر از این مقدار باشد، احتمال این که آن خوشه ناشی از داده‌های نویزی باشد افزایش پیدا می‌کند. مقدار ناشی از داده‌های نویزی مقدار پیشنهادی در نظر گرفته می‌شود که  $\epsilon = 0.01$  به‌عنوان مقدار  $\epsilon$  بر مفهوم سه انحراف معیار ( $3\sigma$ ) در توزیع نرمال انتخاب شده است. البته انتخاب مقدار  $\epsilon$  بر سخت‌گیری یا سهل‌گیری اصل حذف نويز تأثیر می‌گذارد؛ به این صورت که هرچه  $\epsilon$  کوچک‌تر باشد، حذف نويز سخت‌گیرانه‌تر می‌شود و ممکن است به تولید قوانین بیشتری منجر شود. در مقابل، هرچه  $\epsilon$

به نقاط بهینه و حدی خود همگرا می‌شود و این باعث می‌شود تغییرات ساختاری بسیار کمتری نسبت به نرخ نمونه‌برداری داشته باشیم و در نتیجه این امر به ما اجازه استفاده مستقیم از RLS را می‌دهد و این موضوع در نتایج (مقایسه بین به‌روزرسانی محلی و به‌روزرسانی سراسری) به طور قابل ملاحظه‌ای مشهود است.

### اندازه‌گیری کیفیت خوشه‌ها

سیستم فازی در حال تکامل دارای ساختاری پویا و باز است؛ به این معنا که پایگاه قوانین آن می‌تواند با افزودن قوانین فازی جدید گسترش یابد یا با حذف برخی قوانین کوچک‌تر شود. به عبارت دیگر، برای جلوگیری از مشکل برآزش بیش از حد<sup>۵۲</sup>، برخی روش‌های موجود در منابع علمی از تکنیک‌های هرس قوانین<sup>۵۳</sup> استفاده می‌کنند. در الگوریتم GODFIS نیز یک مکانیزم نظارت آنالین به نام اصل حذف نويز معرفی شده است که وظیفه ارزیابی کیفیت ساختار خوشه‌بندی را بر عهده دارد. این روش نظارت آنالین نسخه بهبودیافته‌ای از معیار سودمندی<sup>۵۴</sup> ارائه شده در مقاله [۶۳] محسوب می‌شود. معیار سودمندی نشانگر درجه فعال‌سازی نسبی انباشته یک قانون است (فرمول (۲۳)):

$$U_{ik} = \frac{\sum_{k=1}^t \theta_k}{k - I_i^*}, \quad k = [1, t] \quad (23)$$

در اینجا  $I_i^*$  نشان‌دهنده مرحله‌ای است که در آن یک داده ورودی به نقطه کانونی  $I_i^*$  تبدیل می‌شود، به بیان دیگر، زمان شکل‌گیری یا ایجاد آن قانون فازی را مشخص می‌کند. معیار سودمندی در واقع شاخصی برای سنجش میزان اثربخشی و مفید بودن هر قانون فازی است. هدف استفاده از معیار کیفیت حذف خوشه‌های بلااستفاده در ساختار خوشه است. به عبارت دیگر، این مکانیزم به صورت فعال قوانینی را که درجه فعال‌سازی آن‌ها پایین است و به طور مداوم کاربرد کمی دارند، شناسایی کرده و حذف می‌کند (فرمول (۲۴)).

$$IF \ U_{ik} < \epsilon \ \text{ THEN:} \quad (24)$$

$$\text{ Prune } \mathcal{R}_i, \quad r_k \leftarrow r_k - 1$$

<sup>52</sup> Overfitting

<sup>53</sup> Rules Pruning

<sup>54</sup> Utility Measure

بزرگتر باشد، حذف نویز آسان‌تر انجام می‌گیرد و تعداد بیشتری از قوانین حذف می‌شوند.  
فلوچارت عملیاتی اصل حذف نویز به صورت زیر ارائه می‌شود.

## فلوچارت عملیاتی اصل حذف نویز

### شروع

برای هر قانون فازی ( $i$ ) در زمان  $k$ :

✓ مقدار مطلوبیت ( $U_{ik}$ ) را محاسبه کنید.

✓ اگر کیفیت خوشه پایین باشد ( $U_{ik} < \epsilon$ )، و قانون کم‌اثر

باشد ( $nPop_{ik} < \epsilon \times \frac{k}{r}$ ):

▪ حذف قانون فازی  $i$  به روزرسانی تعداد قوانین

$$(r_k \leftarrow r_k - 1)$$

### پایان

\* آستانه سودمندی (تنظیم تجربی)  $\epsilon \in [0.03, 0.1]$

\* ضریب حذف خوشه‌های کم اهمیت (کم جمعیت) - مبتنی بر

$3\sigma$  توزیع نرمال ( $\epsilon = 0.01$ )

## الگوریتم GODFIS

در ادامه و با توجه به مباحث قبلی، این بخش مراحل دقیق اجرای الگوریتم GODFIS را تشریح می‌کند. الگوریتم GODFIS از ساختاری باز و انعطاف‌پذیر برخوردار است که می‌تواند با تغییر الگوی داده‌ها، مدل موجود را توسعه داده و آن را با شرایط جدید تطبیق دهد. ماهیت بازگشتی این الگوریتم نیز آن را از نظر محاسباتی کارآمد می‌سازد. قوانین هم‌زمان با خوشه‌بندی فضای ورودی توسط GOECA، تولید و بروز می‌شود. آموزش مدل با برآورد اولیه مدل یعنی استخراج  $r$  قانون اولیه و برآورد مقادیر  $\theta_k$  و  $C_k$  به صورت زیر شروع می‌شود:

### ✓ مرحله اول: آماده‌سازی اولیه

(۱) ورود داده‌های آموزشی:  $n_0$  جفت داده‌های ورودی - خروجی  $[1, n_0]$ ،  $(x_k^T, y_k)$  وارد سیستم می‌شوند.

(۲) پیش‌پردازش داده‌ها: نرمال‌سازی داده‌ها و انتخاب ویژگی‌های ورودی.

### ✓ مرحله دوم: استخراج ساختار اولیه

(۳) خوشه‌بندی آفلاین: با استفاده از GOECA آفلاین،  $r$  خوشه اول استخراج می‌شود. با خوشه بندی فضای

مسئله، پارامترهای بخش مقدم قوانین مشخص می‌شود (مراکز خوشه‌ها  $CC_j$ ، شعاع هر خوشه در هر بعد  $\sigma_{ij}$ ، تعداد اعضای هر خوشه  $nPop_j$ ).

(۴) آموزش اولیه مدل TSK: مقدار ضرایب توابع خطی (ماتریس  $\theta$ ) با استفاده از رابطه (۱۵) برای حالت به روزرسانی سراسری تالی (و به طور جایگزین با استفاده از رابطه (۲۰) برای حالت به روزرسانی محلی تالی) تعیین می‌شود.

### ✓ مرحله سوم: یادگیری آنلاین

(۵) پردازش داده جدید و اصلاح مقدم: داده جدید  $x_k$  با توجه به خوشه‌های موجود با استفاده از GOECA مورد ارزیابی قرار گرفته و پارامترهای مقدم اصلاح می‌شوند. سه حالت برای خوشه‌بندی داده جدید وجود دارد:

- داده جدید داخل شعاع یکی از خوشه‌های موجود واقع شود: فقط مرکز خوشه مورد نظر (با فرمول (۹)) اصلاح می‌شود.
- داده جدید خارج شعاع خوشه‌های موجود واقع شود و به نزدیکترین خوشه‌ها موجود تخصیص یابد ( $Min(D_i^k) < D_{thr}$ ). مرکز و شعاع خوشه با کمک روابط (۹) و (۱۱) بروز می‌شود.
- داده جدید، خوشه جدیدی را ایجاد کند ( $Min(D_i^k) > D_{thr}$ ). خوشه جدید با مرکزیت  $x_k$  ( $CC_{r+1} = x_k$ ) و شعاع  $\sigma_{i,r+1} = 0$  شکل می‌گیرد.

(۶) به روزرسانی پارامترهای توابع خطی تالی: به روزرسانی سراسری با RLS (روابط (۱۶) و (۱۷)) یا به روزرسانی محلی با WRLS (روابط (۲۱) و (۲۲)) انجام می‌شود.

### ✓ مرحله چهارم: کاهش پیچیدگی و پایداری در برابر نویز

(۷) اعمال اصل حذف نویز: خوشه‌ها با معیار (۲۵) مورد ارزیابی قرار می‌گیرند در صورت احراز شرایط، به

مرحله	هدف	شماره فرمول
حذف نویز	هرس قوانین کم کاربرد یا نویزی (ترکیب کیفیت خوشه و تعداد داده‌ها)	(۲۵)

در بخش بعد فرایند شناسایی و کنترل در الگوریتم

GODFIS ارائه می‌شود.

### نحوه شناسایی و کنترل در GODFIS

همان‌طور که گفته شد یکی از مزیت‌ها اصلی EFSها عدم نیاز به معادلات دقیق دینامیکی می‌باشد، برخلاف روش‌های سنتی مبتنی بر FPT که به معادلات دقیق دینامیکی ۶ درجه آزادی (DoF-6) مولتی روتورها وابسته هستند [۳، ۴]. الگوریتم GODFIS نیز به‌عنوان یک EFS نیازی به مدل‌های ریاضی از پیش تعریف‌شده ندارد. این الگوریتم از داده‌های ورودی حسگری (مانند سیگنال‌های شتاب‌سنج و ژيروسکوپ) برای شناسایی دینامیک‌های غیرخطی مولتی روتورها به‌صورت بلادرنگ استفاده می‌کند. با بهره‌گیری از خوشه‌بندی بازگشتی و شناسایی سراسری پارامترهای تالی، GODFIS ساختار قوانین فازی و پارامترهای کلیدی آن (تعداد قوانین، مراکز خوشه‌ها، شعاع خوشه‌ها، و ضرایب تالی) را به‌صورت تطبیقی به‌روزرسانی می‌کند. این رویکرد نه تنها پیچیدگی مدل‌سازی دینامیکی را کاهش می‌دهد، بلکه امکان تطبیق با عدم قطعیت‌ها و تغییرات محیطی را فراهم می‌کند. به عبارت دیگر، به‌جای استفاده از معادلات غیرخطی پیچیده مانند  $\ddot{x} = f(u, \phi, \theta, \psi)$  از داده‌های خام حسگرها برای تولید مدل‌های فازی استفاده می‌کند که دینامیک‌های سیستم را با دقت بالا تقریب می‌زنند. شکل ۸ بلوک دیاگرام GODFIS را نشان می‌دهد. این نمودار یک کنترلر پیشرفته مبتنی بر شبکه‌های فازی در حال تکامل (GODFIS) را نشان می‌دهد که برای کنترل یک سیستم غیرخطی مولتی روتور استفاده می‌شود. هدف اصلی این است که خروجی سیستم  $Z$ ، مثلاً ارتفاع، موقعیت یا زاویه) به دقت یک سیگنال مرجع مطلوب  $Z_{ref}$  را دنبال کند.

عنوان خوشه متشکل از داده‌های نویزی از پایگاه قوانین حذف می‌شوند.

### ✓ شرایط توقف:

اگر داده جدیدی وجود نداشت، پایان فرایند در غیر این صورت به ۵ برگردد.

به طور کلی الگوریتم GODFIS شامل ویژگی‌های کلیدی

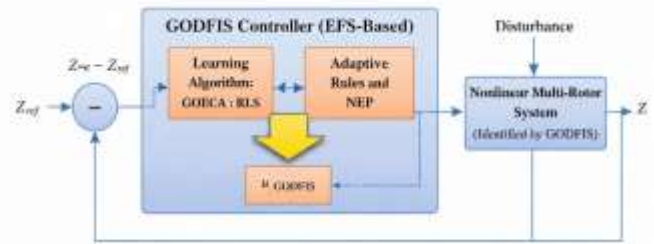
زیر می‌باشد:

- انعطاف‌پذیری ساختاری: رشد/کاهش هوشمندانه پایگاه قوانین
- کارایی محاسباتی: به‌روزرسانی بازگشتی بدون نیاز به بازآموزی کامل
- مقاومت به نویز: مکانیزم تشخیص خوشه‌های نویزی
- بهینه‌سازی مقدم و تالی: انطباق به‌روزرسانی مقدم و تالی با هدف بهینه‌سازی سراسری

این ویژگی‌ها به این الگوریتم کمک می‌کند تا این الگوریتم عملکرد قابل توجهی نسبت به الگوریتم‌های موجود داشته باشد. خلاصه مراحل الگوریتم GODFIS و فرمول‌های مرتبط در جدول ۳ ارائه شده است.

### جدول ۳. خلاصه مراحل الگوریتم GODFIS و فرمول‌های مربوطه

مرحله الگوریتم	هدف	شماره فرمول
تابع عضویت	توابع عضویت گاوسی (تعمیم‌پذیری مناسب و مقدم پوشش کل فضای ویژگی‌ها)	(۵)
معیار فاصله	مقایسه داده جدید با خوشه‌ها	(۶)
خوشه‌بندی	تعیین بهینه مرکز خوشه (به‌روزرسانی بر اساس مرکز ثقل بازگشتی)	(۹)
مقدم	تعیین شعاع خوشه در هر بعد (رفع محدودیت کروی بودن)	(۱۱)
مقادیر اولیه	تعیین اولیه ضرایب تالی به روش سراسری	(۱۵)
ضرایب تالی	تعیین اولیه ضرایب تالی به روش محلی	(۲۰)
به‌روزرسانی بازگشتی	به‌روزرسانی هم‌زمان ضرایب تالی (به روش سراسری)	(۱۶) و (۱۷)
ضرایب تالی	به‌روزرسانی مستقل ضرایب تالی برای هر قانون (به روش محلی)	(۲۱) و (۲۲)



شکل ۸. بلوک دیاگرام الگوریتم GODFIS

✓ خروجی GODFIS: سیگنال کنترلی نهایی که از ترکیب الگوریتم یادگیری و قوانین تطبیقی تولید می‌شود.

#### ۴. سیستم مولتی روتور غیرخطی: این بخش همان

سیستم فیزیکی تحت کنترل (مولتی روتور) است. این بخش توسط GODFIS از قبل شناسایی و مدل‌سازی شده است (یعنی GODFIS یک مدل دینامیکی از رفتار سیستم ساخته است که به کنترلر کمک می‌کند تا پیش‌بینی بهتری از عکس‌العمل سیستم داشته باشد). سیگنال کنترلی (u\_GODFIS) به این بخش وارد می‌شود. این سیستم به دلیل رفتار غیرخطی و پیچیده با کنترلرهای خطی ساده به خوبی کنترل نمی‌شوند.

#### ۵. اختلال<sup>۵۵</sup>: اختلال یا تغییرات ناگهانی به هر عاملی گفته

می‌شود که می‌تواند عملکرد سیستم را از حالت ایده‌آل یا مورد انتظار خارج کند. این اختلالات می‌توانند شامل نویز، تغییرات ناگهانی در محیط (مثل وزش باد ناگهانی)، ورودی‌های ناخواسته، یا تغییرات در دینامیک سیستم باشند. در نمودار ارائه‌شده، اختلال به‌عنوان سیگنالی سیستم فیزیکی مولتی روتور را تحت تأثیر قرار می‌دهد. هدف سیستم کنترلی، مانند کنترلر GODFIS، کاهش اثرات این اختلالات و حفظ پایداری و دقت عملکرد است.

#### ۶. خروجی سیستم (Z) و حلقه بازخورد: این خروجی

واقعی سیستم (مثلاً ارتفاع یا موقعیت فعلی مولتی روتور) است که توسط سنسورها اندازه‌گیری می‌شود و برای محاسبه خطا و همچنین الگوریتم یادگیری، فیدبک (بازخورد) می‌شود و حلقه کنترلی را می‌بندد.

این نمودار یک سیستم کنترلی با بازخورد را نشان می‌دهد که در آن کنترلر GODFIS با استفاده از یادگیری آنلاین و بهینه‌سازی، پویایی‌های سیستم‌های پیچیده و غیرخطی سیستم مولتی روتور را مدیریت می‌کند. این سیستم نقاط قوت قابل توجهی دارد که در ادامه لیست می‌شوند:

#### ۱. توانایی بالا در مدل‌سازی و کنترل سیستم‌های غیرخطی

پهچیده: ترکیب شبکه عصبی و منطق فازی (ANFIS) به سیستم این قدرت را می‌دهد که هرگونه غیرخطی بودن، نامعلومی و پویایی پیچیده در یک سیستم مولتی

#### ۱. سیگنال ورودی (سیگنال مرجع $Z_{ref}$ ): فرایند با ورود

یک سیگنال ورودی آغاز می‌شود. این سیگنال ورودی مطلوب یا دستور مورد نظر برای سیستم است. برای مثال، اگر ارتفاع پهباد باشد،  $Z_{ref}$  می‌تواند یک ارتفاع مشخص یا یک مسیر پروازی برنامه‌ریزی شده باشد.

#### ۲. محاسبه خطا ( $e = Z - Z_{ref}$ ): ورودی با خروجی

سیستم مقایسه شده و خطا محاسبه می‌شود. این خطا به کنترلر GODFIS منتقل می‌شود. این خطا معیار اصلی برای کنترلر است تا عملکرد سیستم را اصلاح کند.

#### ۳. کنترلر GODFIS (مبتنی بر EFS): این واحد هسته

اصلی سیستم کنترل است. این بخش، خطای ورودی ( $e$ ) را می‌گیرد و بر اساس قواعد فازی که در خود دارد، یک سیگنال کنترلی مناسب تولید می‌کند تا سیستم را هدایت کند. این بخش با دریافت خطا، عملکرد کنترلر را تحلیل کرده و قوانین و پارامترهای آن را طوری به روز می‌کند که خطا در آینده به حداقل برسد. این واحد شامل سه بخش اصلی است:

#### ✓ الگوریتم یادگیری GOCEA+RLS: این بخش

ترکیبی از الگوریتم در حال تکاملی مبتنی بر مرکز ثقل (GOCEA) و روش کمترین مربعات بازگشتی (RLS) است که برای یادگیری پارامترها استفاده می‌شود.

#### ✓ قوانین تطبیقی و اصل حذف نویز: این قوانین بر

اساس نیازهای سیستم به‌صورت پویا تنظیم می‌شوند. قوانین در این بخش بسته به شرایط می‌توانند به‌روزرسانی شوند، رشد کنند و یا با اصل حذف نویز هرس شوند. به عبارت دیگر، در این سیستم، ساختار و پارامترها (مانند قواعد فازی) به مرور زمان و با دریافت داده‌های جدید می‌تواند تکامل یابد.

این مزایای قابل توجه باعث می‌شود GODFIS الگوریتم قدرتمندی برای شناسایی و کنترل دینامیک مولتی روتورها به شمار رود.

در بخش بعد نتایج محاسباتی الگوریتم پیشنهادی با سایر الگوریتم‌های موجود در ادبیات مورد ارزیابی قرار می‌گیرد.

### نتایج محاسباتی

در این بخش، اعتبارسنجی الگوریتم GODFIS مورد ارزیابی قرار می‌گیرد. این ارزیابی شامل سه زیربخش اصلی است:

۱. بررسی عملکرد الگوریتم به صورت عمومی: در این زیربخش، عملکرد الگوریتم پیشنهادی با روش‌های پیشین بر روی داده‌های معیار مقایسه می‌شود تا توانایی GODFIS در ارائه پیش‌بینی‌های دقیق مورد بررسی قرار گیرد. برای این منظور، از سری‌های زمانی آشفته مکی-گلاس (MG)<sup>۵۸</sup> [۱۱۴] و مجموعه داده Delta Ailerons [۱۱۵] استفاده می‌شود.
۲. مقایسه به روزرسانی محلی و سراسری پارامترهای تالی: در این زیربخش، دو رویکرد موجود در به روزرسانی پارامترهای تالی — یعنی به روزرسانی محلی و سراسری — با یکدیگر مقایسه می‌شوند تا تأثیر هر یک بر عملکرد کلی سیستم GODFIS مشخص گردد.
۳. بررسی عملکرد الگوریتم کاربردهای شناسایی و کنترل مولتی روتورها: این زیربخش به ارزیابی کارایی GODFIS در حوزه کنترل اختصاص دارد. هدف، بررسی میزان توانایی الگوریتم در پاسخ‌گویی به نیازهای عملی و مقایسه آن با سایر الگوریتم‌های EFS تعریف‌شده در حوزه کنترل است. در این راستا، از مثال کنترل غیرخطی (NLC)<sup>۵۹</sup> [۴۵] و مجموعه داده PX4 [۱۱۶] بهره گرفته می‌شود.
۴. بررسی عملکرد اصل حذف نویز: این زیربخش به ارزیابی کارایی اصل حذف نویز (NEP) می‌پردازیم. هدف، بررسی میزان توانایی اصل حذف نویز در مواجهه با نویز و تأثیر عملکرد آن بر خروجی الگوریتم GODFIS می‌باشد.

روتور (مانند اثرات آیرودینامیکی، تلاطم باد، کوپلینگ بین محورها) را یاد بگیرد و بر آن غلبه کند. کنترلرهای خطی کلاسیک در برابر چنین سیستم‌هایی بسیار ضعیف عمل می‌کنند.

۲. سازگاری و یادگیری برخط (Online Adaptation & Learning): این سیستم یک کنترلر تطبیقی است. یعنی نه تنها برای یک شرایط خاص، بلکه می‌تواند خود را با تغییرات محیطی (مانند وزش باد ناگهانی) یا تغییر در دینامیک خود سیستم (مثلاً شکستگی یک پره) در حین کار تطبیق دهد. این ویژگی برای کاربردهای واقعی مولتی روتورها حیاتی است.
۳. عدم نیاز به مدل ریاضی دقیق از سیستم (Model-Free Approach): یکی از بزرگ‌ترین مزایا این است که برای طراحی این کنترلر، نیاز به یک مدل دقیق و بسیار پیچیده از مولتی روتور نیست. سیستم خودش در حین کار و با داده‌های ورودی-خروجی، یک مدل شناسایی می‌کند (همان بخش Identified by GODFIS). این فرایند طراحی کنترلر را بسیار ساده‌تر می‌کند.
۴. یادگیری پیشرفته: ترکیب الگوریتم GOCEA و RLS امکان بهینه‌سازی مداوم و یادگیری خودکار پارامترها را فراهم می‌کند، که دقت و کارایی کنترل را افزایش می‌دهد. الگوریتم GOECA به سیستم اجازه می‌دهد نه تنها پارامترها، بلکه ساختار خود را نیز تغییر دهد (مثلاً قواعد فازی جدیدی ایجاد یا قواعد بی‌فایده را حذف کند). این یعنی سیستم می‌تواند پیچیدگی خود را بر اساس پیچیدگی مسئله تنظیم کند و از ساده‌نگاری<sup>۵۶</sup> یا پیچیده‌نگاری<sup>۵۷</sup> بپرهیزد.
۵. همگرایی سریع و پایداری (با RLS): الگوریتم RLS یک روش شناخته شده و بسیار کارآمد برای تنظیم پارامترها است که به صورت بازگشتی و با سرعت نسبتاً بالایی همگرا می‌شود. این به سیستم کمک می‌کند تا به سرعت به تغییرات پاسخ دهد.
۶. مقاومت در برابر عدم قطعیت‌ها: هسته فازی سیستم به آن قابلیت مواجهه با داده‌های نادقیق، و کیفی می‌دهد.

<sup>۵۶</sup> Underfitting<sup>۵۷</sup> Overfitting<sup>۵۸</sup> Mackey-Glass<sup>۵۹</sup> Nonlinear Control

تولید می‌شوند. مدل پیش‌بینی به صورت زیر تعریف شده است (فرمول (۲۹)).

$$\hat{x}(t + 85) = f(x(t), x(t - 6), x(t - 12), x(t - 18)) \quad (29)$$

برای مقایسه عملکرد پیش‌بینی مدل‌ها از شاخص‌های NDEI، تعداد قوانین فازی (#(Rule)) و زمان آموزش ( $t_{exe}$ ) استفاده شده است. نتیجه تست مدل پیشنهادی با نتایج مدل‌های پیشین در جدول ۴ آورده شده است (از آنجایی که عملکرد سیستم‌های فازی در حال تکامل معمولاً به تنظیمات پارامتر حساس هستند، برای مقایسه منصفانه، نتایج مستقیماً از ادبیات به دست می‌آیند [۱۰۳، ۸۴-۸۶، ۳۷])

جدول ۴. مقایسه‌ی عملکرد GODFIS با سایر روش‌ها روی داده MG

Model	Model Type	NDEI	#(Rule)	$t_{exe}$
GODFIS	T-S	۰.۱۰۶	۴۲	۹.۳
LEOA [۱۰۳]	T-S	۰.۲۴۸	۴۲	۱۴۴.۸
ALMMo [۱۰۸]	T-S	۰.۴۰۲	۹	۰.۵
PSO-ALMMo [۴۹]	T-S	۰.۱۹۱	۸	۳۱۴.۳
CEFNS [۸۳]	T-S	۰.۲۶۴	۵	۰.۴
DENFIS [۳۶]	T-S	۰.۲۷۸	۵۸	۰.۶۲
EFuMo [۱۱۷]	T-S	۰.۱۳۹	۴۱	—
EFS-SLAT [۸۷]	T-S	۰.۱۱۴	۸	—
ESAFIS [۷۰]	T-S	۰.۲۹۶	۶	۵.۸
eTS [۳۵]	T-S	۰.۳۸۱	۹	۳.۱
eTS+ [۶۳]	T-S	۰.۳۹۲	۱۰	—
Simpl_eTS [۶۱]	T-S	۰.۳۹۴	۱۱	۴.۴
Simpl_eTS+ [۴۰]	T-S	۰.۳۷۵	۱۵	—
GENEFIS [۷۲]	T-S	۰.۲۸	۱۹	۴.۵
GSETSK [۷۳]	T-S	۰.۳۴۷	۱۹	—
McFIS [۶۸]	T-S	۰.۲۵	۱۰	۸.۲
PANFIS [۷۱]	T-S	۰.۳۰۱	۱۹	۴.۵
RMCEFS [۸۴]	T-S	۰.۱۱۷	۵	۰.۳
SAFIS [۶۹]	RBF	۰.۳۷۶	۶	—
SEFS [۸۶]	T-S	۰.۱۲۹	۴	۰.۴
eT2RFNN [۸۲]	T-S	۰.۳۲	۳	—
SPATFIS [۸۵]	T-S	۰.۲۷۹	۳۰	—
DeTS [۶۵]	T-S	۰.۴۴	۳	—

<sup>۶۲</sup> مقادیری که با "-" گزارش شده در منبع ذکر نشده است.

برای ارزیابی عملکرد الگوریتم GODFIS در بخش‌های فوق، از معیارهای ریشه میانگین مربعات خطا (RMSE)<sup>۶۰</sup> (فرمول (۲۶))، شاخص خطای بی‌بعد (NDEI)<sup>۶۱</sup> (فرمول (۲۷))، تعداد قوانین تولیدشده و زمان محاسباتی ( $t_{exe}$ ) استفاده می‌شود.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (26)$$

$$NDEI = \frac{RMSE}{std(y)} \quad (27)$$

### اعتبار سنجی مدل پیشنهادی به وسیله سری زمانی مکی گلاس

در این بخش عملکرد الگوریتم GODFIS نسبت به الگوریتم‌های پیشین روی داده‌های MG [۱۱۴] مورد ارزیابی قرار می‌گیرد که یک مسئله پیش‌بینی سری‌های زمانی آشفته می‌باشد. مسئله MG یکی از مسائل کلاسیک و پرکاربرد در حوزه پیش‌بینی سری‌های زمانی است و به‌عنوان یک مرجع استاندارد برای ارزیابی مدل‌های پیش‌بینی مورد استفاده قرار می‌گیرد. در مطالعات مختلفی از این مجموعه‌داده برای ارزیابی عملکرد سیستم‌های فازی در حال تکامل استفاده شده است [۱۰۳، ۳۷].

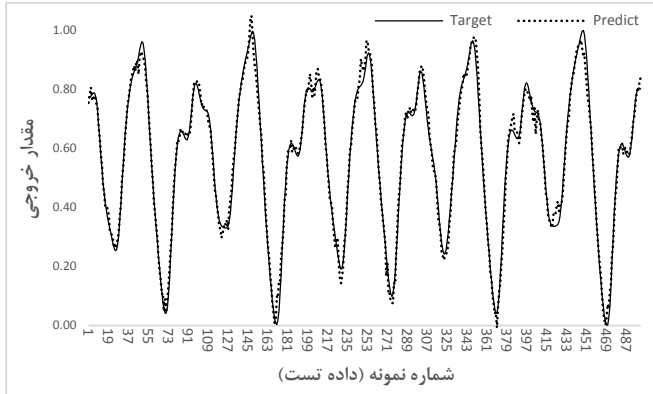
سری زمانی MG از حل معادله دیفرانسیل تأخیری زیر به دست می‌آید (رابطه (۲۸)).

$$\frac{dx(t)}{dt} = \frac{ax(t - \tau)}{1 + x^{10}(t - \tau)} - bx(t) \quad (28)$$

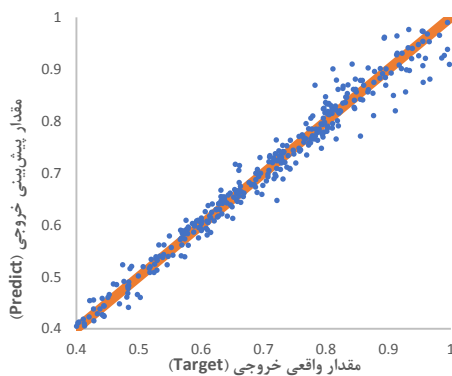
که در آن  $a = 0.2$ ،  $b = 0.1$ ،  $\tau \geq 17$ . در این پژوهش، مطابق روال رایج در تحقیقات پیشین، مقدار  $\tau = 17$  در نظر گرفته شده است. ۳۵۰۰ جفت ورودی-خروجی با استفاده از روش عددی Runge-Kutta مرتبه چهارم و با مقدار اولیه  $x(0) = 1.2$  تولید شده‌اند. ۳۰۰۰ نمونه آموزشی اول در محدوده  $t = 201$  تا  $t = 3200$  و نمونه‌های آزمایشی باقی مانده از  $t = 5001$  تا  $t = 5500$

<sup>۶۰</sup> Root Mean Square Error

<sup>۶۱</sup> Non-Dimensional Error Index



شکل ۹. مقایسه بین مقدار پیش‌بینی و مقدار واقعی برای 500 داده اول داده‌های تست MG



شکل ۱۰. مقایسه بین مقدار پیش‌بینی و واقعی MG روی خط ۴۵ درجه

شکل ۱۱ رد خطای پیش‌بینی الگوریتم GODFIS را در طول اجرای آن روی داده‌های تست نشان می‌دهد. همان‌طور که مشاهده می‌شود، قدر مطلق خطای پیش‌بینی از ۰,۰۹ تجاوز نمی‌کند. علاوه بر این، در اغلب نقاط تست، خطای پیش‌بینی کمتر از ۰,۰۶ و حتی در بسیاری از موارد کمتر از ۰,۰۴ است. این نتایج عملکرد GODFIS را به مراتب برتر از الگوریتم LEOA نشان می‌دهد، چراکه در گزارش‌های مربوط به LEOA (ارائه شده در [۱۰۳])، قدر مطلق خطا تا حدود ۰,۳ گزارش شده است.

شکل ۱۲ هیستوگرام خطاهای پیش‌بینی برای نقاط تست نمایش داده شده است. توزیع این خطاها نشان می‌دهد که الگوی خطا به شکل تقریباً نرمال است، که این موضوع نیز از ویژگی‌های مثبت مدل محسوب می‌شود. چراکه پراکندگی نرمال خطاها نشان می‌دهد که مدل فاقد سوگیری سیستماتیک است و خطاهای آن به صورت تصادفی حول مقدار واقعی

همان‌طور که در جدول ۴ نیز مشاهده می‌شود، GODFIS در مقایسه با دیگر مدل‌ها بهترین عملکرد را به لحاظ شاخص NDEI ارائه داده است. این موضوع نشان می‌دهد که GODFIS در پیش‌بینی دقیق داده‌های سری زمانی آشفته بسیار موفق عمل می‌کند. با این حال، GODFIS نسبت به برخی از الگوریتم‌های دیگر، قوانین فازی بیشتری را برای پیش‌بینی تولید می‌کند و زمان اجرای طولانی‌تر دارد، هرچند این زمان همچنان در سطح قابل قبولی قرار دارد. این رفتار ناشی از طراحی و ساختار الگوریتم GODFIS است؛ چراکه این الگوریتم به گونه‌ای طراحی شده تا همواره سعی کند اطلاعات تاریخی بیشتری را حفظ کند و به این ترتیب، بهینگی سراسری مدل و در نتیجه دقت بالا را تضمین نماید.

با در نظر گرفتن اینکه تنها مدل آنلاینی که روی بهینگی سراسری EFSها تمرکز داشته، مدل LEOA است، می‌توان گفت GODFIS به مراتب عملکرد بهتری نسبت به LEOA دارد. برخلاف LEOA، الگوریتم GODFIS توانسته است با حفظ تعداد قوانین مشابه، هم از نظر دقت (NDEI) و هم از نظر زمان اجرا عملکرد بهتری ارائه دهد.

علاوه بر LEOA، الگوریتم PSO-ALMMo دیگر تحقیقی است که روی بهینگی سراسری EFS انجام شده، اگرچه این الگوریتم، آنلاین نیست و با روش متاهوریستیک به بهینه‌سازی سراسری پرداخته است، با این حال، GODFIS توانسته از نظر دقت پیش‌بینی (NDEI) و زمان محاسباتی عملکرد بهتری نسبت به PSO-ALMMo ارائه دهد، اگرچه در مقایسه با آن، تعداد قوانین فازی بیشتری تولید می‌کند.

شکل ۹ و شکل ۱۰ مقایسه‌ای بین مقدار پیش‌بینی شده داده‌های تست و مقدار واقعی آن‌ها ارائه می‌دهد. شکل ۹ مقایسه‌ای بین مقدار واقعی ۵۰۰ نقطه داده تست و پیش‌بینی‌های مدل را ارائه می‌دهد، همان‌طور که مشاهده می‌شود به جز تعداد معدودی از نقاط، GODFIS موفق شده داده‌ها را با دقت بسیار بالا پیش‌بینی کند. در شکل ۱۰ این مقایسه به صورت نمودار پراکندگی روی خط ۴۵ درجه ارائه شده است (بدیهی است که هرچه فاصله نقاط از خط ۴۵ درجه کمتر باشد مدل پیش‌بینی دقیق‌تری را ارائه می‌دهد). همان‌طور که مشاهده می‌شود، این شکل بیانگر عملکرد دقیق و قابل اطمینان مدل GODFIS در پیش‌بینی داده‌های MG است.

پراکنده شده‌اند، موضوعی که نشان‌دهنده پایداری و دقت بالا در پیش‌بینی‌های GODFIS می‌باشد.

۸۴، ۸۷] مقایسه شده است (از منظر معیار RMSE، تعداد قانون و زمان محاسباتی).

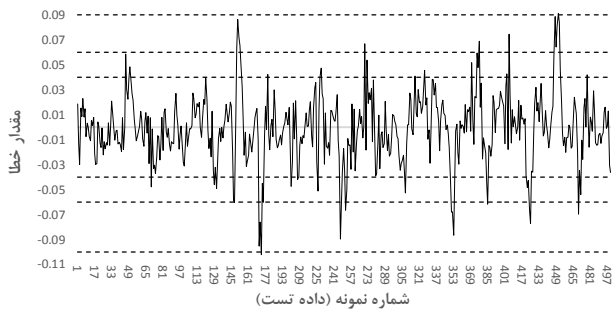
جدول ۵. مقایسه‌ی GODFIS نسبت به روش‌های دیگر در ادبیات

روی مجموعه داده Delta Ailerons

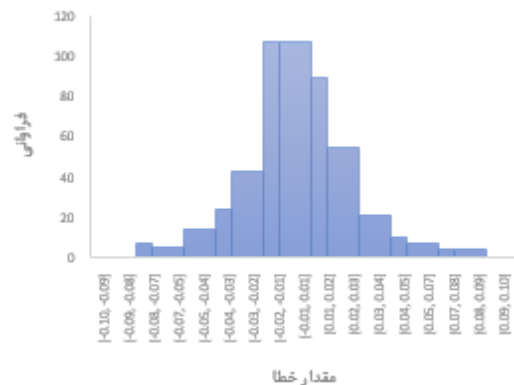
Model	Model Type	RMSE	#(Rule)	$t_{exe}$
GODFIS	T-S	۰.۰۴۴۹	۲	۱۱.۷
ALMMo [۱۰۸]	T-S	۰.۰۵۱۳	۱۰	۴
CEFNS [۸۳]	T-S	۰.۰۵۰۲	۳	۳
DENFIS [۳۶]	T-S	۰.۰۴۹۷	۱۱	-
EFS-SLAT [۸۷]	T-S	۰.۰۴۱۲	۸	-
ESAFIS [۷۰]	T-S	۰.۰۵۰۶	۱۳	۱۲.۴
eTS [۳۵]	T-S	۰.۰۵۱۳	۴	۲.۱
McFIS [۶۸]	T-S	۰.۰۵۰۹	۱۵	-
RMCEFS [۸۴]	T-S	۰.۰۴۹۸	۲	۰.۲
SAFIS [۶۹]	RBF	۰.۰۵۴۹	۱۴	۶.۸
SEFS [۸۶]	T-S	۰.۰۴۷۶	۷	۰.۴
Simpl_eTS [۶۱]	T-S	۰.۰۵۱۲	۴	۲

همان‌طور که در جدول ۵ مشاهده می‌شود، الگوریتم GODFIS در مقایسه با سایر روش‌های موجود عملکرد قابل قبولی از خود نشان داده است. به طور مشخص، از نظر دقت پیش‌بینی (شاخص RMSE)، GODFIS در جایگاه دوم پس از الگوریتم EFS-SLAT قرار دارد و به لحاظ تعداد قوانین جایگاه اول (مشترکاً با الگوریتم RMCEFS) را دارد. این نتایج نشان می‌دهند که GODFIS با وجود ساختار پویا و یادگیری آنلاین، توانسته است هم مدلی فشرده و ساده ایجاد کند و هم دقت مناسبی ارائه دهد. البته، به دلیل ساختار ویژه GODFIS که بر دستیابی به بهینگی سراسری تمرکز دارد، زمان اجرای بیشتری نسبت به برخی روش‌ها دارد که این افزایش زمان کاملاً قابل انتظار و توجیه‌پذیر است.

از سوی دیگر، الگوریتم PSO-ALMMo که نسخه‌ای تکاملی از ALMMo محسوب می‌شود و محدود به سناریوهای آنلاین است، روی بهینگی سراسری EFS تحقیق کرده است (نتایج این الگوریتم‌ها از [۴۹] اخذ شده است). اگرچه الگوریتم PSO-ALMMo مدل آنلاینی نیست، با این حال کیفیت جواب GODFIS نسبت به الگوریتم PSO-ALMMo، عملکرد قابل دفاعی دارد (جدول ۶). همان‌طور که در جدول ۶



شکل ۱۱. رد خطا پیش‌بینی بروی داده‌های تست MG



شکل ۱۲. نمودار هیستوگرام خطاها روی داده‌های تست MG

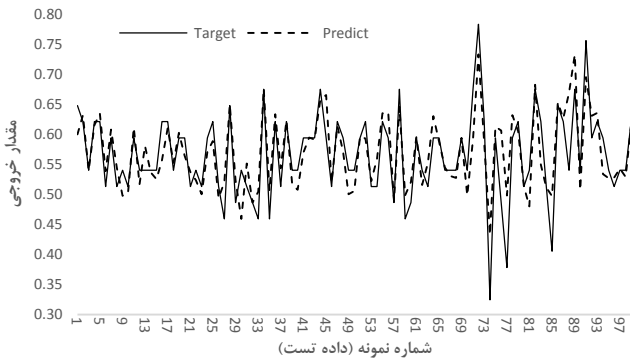
### اعتبار سنجی مدل پیشنهادی روی مجموعه داده Delta Ailerons

برای ارزیابی دقیق‌تر عملکرد الگوریتم GODFIS، از مجموعه‌داده Delta Ailerons استفاده شده است. این مجموعه داده از پایگاه داده KEEL<sup>۶۳</sup> استخراج شده است و مربوط به سیستم کنترل سطوح پروازی هواپیمای F16 می‌باشد [۱۱۵]. مدل با استفاده از پنج متغیر ورودی شامل نرخ چرخش (RollRate)، نرخ پیچش (PitchRate)، پیچ فعلی (currPitch)، چرخش فعلی (currRoll)، و تفاوت نرخ چرخش (diffRollRate) به پیش‌بینی تغییرات سطح کنترل هواپیما (Sa) می‌پردازد.

در جدول ۵ عملکرد GODFIS با سایر EFS گزارش شده در ادبیات روی در مجموعه داده Delta Ailerons [۳۷، ۴۹،

<sup>63</sup> Available at: <https://sci2s.ugr.es/keel/datasets.php>

شکل ۱۳. مقایسه مقدار پیش‌بینی و واقعی داده‌های تست Delta Ailerons روی خط ۴۵ درجه



شکل ۱۴. مقایسه بین مقدار پیش‌بینی و مقدار واقعی برای ۱۰۰ داده اول داده‌های تست Delta Ailerons

در شکل ۱۵، رد خطای پیش‌بینی الگوریتم GODFIS در طول اجرای آن روی ۱۰۰ نقطه‌ی اول از داده‌های تست نمایش داده شده است. همان‌طور که مشاهده می‌شود، قدر مطلق خطای پیش‌بینی در هیچ‌کدام از نقاط از مقدار ۰٫۱۳ فراتر نرفته است. علاوه بر این، در بیشتر نقاط، خطا کمتر از ۰٫۰۶ و حتی در بخش قابل توجهی از نقاط، کمتر از ۰٫۰۴ است. این امر بیانگر پایداری و دقت بالای الگوریتم GODFIS در فرایند پیش‌بینی است، خصوصاً در مواجهه با داده‌های واقعی و در شرایط عملیاتی. شکل ۱۶ نمودار هیستوگرام خطاهای پیش‌بینی نقاط تست را نمایش می‌دهد و همان‌طور که می‌بینید توزیع خطاها تقریباً نرمال است. در مجموع، شکل ۱۵ و شکل ۱۶ به خوبی تأیید می‌کنند که الگوریتم GODFIS علاوه بر دقت، از نظر پراکندگی خطا و کیفیت آماری پیش‌بینی‌ها نیز قابل اطمینان است.



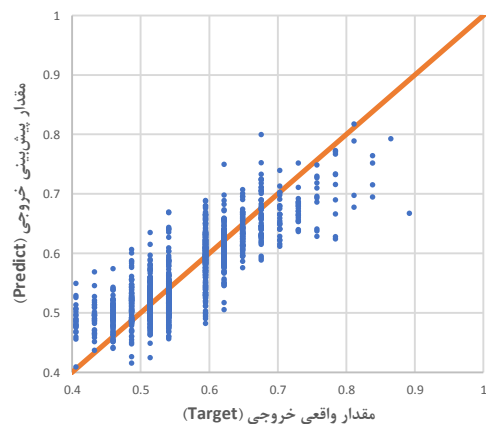
شکل ۱۵. رد خطا پیش‌بینی روی ۱۰۰ داده تست اول Delta Aileron

می‌بینید، اگرچه به لحاظ معیار NDEI، الگوریتم PSO-ALMMo کیفیت بهتری دارد و GODFIS در رتبه دوم قرار دارد، ولی به لحاظ تعداد قانونی تولید شده، GODFIS عملکرد چشمگیر نسبت به دو الگوریتم دیگر دارد و ساختاری بسیار فشرده‌تری ارائه می‌دهد. همچنین از نظر زمان محاسباتی GODFIS ( $t_{exe}$ ) به مراتب سریع‌تر و کارآمدتر از PSO-ALMMo عمل می‌کند. این نتایج نشان می‌دهد که GODFIS توانسته است با حفظ دقت مناسب، تعادلی بین سادگی ساختار، سرعت اجرا و کیفیت مدل برقرار کند، و در عین حال ویژگی مهم یادگیری آنلاین و سازگاری لحظه‌ای با تغییرات داده را نیز حفظ کند (مزیتی که الگوریتم‌های آفلاین مانند PSO-ALMMo از آن بی‌بهره‌اند).

جدول ۶. مقایسه‌ی GODFIS، ALMMo، و PSO-ALMMo روی

مجموعه داده Delta Ailerons				
Model	Model Type	NDEI	#(Rule)	$t_{exe}$
GODFIS	T-S	۰.۵۴۳۷	۲	۱۱.۷
ALMMo [۱۰۸]	T-S	۰.۵۴۹۹	۱۰	۰.۵
PSO-ALMMo [۴۹]	T-S	۰.۵۳۵۱	۱۰	۳۶۲.۴

در شکل ۱۳ مقایسه‌ای بین مقادیر پیش‌بینی شده و مقادیر واقعی داده‌های تست ارائه شده است که در آن، نقاط روی خط ۴۵ درجه ترسیم شده‌اند. شکل ۱۴ نیز مقایسه‌ای بین مقدار پیش‌بینی شده داده‌های تست و مقدار واقعی آن‌ها روی محور  $y$  ارائه می‌دهد. برای وضوح بیشتر ۱۰۰ نقطه اول داده‌های تست انتخاب شده‌اند، همان‌طور که مشاهده می‌شود به جز تعداد معدودی از نقاط، الگوریتم با دقت بالایی نقاط را پیش‌بینی کرده است.



همان‌طور که در جدول ۷ مشاهده می‌شود، تعداد قوانین تولید شده در هر دو روش برابر است و این امر به دلیل اینکه مراحل تشخیص مقدم در هر دو روش یکسان است امری بدیهی است. در شاخص‌های RMSE و NDEI، الگوریتم GODFIS به دلیل تشخیص درست و بهینه‌ی پارامترهای مقدم (با استفاده از مفهوم مرکز ثقل) در روش به‌روزرسانی سراسری تالی، قادر به پیش‌بینی‌های دقیق‌تری نسبت به روش محلی است که این امر با یافته‌های مقاله [۱۱۳] (که برای شرایط بهینه و آفلاین است) مطابقت دارد. بنابراین، در آن پژوهش نیز ثابت شده که روش‌های سراسری در مقایسه با روش‌های محلی، به شرط اینکه بخش مقدم به درستی تشخیص داده شود، از دقت پیش‌بینی بالاتری برخوردارند.

به طور کل می‌توان این‌طور نتیجه‌گیری کرد که برتری عملکرد الگوریتم GODFIS حاصل ترکیب موفقیت آمیز دو ویژگی کلیدی این الگوریتم است:

- الگوریتم خوشه‌بندی تکاملی بهینه‌شده سراسری (GOECA)

- مکانیزم به‌روزرسانی سراسری پارامترها

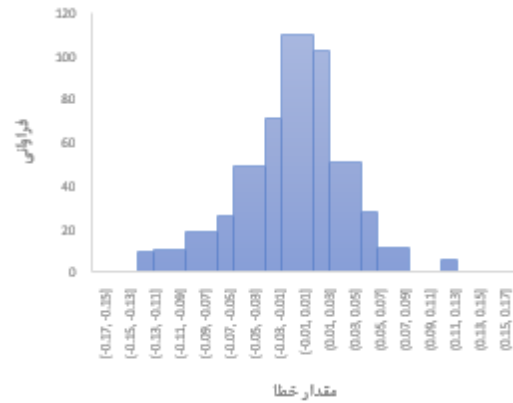
این ترکیب هوشمندانه منجر به دستیابی به نتایج چشمگیری در خروجی‌های سیستم شده است که برتری محسوس GODFIS را نسبت به سایر روش‌های موجود تأیید می‌کند. به عبارت دیگر، بهبود مشاهده شده در عملکرد، نشان‌دهنده کارایی بالای معماری پیشنهادی و انتخاب مناسب استراتژی‌های بهینه‌سازی در طراحی این الگوریتم می‌باشد.

### اعتبار سنجی مدل پیشنهادی در کاربری کنترل

#### روی مثال NLC

این مثال، در مقاله [۴۵] برای اثبات عملکرد یک مدل پیش‌بینی بلادرنگ برای مسائل کنترل استفاده شده است و یک سیستم غیرخطی را زیر در نظر می‌گیرد (فرمول (۳۰)).

$$y(t) = 0.9722y(t-1) + 0.378u(t-1) - 0.1295u(t-2) - 0.3103y(t-1)u(t-1) - 0.04228y^2(t-2) + 0.1663y(t-2)u(t-2) - 0.03259y^2(t-1)y(t-2) - 0.3513y^2(t-1)u(t-2) + 0.3084y(t-1)y(t-2)u(t-2) + 0.1087y(t-2)u(t-1)u(t-2) + v(t) \quad (30)$$



شکل ۱۶. نمودار هیستوگرام خطاها روی داده‌های تست Delta Ailerons

### مقایسه‌ی به‌روزرسانی محلی و یا سراسری پارامترهای تالی

همان‌طور که گفته شد، الگوریتم GODFIS برای به‌روزرسانی سراسری پارامترهای تالی طراحی شده است. این رویکرد سراسری منجر به ارائه نتایج با کیفیت بالاتر می‌شود، چرا که تمامی روابط و تعاملات بین پارامترها را به صورت یکپارچه در نظر می‌گیرد. به‌منظور بررسی دقیق‌تر این موضوع، تمامی مسائل معیار با استفاده از روش به‌روزرسانی محلی پارامترهای تالی نیز حل شده‌اند. مقایسه نتایج حاصل از دو رویکرد (به‌روزرسانی سراسری در برابر به‌روزرسانی محلی) در جدول ۷ ارائه شده است.

جدول ۷. مقایسه‌ی به‌روزرسانی سراسری تالی در مقابل به‌روزرسانی محلی تالی در الگوریتم GODFIS

Dataset	Model	RMS E	NDE I	#(Rule )	$t_{exe}$
Mackey -Glass	GODFI S (Global)	۰.۰۲۷۰	۰.۱۰۶	۴۲	۹.۳
	GODFI S (Local)	۰.۰۸۴۳	۰.۳۳۲	۴۲	۸.۴
Delta Ailerons	GODFI S (Global)	۰.۰۴۴۹	۰.۵۴۳	۲	۱۱.۷
	GODFI S (Local)	۰.۰۴۵۶	۰.۵۴۹	۲	۱۱.۳

آوری داده‌ها از مولتی روتور مبتنی بر Pixhawk در آزمایشگاه پهپاد دانشگاه نیو ساوت ولز، پردیس کانبرا انجام شده و در مدل آزمایشی، از یک چارچوب خودکار پیش‌رونده بهره گرفته شده که توسط یک پروژه سخت‌افزاری متن‌باز و مستقل به نام PX4 [۱۲۱] برنامه‌ریزی شده است.

این یک فرایند شناسایی فازی غیرخطی MIMO با چهار ورودی و چهار خروجی (ارتفاع عمودی و سه حرکت چرخشی) در محدوده زمانی ۰ تا ۱۸۰ ثانیه و گام زمانی ۰.۰۲ ثانیه، می‌باشد. هر مجموعه داده ورودی شامل حدود ۹۰۰۰ نمونه است که ۶۰٪ نمونه‌های ابتدایی برای آموزش سیستم و ۴۰٪ باقیمانده برای تست سیستم استفاده می‌شود. مقایسه بین مدل GODFIS و مدل eTS مورد استفاده در مقاله [۱۱۶] بر اساس معیار RMSE در جدول ۹ ارائه شده است.

جدول ۹. مقایسه خروجی GODFIS و eTS با استفاده از معیار

Properties	RMSE	
	eTS	GODFIS
Altitude Z	۰.۰۰۲۰	۰.۰۰۰۶
Roll	۰.۰۱۷	۰.۰۰۵۳
Pitch	۰.۰۱۳	۰.۰۰۳۶
Yaw	۰.۰۰۴۶	۰.۰۰۰۱

نتایج به وضوح نشان می‌دهد که الگوریتم GODFIS در تمامی چهار خروجی مورد بررسی، عملکردی برتر نسبت به سیستم eTS ارائه کرده است. این برتری از قبل قابل پیش‌بینی بود، چرا که GODFIS با بهره‌گیری از مکانیزم‌های بهینه‌سازی پیشرفته، توانسته است عملکردی متمایز نسبت به دیگر الگوریتم‌های EFS فاقد این مکانیزم‌ها ارائه دهد. این یافته‌ها به وضوح نشان می‌دهد که ادغام مکانیزم‌های بهینه‌سازی در طراحی GODFIS، منجر به دستیابی به چنین نتایج قابل توجهی شده است.

#### اعتبارسنجی عملکرد اصل حذف نویز

به منظور ارزیابی عملی و کمی عملکرد اصل حذف نویز، آزمایش‌های مختلفی بر روی چهار بنچمارک معتبر قبلی (MG [۱۱۴]، Ailerons [۱۱۵]، NLC [۴۵]، و PX4 [۱۱۶]) انجام شد. در تمامی این آزمایش‌ها، مقادیر قابل توجهی نویز گاوسی با میانگین صفر و واریانس متغیر بین ۰.۰۱ تا ۰.۱ به داده‌ها اضافه گردید تا شرایط عملیاتی غیرایده‌آل شبیه‌سازی

در این آزمایش، فرمول (۳۰) ناشناخته فرض می‌شود و هدف پیش‌بینی و ردیابی خروجی سیستم یعنی  $y(k)$  می‌باشد.

از آنجایی که تغییر در انتخاب ورودی ممکن است به طور قابل توجهی بر عملکرد الگوریتم تأثیر بگذارد، برای مقایسه GODFIS با سایر روش‌های پیشنهادی تنظیمات ارائه شده در مقاله [۴۵] به کار گرفته شده است. در این حالت، تخمین  $y(t)$  بر اساس فرمول (۳۱) و با شرایط اولیه  $u(1) = u(2) = 1$  و  $y(1) = y(2) = 0$  انجام می‌شود:

$$\hat{y}(t) = f(y(t-1), y(t-2), u(t-1), u(t-2)) \quad (31)$$

همچنین برای نشان دادن دقت مدل فرض می‌شود که ورودی به صورت  $u(t) = \sin(t)$ ;  $3 \leq t \leq 400$  شناخته شده باشد.

عملکرد مدل‌ها بر اساس معیار RMSE سنجیده شده و نتایج در جدول ۸ ارائه شده است. همان‌طور که قابل مشاهده است، پیش‌بینی‌های به دست آمده توسط GODFIS نسبت به خروجی واقعی، دقت بالاتری نسبت به مدل‌های دیگر دارد که نشان‌دهنده توانایی بالای GODFIS در مدل‌سازی سیستم‌های غیرخطی است.

جدول ۸. مقایسه GODFIS با سایر روش‌ها روی داده کنترل

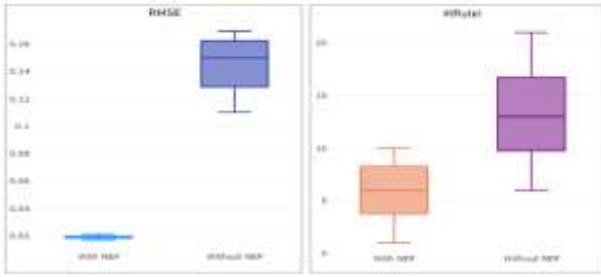
Model	Model Type	RMSE	غیر خطی
			$t_{exe}$
GODFIS	T-S	۰.۰۱۷	۲.۷
SORBFNN [۴۵]	RBF	۰.۰۲۳	-
MLP [۱۱۸]	MLP	۱.۰۲۱	-
MRAN [۱۱۹]	RBF	۰.۳۲۵	-
GGAP-RBF [۱۲۰]	RBF	۰.۱۰۶	-

#### اعتبارسنجی مدل پیشنهادی در کاربری کنترل روی

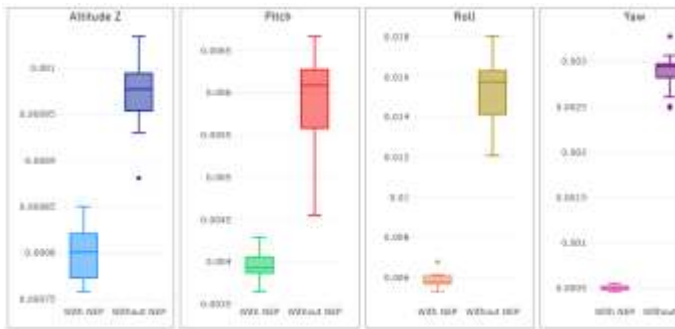
##### مجموعه داده PX4

در مقاله [۱۱۶]، از سیستم فازی eTS برای توسعه یک مدل کنترل مولتی روتور با چند ورودی و چند خروجی (MIMO)<sup>۶۴</sup> استفاده شده است. این مدل بر اساس داده‌های آزمایشی پرواز مولتی روتور توسعه یافته است. فرایند جمع

<sup>64</sup> Multiple-Input-Multiple-Output



شکل ۱۹. مقایسه‌ی معیار RMSE روی دیتای NLC با اعمال و بدون اعمال اصل حذف نویز



شکل ۲۰. مقایسه‌ی معیار RMSE روی دیتای PX4 با اعمال و بدون اعمال اصل حذف نویز

این نتایج به طور یکپارچه تأیید می‌کنند که NEP با کاهش حساسیت مدل به نویز، دقت پیش‌بینی را افزایش داده و پیچیدگی مدل را بهینه می‌کند. این نتایج در سناریوهای متنوع سری‌های زمانی و سیستم‌های کنترلی تأیید شده‌اند. تحلیل‌های انجام‌شده از بنچمارک‌های مختلف به‌وضوح نشان می‌دهند که اصل حذف نویز نه تنها یک روش نظری، بلکه یک راهکار عملی و بسیار مؤثر برای بهبود دقت، پایداری و کارایی مدل‌ها در داده‌های پر از نویز است. این اصل با موفقیت توانسته است بر چالش‌های موجود در سیستم‌های آشوبناک، غیرخطی و عملیاتی غلبه کند و پتانسیل بالای خود را برای کاربرد در سیستم‌های واقعی و بحرانی به اثبات برساند.

### نتیجه‌گیری

مدل‌سازی و کنترل سیستم‌های پیچیده و بسیار غیرخطی، با دقت بالا، همواره یکی از چالش‌های اساسی در مهندسی کنترل بوده است. محدودیت‌های تکنیک‌های مرسوم FPT، سیستم‌های فازی و عصبی-فازی مختلف را به گزینه‌ای مناسب برای مقابله با این چالش تبدیل کرده است. با این حال، اغلب این سیستم‌ها مبتنی بر یادگیری دسته‌ای عمل می‌کنند که

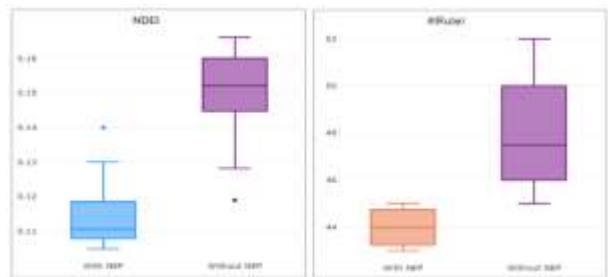
شود. این مدل نویز، شبیه‌ساز نویزهای سنسوری رایج در شتاب‌سنج‌ها، ژيروسکوپ‌ها و GPS مولتی روتورها است.

در بنچمارک MG، کاربرد NEP منجر به کاهش حدود ۲۳ درصدی در خطای NDEI و حدود ۱۰ درصدی در تعداد قوانین شد که نشان‌دهنده بهبود هم‌زمان دقت و ساده‌سازی ساختار مدل است (شکل ۱۷).

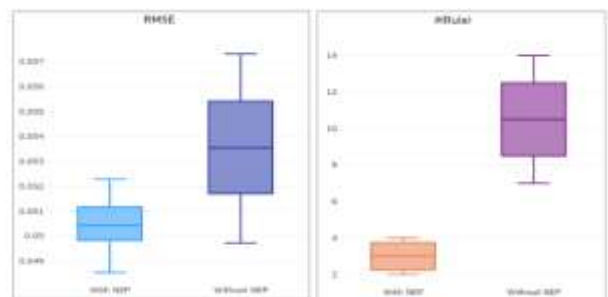
در بنچمارک Delta Ailerons، NEP توانست معیار RMSE را حدود ۱۱ درصد و تعداد قوانین را حدود ۶۷ درصد کاهش دهد که نشان‌دهنده بهبود چشمگیر در دقت و ساده‌سازی مدل است (شکل ۱۸).

در بنچمارک NLC، کاهش قابل‌توجه ۸۵ درصدی در معیار RMSE و ۶۶ درصدی در تعداد قوانین در حضور NEP مشاهده شد که بیانگر کارایی بالای NEP در مدیریت و حذف نویز در داده‌ها است (شکل ۱۹).

نهایتاً، در بنچمارک PX4، تأثیر NEP بر متغیرهای مختلف شامل Altitude Z، Pitch، Roll و Yaw بررسی شد. نتایج نشان داد که خطاها به ترتیب ۱۵ درصد، ۴۲ درصد، ۶۲٫۵ درصد و ۸۳ درصد کاهش یافته‌اند (شکل ۲۰).



شکل ۱۷. مقایسه معیار NDEI روی دیتای MG با اعمال و بدون اعمال اصل حذف نویز



شکل ۱۸. مقایسه‌ی معیار RMSE روی دیتای Delta Ailerons با اعمال و بدون اعمال اصل حذف نویز

گامی مؤثر در جهت حل چالش‌های کنترل سیستم‌های پیچیده غیرخطی برداشته است.

در حوزه تحقیقات آتی، با توجه به توانمندی و کاربرد گسترده GODFIS، می‌توان از آن در مطالعات موردی گوناگون در زمینه پیش‌بینی و کنترل بهره گرفت. همچنین، با وجود کیفیت بالای نتایج، می‌توان در آینده به بررسی ایجاد ارتباط میان بهینه‌سازی بخش مقدم و تالی و انجام بهینه‌سازی هم‌زمان آن‌ها پرداخت. از سوی دیگر، توسعه نسخه‌ای ساده‌تر از یک EFS با کاهش پارامترهای موردنیاز نیز ادامه خواهد یافت که بی‌تردید موجب کاهش پیچیدگی و گشودن افق‌های تازه در توسعه سیستم‌های پیشرفته مدل‌سازی و کنترل مولتی‌روتورها خواهد شد.

### منابع و مراجع

- [1] Pourmand, M., Rostamy Malkhalifeh, M., Afshar Kazemi, M. A., & Keramati, M. A. (2025). Measuring the Efficiency of Iranian Airlines Manager Using Network Data Envelopment Analysis. *Journal of Aeronautical Engineering*, 27(1), 86–101.
- [2] Sadeghimoghaddam, M., Safari, H., Dehaghan, M., & Rezaeifar, M. (2023). Analysis of aviation companies' efficiency using Data Envelopment Analysis. *Journal of Aeronautical Engineering*, 25(2), 48–67.
- [3] Mostafa, S. A., Ahmad, M. S., & Mustapha, A. (2019). Adjustable autonomy: a systematic literature review. *Artificial Intelligence Review*, 51(2), 149–186.
- [4] Ferdaus, M. M., Anavatti, S. G., Pratama, M., & Garratt, M. A. (2020). Towards the use of fuzzy logic systems in rotary wing unmanned aerial vehicle: a review. *Artificial Intelligence Review*, 53(1), 257–290.
- [5] Simplício, P. V. G., Benevides, J. R. S., Inoue, R. S., & Terra, M. H. (2024). Robust and intelligent control of quadrotors subject to wind gusts. *IET Control Theory & Applications*, 18(18), 2594–2611.
- [6] Hong, Y.-S. T., & White, P. A. (2009). Hydrological modeling using a dynamic neuro-fuzzy system with on-line and local learning algorithm. *Advances in Water Resources*, 32(1), 110–119.
- [7] Garibaldi, J. M. (2019). The need for fuzzy AI. *IEEE/CAA Journal of Automatica Sinica*, 6(3), 610–622.

منجر به عملکرد ضعیف در مدیریت عدم قطعیت‌ها می‌شود. هرچند تنظیم پارامترهای سیستم فازی تا حدی این مشکل را برطرف می‌کند، اما به دلیل ساختار ایستا، این سیستم‌ها قادر به سازگاری با تغییرات ناگهانی در رفتار سیستم‌های پیچیده و بسیار غیرخطی نیستند.

در چنین شرایطی، سیستم‌های فازی مبتنی بر EFS گزینه‌ای مناسب به شمار می‌آیند، زیرا می‌توانند ساختار خود را برای مقابله با تغییرات شدید، به صورت پویا تغییر دهند. در این مقاله، یک الگوریتم EFS آنلاین جدید با نام GODFIS برای مدل‌سازی و کنترل سیستم‌های پیچیده و بسیار غیرخطی ارائه شده است. با توجه به فقدان تمرکز کافی روش‌های موجود در ادبیات بر بهینه‌سازی EFS، تلاش می‌کند این شکاف را با بهره‌گیری از مفهوم مرکز ثقل (که در روش‌های آفلاین رایج است) و همچنین بهینه‌سازی سراسری پارامترهای تالی پوشش دهد. در این راستا، یک روش نوین با نام GOECA که مبتنی بر محاسبه بازگشتی مرکز ثقل است، برای بهینه‌سازی خوشه‌بندی در فضای جواب و تعیین پارامترهای مقدم ارائه می‌شود.

GODFIS، همانند سایر سیستم‌های خودسازمانده، توانایی رشد، هرس و تنظیم ساختار خود را در فرایند یادگیری آنلاین دارد. افزون بر این، این الگوریتم به یک تکنیک جدید با نام اصل حذف نویز مجهز است که ضمن حفظ کیفیت نتایج و جلوگیری از اثرات نامطلوب نویز، امکان کاهش ابعاد به صورت آنلاین را برای مقابله با نفرین ابعاد فراهم می‌کند. تمامی مکانیزم‌های یادگیری در GODFIS به صورت تک‌گذری و آنلاین انجام می‌شوند و نیازی به یادگیری مجدد داده‌های تاریخی وجود ندارد.

این روش بر روی داده‌های معیار متداول با دیگر روش‌های موجود در ادبیات مقایسه شده و نتایج آزمایش‌ها بیانگر دقت و کارایی بالای آن است. برای نمایش کارایی عملی در کاربردهای کنترل، GODFIS بر روی دو مجموعه داده معیار مرتبط با این حوزه پیاده‌سازی شده است. نتایج نشان می‌دهد این الگوریتم علاوه بر ارائه پاسخی با کیفیت بالا، قابلیت استفاده در مسائل مختلف کنترل — از جمله مدل‌سازی و کنترل مولتی‌روتورها — را دارد. این نتایج نشان می‌دهد که این الگوریتم با ترکیب هوشمندانه تکنیک‌های بهینه‌سازی و یادگیری تطبیقی،

- [20] Hailemichael, A., & Karimoddini, A. (2023). Development of a Robust Interval Type-2 TSK Fuzzy Logic Controlled UAV Platform. *Journal of Intelligent & Robotic Systems*, 107(2), 27.
- [21] Zareb, M., Ayad, R., & Nouibat, W. (2013). Fuzzy-PID hybrid control system to navigate an autonomous mini-Quadrotor. In *3rd International Conference on Systems and Control* (pp. 906–913).
- [22] Siyuan, C., Ting, Z., Yanchun, C., Jianliang, C., Junyi, S., & Jin, L. (2012). A research on quadrotor attitude control based on Fuzzy-PI method. In *Proceedings of the 31st Chinese Control Conference* (pp. 3548–3551).
- [23] Gautam, D., & Ha, C. (2013). Control of a Quadrotor Using a Smart Self-Tuning Fuzzy PID Controller. *International Journal of Advanced Robotic Systems*, 10(11), 380.
- [24] Sangyam, T., Laohapiengsak, P., Chongcharoen, W., & Nilkhamhang, I. (2010). Path tracking of UAV using self-tuning PID controller based on fuzzy logic. In *Proceedings of SICE Annual Conference 2010* (pp. 1265–1269).
- [25] Mehranpour, M. R., Shahri, A. M., Emamgholi, O., & Farrokhi, M. (2013). A new fuzzy adaptive control for a Quadrotor flying robot. In *2013 13th Iranian Conference on Fuzzy Systems (IFSC)* (pp. 1–5).
- [26] Yanjun, L., Tianqi, X., & Xiaodong, Z. (2016). A fuzzy self-tuning PD controller for a quadrotor: Design and implementation. In *2016 Chinese Control and Decision Conference (CCDC)* (pp. 2448–2453).
- [27] Hwang, C. L., & Chau, J. (2012). Fuzzy decentralized sliding-mode under-actuated trajectory-tracking control for quadrotor unmanned aerial vehicle. In *2012 IEEE International Conference on Fuzzy Systems* (pp. 1–10).
- [28] Khebbache, H., & Tadjine, M. (2013). Robust fuzzy backstepping sliding mode controller for a quadrotor unmanned aerial vehicle. *Journal of Control Engineering and Applied Informatics*, 15(2), 3–11.
- [29] Niroumand, F. J., Fakharian, A., & Seyedsajadi, M. S. (2013). Fuzzy integral backstepping control approach in attitude stabilization of a quadrotor UAV. In *2013 13th Iranian Conference on Fuzzy Systems (IFSC)* (pp. 1–6).
- [8] Jang, J. S. R. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665–685.
- [9] Chuen-Tsai, S. (1994). Rule-base structure identification in an adaptive-network-based fuzzy inference system. *IEEE Transactions on Fuzzy Systems*, 2(1), 64–73.
- [10] Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1), 116–132.
- [11] Wang, L.-X., & Mendel, J. M. (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE transactions on Neural Networks*, 3(5), 807–814.
- [12] Takagi, T., & Sugeno, M. (1983). Derivation of Fuzzy Control Rules from Human Operator's Control Actions. *IFAC Proceedings Volumes*, 16(13), 55–60.
- [13] Sugeno, M., & Kang, G. T. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1), 15–33.
- [14] Melin, P., Mancilla, A., Lopez, M., & Mendoza, O. (2007). A hybrid modular neural network architecture with fuzzy Sugeno integration for time series forecasting. *Applied Soft Computing*, 7(4), 1217–1226.
- [15] Melin, P., & Castillo, O. (2003). Adaptive intelligent control of aircraft systems with a hybrid approach combining neural networks, fuzzy logic and fractal theory. *Applied Soft Computing*, 3(4), 353–362.
- [16] Melin, P., & Castillo, O. (2007). An intelligent hybrid approach for industrial quality control combining neural networks, fuzzy logic and fractal theory. *Information Sciences*, 177(7), 1543–1557.
- [17] Ontiveros-Robles, E., Melin, P., & Castillo, O. (2018). Comparative analysis of noise robustness of type 2 fuzzy logic controllers. *Kybernetika*, 54(1), 175–201.
- [18] Castillo, O., Cervantes, L., Soria, J., Sanchez, M., & Castro, J. R. (2016). A generalized type-2 fuzzy granular approach with applications to aerospace. *Information Sciences*, 354, 165–177.
- [19] Chen, W., Wang, Z., & Zhou, Z. (2025). Optimized interval type-2 fuzzy global sliding mode control for quadrotor robot. *Meccanica*, 60(2), 457–474.

- [42] Al-Mahasneh, A. J., Anavatti, S. G., & Garratt, M. (2017). Nonlinear multi-input multi-output system identification using neuro-evolutionary methods for a quadcopter. In 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI) (pp. 217–222).
- [43] Yang, G., & Meng Joo, E. (2003). Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems. *IEEE Transactions on Fuzzy Systems*, 11(4), 462–477.
- [44] Barros, J. C. d., & Dexter, A. L. (2007). Evolving Fuzzy Model-based Adaptive Control. In 2007 IEEE International Fuzzy Systems Conference (pp. 1–5).
- [45] Han, H. G., Wu, X. L., & Qiao, J. F. (2013). Real-Time Model Predictive Control Using a Self-Organizing Neural Network. *IEEE Transactions on Neural Networks and Learning Systems*, 24(9), 1425–1436.
- [46] Ferdaus, M. M., Pratama, M., Anavatti, S. G., & Garratt, M. A. (2019). PALM: An Incremental Construction of Hyperplanes for Data Stream Regression. *IEEE Transactions on Fuzzy Systems*, 27(11), 2115–2129.
- [47] Ferdaus, M. M., Pratama, M., Anavatti, S. G., Garratt, M. A., & Pan, Y. (2020). Generic Evolving Self-Organizing Neuro-Fuzzy Control of Bio-Inspired Unmanned Aerial Vehicles. *IEEE Transactions on Fuzzy Systems*, 28(8), 1542–1556.
- [48] Ferdaus, M. M., Pratama, M., Anavatti, S. G., Garratt, M. A., & Lughofer, E. (2020). PAC: A novel self-adaptive neuro-fuzzy controller for micro aerial vehicles. *Information Sciences*, 512, 481–505.
- [49] Gu, X., Shen, Q., & Angelov, P. P. (2021). Particle Swarm Optimized Autonomous Learning Fuzzy System. *IEEE Transactions on Cybernetics*, 51(11), 5352–5363.
- [50] Bouabdallah, S., Noth, A., & Siegwart, R. (2004). PID vs LQ control techniques applied to an indoor micro quadrotor. In \*2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)\* (pp. 2451–2456). IEEE.
- [51] Bacik, J., Perdukova, D., & Fedor, P. (2015). Design of Fuzzy Controller for Hexacopter Position Control. In Springer International Publishing (pp. 193–202). Cham.
- [30] Freeman, J. A. S., & Saad, D. (1997). Online Learning in Radial Basis Function Networks. *Neural Computation*, 9(7), 1601–1622.
- [31] French, R. M. (1992). Semi-distributed Representations and Catastrophic Forgetting in Connectionist Networks. *Connection Science*, 4(3-4), 365–377.
- [32] Heskes, T. M., & Kappen, B. (1993). On-line learning processes in artificial neural networks. In J. G. Taylor (Ed.), *North-Holland Mathematical Library* (pp. 199–233). Elsevier.
- [33] Robins, A. V., & Frean, M. R. (1998). Local learning algorithms for sequential tasks in neural networks. *J. Adv. Comput. Intell. Informatics*, 2(6), 221–227.
- [34] Rummery, G. A., & Niranjan, M. (1994). On-line Q-learning using connectionist systems. University of Cambridge, Department of Engineering, Cambridge, UK.
- [35] Angelov, P. P., & Filev, D. P. (2004). An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1), 484–498.
- [36] Kasabov, N. K., & Qun, S. (2002). DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10(2), 144–154.
- [37] Gu, X., Han, J., Shen, Q., & Angelov, P. P. (2023). Autonomous learning for fuzzy systems: a review. *Artificial Intelligence Review*, 56(8), 7549–7595.
- [38] Angelov, P. (2004). A fuzzy controller with evolving structure. *Information Sciences*, 161(1), 21–35.
- [39] Lughofer, E. D. (2008). FLEXFIS: A Robust Incremental Learning Approach for Evolving Takagi-Sugeno Fuzzy Models. *IEEE Transactions on Fuzzy Systems*, 16(6), 1393–1410.
- [40] Angelov, P. (2011). Fuzzily Connected Multimodel Systems Evolving Autonomously From Data Streams. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(4), 898–910.
- [41] Ferdaus, M. M., Pratama, M., Anavatti, S. G., & Garratt, M. A. (2019). Online identification of a rotary wing Unmanned Aerial Vehicle from data streams. *Applied Soft Computing*, 76, 313–325.

- [63] Angelov, P. (2010). Evolving Takagi-Sugeno Fuzzy Systems from Streaming Data (eTS+). In *Evolving Intelligent Systems* (pp. 21–50).
- [64] Angelov, P. P., & Zhou, X. (2008). Evolving Fuzzy-Rule-Based Classifiers From Data Streams. *IEEE Transactions on Fuzzy Systems*, 16(6), 1462–1475.
- [65] Baruah, R. D., & Angelov, P. (2014). DEC: Dynamically Evolving Clustering and Its Application to Structure Identification of Evolving Fuzzy Models. *IEEE Transactions on Cybernetics*, 44(9), 1619–1631.
- [66] Rubio, J. d. J. (2009). SOFMLS: Online Self-Organizing Fuzzy Modified Least-Squares Network. *IEEE Transactions on Fuzzy Systems*, 17(6), 1296–1309.
- [67] Lemos, A., Caminhas, W., & Gomide, F. (2011). Multivariable Gaussian Evolving Fuzzy Modeling System. *IEEE Transactions on Fuzzy Systems*, 19(1), 91–104.
- [68] Subramanian, K., & Suresh, S. (2012). A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system. *Applied Soft Computing*, 12(11), 3603–3614.
- [69] Rong, H.-J., Sundararajan, N., Huang, G.-B., & Saratchandran, P. (2006). Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction. *Fuzzy Sets and Systems*, 157(9), 1260–1275.
- [70] Rong, H.-J., Sundararajan, N., Huang, G.-B., & Zhao, G.-S. (2011). Extended sequential adaptive fuzzy inference system for classification problems. *Evolving Systems*, 2(2), 71–82.
- [71] Pratama, M., Anavatti, S. G., Angelov, P. P., & Lughofer, E. (2014). PANFIS: A Novel Incremental Learning Machine. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1), 55–68.
- [72] Pratama, M., Anavatti, S. G., & Lughofer, E. (2014). GENEFIS: Toward an Effective Localist Network. *IEEE Transactions on Fuzzy Systems*, 22(3), 547–562.
- [73] Nguyen, N. N., Zhou, W. J., & Quek, C. (2015). GSETSK: a generic self-evolving TSK fuzzy neural network with a novel Hebbian-based rule reduction approach. *Applied Soft Computing*, 35, 29–42.
- [52] Kim, B. S., & Calise, A. J. (1997). Nonlinear Flight Control Using Neural Networks. *Journal of Guidance, Control, and Dynamics*, 20(1), 26–33.
- [53] Kurnaz, S., Cetin, O., & Kaynak, O. (2010). Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. *Expert Systems with Applications*, 37(2), 1229–1234.
- [54] Cervantes, L., & Castillo, O. (2015). Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control. *Information Sciences*, 324, 247–256.
- [55] Chen, X., Li, D., Xu, Z., & Bai, Y. (2011). Robust control of quadrotor MAV using self-organizing interval type-II fuzzy neural networks (SOIT-IIFNNs) controller. *International Journal of Intelligent Computing and Cybernetics*, 4(3), 397–412.
- [56] Cheng, D., Ning, W., & Meng, J. E. (2016). Self-organizing adaptive robust fuzzy neural attitude tracking control of a quadrotor. In *2016 35th Chinese Control Conference (CCC)* (pp. 10724–10729).
- [57] Singh, R., Nishad, D. K., Khalid, S., & Chaudhary, A. (2025). A review of the application of fuzzy mathematical algorithm-based approach in autonomous vehicles and drones. *International Journal of Intelligent Robotics and Applications*, 9(1), 344–364.
- [58] Precup, R.-E., Anh-Tu, N., & Blažič, S. (2024). A survey on fuzzy control for mechatronics applications. *International Journal of Systems Science*, 55(4), 771–813.
- [59] Kasabov, N. (2001). Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(6), 902–918.
- [60] Angelov, P., & Buswell, R. (2002). Identification of evolving fuzzy rule-based models. *IEEE Transactions on Fuzzy Systems*, 10(5), 667–677.
- [61] Angelov, P., & Filev, D. (2005). Simpl\_eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models. In *The 14th IEEE International Conference on Fuzzy Systems, FUZZ '05* (pp. 1068–1073).
- [62] Angelov, P., & Zhou, X. (2006). Evolving Fuzzy Systems from Data Streams in Real-Time. In *2006 International Symposium on Evolving Fuzzy Systems* (pp. 29–35).

- (SPATFIS) and its stability analysis. *Information Sciences*, 505, 84–99.
- [86] Ge, D., & Zeng, X. J. (2019). A Self-Evolving Fuzzy System Which Learns Dynamic Threshold Parameter by Itself. *IEEE Transactions on Fuzzy Systems*, 27(8), 1625–1637.
- [87] Ge, D., & Zeng, X.-J. (2020). Learning data streams online — An evolving fuzzy system approach with self-learning/adaptive thresholds. *Information Sciences*, 507, 172–184.
- [88] Santoso, F., Garratt, M. A., Anavatti, S. G., & Information entropy-based pruning technique. (2020). T2-ETS-IE: A Type-2 Evolutionary Takagi–Sugeno Fuzzy Inference System With the Information Entropy-Based Pruning Technique. *IEEE Transactions on Fuzzy Systems*, 28(10), 2665–2672.
- [89] Gu, X. (2021). Multilayer Ensemble Evolving Fuzzy Inference System. *IEEE Transactions on Fuzzy Systems*, 29(8), 2425–2431.
- [90] Gu, X., & Shen, Q. (2021). A self-adaptive fuzzy learning system for streaming data prediction. *Information Sciences*, 579, 623–647.
- [91] Li, W., Qiao, J., & Zeng, X. J. (2022). Online and Self-Learning Approach to the Identification of Fuzzy Neural Networks. *IEEE Transactions on Fuzzy Systems*, 30(3), 649–662.
- [92] Yu, H., Lu, J., & Zhang, G. (2022). Topology Learning-Based Fuzzy Random Neural Networks for Streaming Data Regression. *IEEE Transactions on Fuzzy Systems*, 30(2), 412–425.
- [93] Wang, Z., & Fei, J. (2022). Fractional-Order Terminal Sliding-Mode Control Using Self-Evolving Recurrent Chebyshev Fuzzy Neural Network for MEMS Gyroscope. *IEEE Transactions on Fuzzy Systems*, 30(7), 2747–2758.
- [94] Rodrigues, S. E., & de Oliveira Serra, G. L. (2022). An approach for evolving neuro-fuzzy forecasting of time series based on parallel recursive singular spectrum analysis. *Fuzzy Sets and Systems*, 443, 1–29.
- [95] Ožbot, M., Lughofer, E., & Škrjanc, I. (2023). Evolving Neuro-Fuzzy Systems-Based Design of Experiments in Process Identification. *IEEE Transactions on Fuzzy Systems*, 31(6), 1995–2005.
- [74] Lughofer, E., Cernuda, C., Kindermann, S., & Pratama, M. (2015). Generalized smart evolving fuzzy systems. *Evolving Systems*, 6(4), 269–292.
- [75] Dovžan, D., & Škrjanc, I. (2011). Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes. *ISA Transactions*, 50(2), 159–169.
- [76] Dovžan, D., & Škrjanc, I. (2011). Recursive clustering based on a Gustafson–Kessel algorithm. *Evolving Systems*, 2(1), 15–24.
- [77] Dovžan, D., Loga, V., & Škrjanc, I. (2012). Solving the sales prediction with fuzzy evolving models. In *WCCI 2012 IEEE World Congress on Computational Intelligence* (p. 10–15). Brisbane, Australia.
- [78] Lima, E., Hell, M., Ballini, R., & Gomide, F. (2010). Evolving Fuzzy Modeling Using Participatory Learning. In *Evolving Intelligent Systems* (pp. 67–86).
- [79] Maciel, L., Gomide, F., & Ballini, R. (2014). Enhanced evolving participatory learning fuzzy modeling: an application for asset returns volatility forecasting. *Evolving Systems*, 5(2), 75–88.
- [80] Maciel, L., Gomide, F., & Ballini, R. (2014). Recursive possibilistic fuzzy modeling. In *2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)* (pp. 9–16).
- [81] Maciel, L., Ballini, R., & Gomide, F. (2017). Evolving possibilistic fuzzy modelling. *Journal of Statistical Computation and Simulation*, 87(7), 1446–1466.
- [82] Pratama, M., Lu, J., Lughofer, E., Zhang, G., & Er, M. J. (2017). An Incremental Learning of Concept Drifts Using Evolving Type-2 Recurrent Fuzzy Neural Networks. *IEEE Transactions on Fuzzy Systems*, 25(5), 1175–1192.
- [83] Bao, R. J., Rong, H. J., Angelov, P. P., Chen, B., & Wong, P. K. (2018). Correntropy-Based Evolving Fuzzy Neural System. *IEEE Transactions on Fuzzy Systems*, 26(3), 1324–1338.
- [84] Rong, H. J., Yang, Z. X., & Wong, P. K. (2020). Robust and Noise-Insensitive Recursive Maximum Correntropy-Based Evolving Fuzzy System. *IEEE Transactions on Fuzzy Systems*, 28(9), 2277–2284.
- [85] Samanta, S., Pratama, M., & Sundaram, S. (2019). A novel Spatio-Temporal Fuzzy Inference System

- [107] Gu, X., & Angelov, P. (2019). Self-boosting first-order autonomous learning neuro-fuzzy systems. *Applied Soft Computing*, 77, 118–134.
- [108] Angelov, P. P., Gu, X., & Príncipe, J. C. (2018). Autonomous Learning Multimodel Systems From Data Streams. *IEEE Transactions on Fuzzy Systems*, 26(4), 2213–2224.
- [109] Soltic, S., Pang, S., Kasabov, N., Worner, S., & Peacock, L. (2004). Dynamic Neuro-fuzzy Inference and Statistical Models for Risk Analysis of Pest Insect Establishment. In *Springer Berlin Heidelberg* (pp. 971–976). Berlin, Heidelberg.
- [110] Talei, A., Chua, L. H. C., Quek, C., & Jansson, P.-E. (2013). Runoff forecasting using a Takagi–Sugeno neuro-fuzzy model with online learning. *Journal of Hydrology*, 488, 17–32.
- [111] Ravi, V., Srinivas, E. R., & Kasabov, N. K. (2007). On-Line Evolving Fuzzy Clustering. In *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)* (pp. 347–351).
- [112] Lughofer, E. (2011). EFS Approaches for Regression and Classification. In *Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications* (pp. 93–164). Springer Berlin Heidelberg.
- [113] Yen, J., Liang, W., & Gillespie, C. W. (1998). Improving the interpretability of TSK fuzzy models by combining global learning and local learning. *IEEE Transactions on Fuzzy Systems*, 6(4), 530–537.
- [114] Mackey, M. C., & Glass, L. (1977). Oscillation and Chaos in Physiological Control Systems. *Science*, 197(4300), 287–289.
- [115] Alcalá-fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2011). KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17.
- [116] Ferdous, M. M., Anavatti, S. G., Garratt, M. A., & Pratama, M. (2017). Evolving fuzzy inference system based online identification and control of a quadcopter unmanned aerial vehicle. In *2017 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)* (pp. 223–228).
- [96] Gu, X., Angelov, P., Han, J., & Shen, Q. (2023). Multilayer Evolving Fuzzy Neural Networks. *IEEE Transactions on Fuzzy Systems*, 31(12), 4158–4169.
- [97] Hu, L., Xu, X., Ren, W., & Han, M. (2024). Hierarchical Evolving Fuzzy System: A Method for Multidimensional Chaotic Time Series Online Prediction. *IEEE Transactions on Fuzzy Systems*, 32(6), 3329–3341.
- [98] Lughofer, E. (2011). *Evolving fuzzy systems-methodologies, advanced concepts and applications*. Springer.
- [99] Fernandez, A., Herrera, F., Cordon, O., Jesus, M. J. d., & Marcelloni, F. (2019). Evolutionary Fuzzy Systems for Explainable Artificial Intelligence: Why, When, What for, and Where to? *IEEE Computational Intelligence Magazine*, 14(1), 69–81.
- [100] Škrjanc, I., Iglesias, J. A., Sanchis, A., Leite, D., Lughofer, E., & Gomide, F. (2019). Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A Survey. *Information Sciences*, 490, 344–368.
- [101] de Campos Souza, P. V. (2020). Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Applied Soft Computing*, 92, 106275.
- [102] Leite, D., Škrjanc, I., & Gomide, F. (2020). An overview on evolving systems and learning from stream data. *Evolving Systems*, 11(2), 181–198.
- [103] Ge, D., & Zeng, X.-J. (2018). Learning evolving T–S fuzzy systems with both local and global accuracy – A local online optimization approach. *Applied Soft Computing*, 68, 795–810.
- [104] Hu, L., Xu, X., Liu, J., Yan, X., & Han, M. (2025). MIIPSO-EFS: Learning system with self-optimized parameters for chaotic time series online prediction. *Knowledge-Based Systems*, 310, 112878.
- [105] Gu, X., Angelov, P., & Rong, H.-J. (2019). Local optimality of self-organising neuro-fuzzy inference systems. *Information Sciences*, 503, 351–380.
- [106] Rong, H. J., Angelov, P. P., Gu, X., & Bai, J. M. (2018). Stability of Evolving Fuzzy Systems Based on Data Clouds. *IEEE Transactions on Fuzzy Systems*, 26(5), 2774–2784.

[117] Dovžan, D., Logar, V., & Škrjanc, I. (2015). Implementation of an Evolving Fuzzy Model (eFuMo) in a Monitoring System for a Waste-Water Treatment Process. *IEEE Transactions on Fuzzy Systems*, 23(5), 1761–1776.

[118] Chandramouli, V., Brion, G., Neelakantan, T. R., & Lingireddy, S. (2007). Backfilling missing microbial concentrations in a riverine database using artificial neural networks. *Water Research*, 41(1), 217–227.

[119] Panchapakesan, C., Palaniswami, M., Ralph, D., & Manzie, C. (2002). Effects of moving the center's in an RBF network. *IEEE Transactions on Neural Networks*, 13(6), 1299–1307.

[120] Bortman, M., & Aladjem, M. (2009). A Growing and Pruning Method for Radial Basis Function Networks. *IEEE Transactions on Neural Networks*, 20(6), 1039–1045.

[121] 3dr.com. (2017). Pixhawk. Retrieved from [https://store.3dr.com/t/pixhawk?\\_ga=1.173771040.262707248.1476173688](https://store.3dr.com/t/pixhawk?_ga=1.173771040.262707248.1476173688) [Accessed: 27 May 2026]