Loss-of-Control Detection of a Quadrotor Using Critical Slowing Down Theory

MSc Thesis

Chris Yoon Sang Chung

TU Delft

Loss-of-Control Detection of a Quadrotor Using Critical Slowing Down Theory

MSc Thesis

Thesis report

by

Chris Yoon Sang Chung

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on May 17, 2024 at 09:30

Thesis committee:

Chair:	Dr.ir. René van Paassen
Supervisors:	Dr.ir. Coen de Visser
	Ir. Jasper van Beers
External examiner:	Dr.ir. Wouter van der Wal
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	May, 2023 - May, 2024
Student number:	4666658

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Faculty of Aerospace Engineering · Delft University of Technology

Preface

The following thesis was written to obtain my Master degree in the profile Control & Simulation at the Faculty of Aerospace Engineering at Delft University of Technology. The thesis involved improving LoC detection in quadrotors using CSD theory, and actuator phase delay as an EWS, alongside providing a quantitative definition of LoC. This research aims to examine the roles of CSD, and actuator phase delay in detecting upcoming LoC events, and explore their integration using a Fuzzy Logic Inference System (FLIS) to enhance the accuracy and robustness of LoC detection while minimizing both false positives and negatives. Additionally, the study will assess how different quadrotors (systems) experiencing different modes of loss of control, such as yaw-induced vs roll-induced LoC maneuvers, impact these parameters, aiming to determine if consistent global trends exist across different systems

The public presentation and defense of this thesis are scheduled for the 17th of May 2024, before a committee comprising Associate Professor Dr.ir. Coen de Visser, Associate Professor Dr ir René van Paassen, and Associate Professor Dr.ir. Wouter van der Wal.

I would like to express my gratitude towards my supervisor Associate Professor Dr.ir. Coen de Visser who has guided me throughout the thesis to reach this important milestone.

Many thanks to Ir. Jasper van Beers, whose support has been integral to my journey. His technical advice, assistance with understanding various complex theories, and assistance with various coding/plotting difficulties, have been invaluable.

I am extremely grateful for my family and friends who have been supporting me throughout this whole process. First and foremost, a special thank you goes to my mom and dad.

엄마, 아빠, 드디어 나의 19년 간의 공부하는 삶이 끝났네요. 어렸을 때 공부를 그렇게 싫어했던 내가 고등학교를 끝내고 대학교와 석사 과정까지 갈 줄은 몰랐어요. 하지만 엄마의 잔소리와 아빠가 경영학 석사를 하지 말라고 했던 추천 덕분에 오늘 이렇게 졸업을 할 수 있었어요. 엄마, 아빠가 내 교육에 투자한 돈을 드디어 갚을 시간이 왔네요. 빨리 취직해서 돈을 벌어, 엄마 아빠 레스토랑에 투자할게요. 내 25년 인생을 잘 키워주셔서 너무 감사합니다. 엄마, 아빠 사랑해요

Next, a special thank you to my late grandmother, who raised me to be the man I am today.

할머니, 저는 오늘 제 석사 졸업을 해요! 어렸을 때부터 그렇게 떼를 쓰던 윤상이가 드디어 어른이 되었네요. 저는 할머니께서 하늘나라에서 저를 지켜보고 계셔서 제가 이렇게 졸업할 수 있었던 것 같아요. 할머니가 저를 계속 하늘나라에서 지켜봐 주신다면, 제가 할머니가 자랑스러워하실 수 있는 훌륭한 손자가 되도록 계속 노력하겠습니다.

Next, a thank you to my sister, who I could always rely on to call whenever I needed a break, and who provided me with life advice, despite being four years younger than me.

Lastly, but certainly not least, I express immense gratitude to my friends, the Foulkeslaanders. Moving to Delft, I was quite pessimistic about finding friends because I had a very skewed perception of engineers. However, within the first week of being in Delft, I was fortunate enough to make life-long friends who also became my roommates. In addition to the original Foulkeslaanders, I extend my thanks to everyone in the Foulkeslaan Extended. Thank you for consistently joining me at Bouwpub whenever I needed a break from studying. You all have made my life in Delft incredibly memorable and enjoyable. I am certain I would not have made it this far without your support I hope you find this thesis informative and engaging.

Chris Yoon Sang Chung

Delft, May 2024

Contents

1 Introduction

1.1. Research Objective

In recent years, quadrotors have gained significant attention due to their versatile applications across various domains, including, but not limited to delivery services, aerial photography, agricultural and infrastructure surveillance, military applications, as well as personal and recreational pursuits. Their popularity can be attributed to their adaptability, ease of operation, and agility, making them become popular commercial assets [1].

The commercial drone market, predominantly composed of quadrotors, hit a valuation of $19.89 billion in 2022, underscoring their growing prominence. Furthermore, projections indicate a continuous compound annual growth rate of 13.9% from 2023 to 2030 [2], highlighting the increasing demand for these devices. With the expanding user base, there is a pressing need for enhanced safety measures, particularly in situations where drones interact with people. A notable incident involving world champion slalom skier Marcel Hirscher, who narrowly avoided a falling drone during a Slalom World Cup event, serves as a reminder of the importance of safety precautions in drone operations [3]. Loss-of-Control (LoC) emerges as the predominant culprit behind quadrotor accidents [4].

Traditionally, LoC in aircraft has been extensively studied, with research primarily focusing on detecting LoC through the determination of the safe flight envelope (SFE), and defining LoC as a deviation from the boundaries of the SFE. Notable studies include Zhang *et al.* [5] [6], who developed a closed-loop flight envelope protection system for dynamic SFE retrieval using damage identification. Van Oort [7] identified the longitudinal SFE for an F-16 under various configurations and conditions. Lombaerts *et al.* [8] introduced an online SFE determination method accounting for model uncertainties in an impaired aircraft. Chongsival *et al.* [9] proposed a LoC prediction method by monitoring aircraft states across five flight envelopes. However, these methods face significant computational challenges, as estimating the SFE through standard Hamilton-Jacobi reachability analyses is known to be computationally intensive due to the curse of dimensionality [10].

For quadrotors, which operate on much tighter time scales, traditional SFE calculation methods are impractical due to these computational constraints. To address the computational constraints, Sun and de Visser [11] investigated the SFE determination for flight envelope protection using a computationally efficient Monte-Carlo approach. Additionally, Kaffa [12] investigated the behavior of the global SFE of a quad-rotor subjected to varying system dynamics, including the effects of longitudinal center of gravity displacements and actuator dynamics, also using a computationally efficient Monte-Carlo to estimate the SFE. However, using the SFE for LoC detection still has two main limitations. Firstly, a key assumption using SFE for LoC prevention is that it presumes that maintaining the existing system SFE is the sole strategy for preventing LoC. However, this overlooks the potential for a poorly designed controller to induce LoC even within the SFE. Secondly, any estimated SFEs emerge from a given (nominal) model of the system which fails to account for dynamic changes or faults in the system.

Consequently, much of the drone-specific research has shifted towards LoC detection and recovery in the presence of faults. Van der Pluijm [13] and Hoppenbrouwer [14] have led efforts in fault-based LoC detection, employing methods such as Critical Slowing Down (CSD) theory and Lyapunov stability analysis. However, results indicate that CSD alone was not suitable for LoC detection as it would result in too

many false positives, and Lyapunov stability analysis required precise state-estimation, often necessitating external motion capturing devices such as OptiTrack.

In terms of LoC recovery in the presence of faults, Sun et al. [15] developed a multi-loop hybrid nonlinear controller for high-speed flight (9m/s) of quadrotors even with a single rotor failure (SRF). Additionally, they introduced an Incremental Nonlinear Fault-Tolerant Controller (FTC) for quadrotors with complete loss of two opposing rotors [16]. Hamadi [17] also made contributions to this field by developing a FTC for a UAV under hardware and software failures experiencing wind gusts. Lu et al. [18] designed an FTC that incorporates active fault detection and isolation, enabling real-world application of quadrotor control under rotor failure.

Despite these advancements, research into LoC in undamaged quadrotors remains sparse. One notable study by Altena *et al.* [19], tackled controller-induced LoC through a scenario-specific approach using neural networks for yaw-induced LoC detection and prediction. LoC is defined as the moment in time where the roll or pitch angle exceeds, and continues to exceed, 90 deg after the start of the yawing maneuver. The two limitations here were: firstly, the definition of LoC was very scenario specific, and does not apply in more general contexts where quadrotors perform maneuvers such as flips. Secondly, this attitude based LoC definition often failed to detect even if the dangerous yawing maneuver resulted in a crash, thus having a poor detection rate.

A gap exists in the current literature for a data-driven LoC detection in quadrotors that does not depend on the presence of hardware faults, is adaptable to various LoC scenarios, and remains computationally viable, without relying on precise state estimation of the quadrotor. Considering this gap, the research objective is to improve LoC detection in quadrotors using CSD theory, and actuator phase delay as an EWS, alongside a quantitative definition of LoC. This study aims to examine the roles of CSD, and actuator phase delay in detecting upcoming LoC events, and explore their integration using a Fuzzy Logic Inference System (FLIS) to enhance the accuracy and robustness of LoC detection while minimizing both false positives and negatives. Additionally, the study will assess how different quadrotors (systems) experiencing different modes of loss of control, such as yaw-induced vs roll-induced LoC maneuvers, impact these parameters, aiming to determine if consistent global trends exist across different systems.

Research Objective

The overarching objective of this research is to enhance quadrotor Loss-of-Control (LoC) detection using Critical Slowing Down (CSD) theory and actuator phase delay as Early Warning Signals (EWS). Through a Fuzzy Logic Inference System (FLIS), we aim to improve both the accuracy and response time of LoC detection while minimizing the occurrence of false positives and negatives. Additionally, we will explore how different LoC modes, such as yaw-induced versus roll-induced, affect these parameters to identify consistent global trends across various quadrotor systems.

1.2. Research Questions

Main Research Question

What early warning signals (EWS) act as precursors for the onset of loss-of-control (LoC) in quadrotors approaching a critical transition, if any?

Sub-Research Questions

RQ1 : How can LoC in quadrotors be accurately detected and defined using a data-driven method, considering data sources, sensor inputs, and data analysis techniques?

- What data sources and sensors can effectively capture relevant information about an upcoming LoC event?
- What data pre-processing techniques are required to extract informative features for LoC detection?
- Can sliding window analysis and the data pre-processing methods applied to the sensor outputs be performed in real-time and onboard the drone for timely LoC detection?

RQ2 : How can different EWS indicators, when used collectively, contribute to more accurate detection and prediction of the onset of LoC in quadrotors, while effectively mitigating erroneous alarms (Type I errors) and missed warnings (Type II errors)?

- What role do Critical Slowing Down (CSD) indicators play in predicting the onset of a critical transition in quadrotors?
- How do actuator saturation levels impact the onset of a critical transition in quadrotors?
- In what ways can the integration of multiple parameters improve the robustness of LoC prediction, ensuring a lower rate of erroneous alarms (Type I errors) and missed warnings (Type II errors)?
- How can the model's predictive performance be evaluated, and what are the most relevant evaluation metrics?

RQ3 : To what extent can these LoC events be detected through parameters such as CSD and actuator phase lag?

- How do aggressive yaw, and roll maneuvers influence these parameters and contribute to LoC risk?
- Do different maneuvers exhibit consistent global trends in CSD, and actuator saturation that can be used for early LoC detection and prediction?
- How are these indicators affected by nominal flight?

1.3. Research Planning

The research strategy is outlined as shown in Figure 1.1, and can be categorized into different work packages:

- **WP1 : Preliminary Research & Literature Report** - This work package will involve investigating CSD applied to the pre-existing Yaw-LoC Data to identify important features, detrending methods, and the optimal window size for observing EWS using CSD parameters. This work package also includes the writing of the Literature Report
- **WP2 : Experimental Setup** - Focus will be on the production of the Gimbal, as well as assembling and tuning the quadrotor.
- **WP3 : Roll-LoC Data Collection** - This phase involves starting the collection of Roll-induced LoC Data.
- **WP4 : CSD Analysis & Thesis Report** - Apply CSD analysis to Roll-LoC Data, utilizing insights from WP1 about CSD parameter tuning. This will involve analyzing and comparing results from both LoC types. Additionally, the final thesis report will be written during this phase.
- **WP5 : Thesis Defense** - Implement the feedback received from the Green Light meeting and prepare for Thesis Defense Presentation.

In addition to the research strategy, an overall Thesis timeline can be seen in the next page. WP1 starts on 12/05/2023 with the preliminary research and literature report. Note that in the current planning, there is a 4-week summer break from July to August. Subsequently, WP2 begins on November 10, 2023 with the production and building of the experimental equipment. WP3 commences on January 1, 2024 with the Roll-LoC Data collection. WP4 starts on February 1, 2024 where the collected data will be processed and analyzed, and working in parallel, the final thesis report and the scientific paper will be written. Finally, WP5 begins on May 4, 2024 with the Green Light meeting, in which the remainder of the time will be used to implement final feedback, and to prepare for the thesis defense.

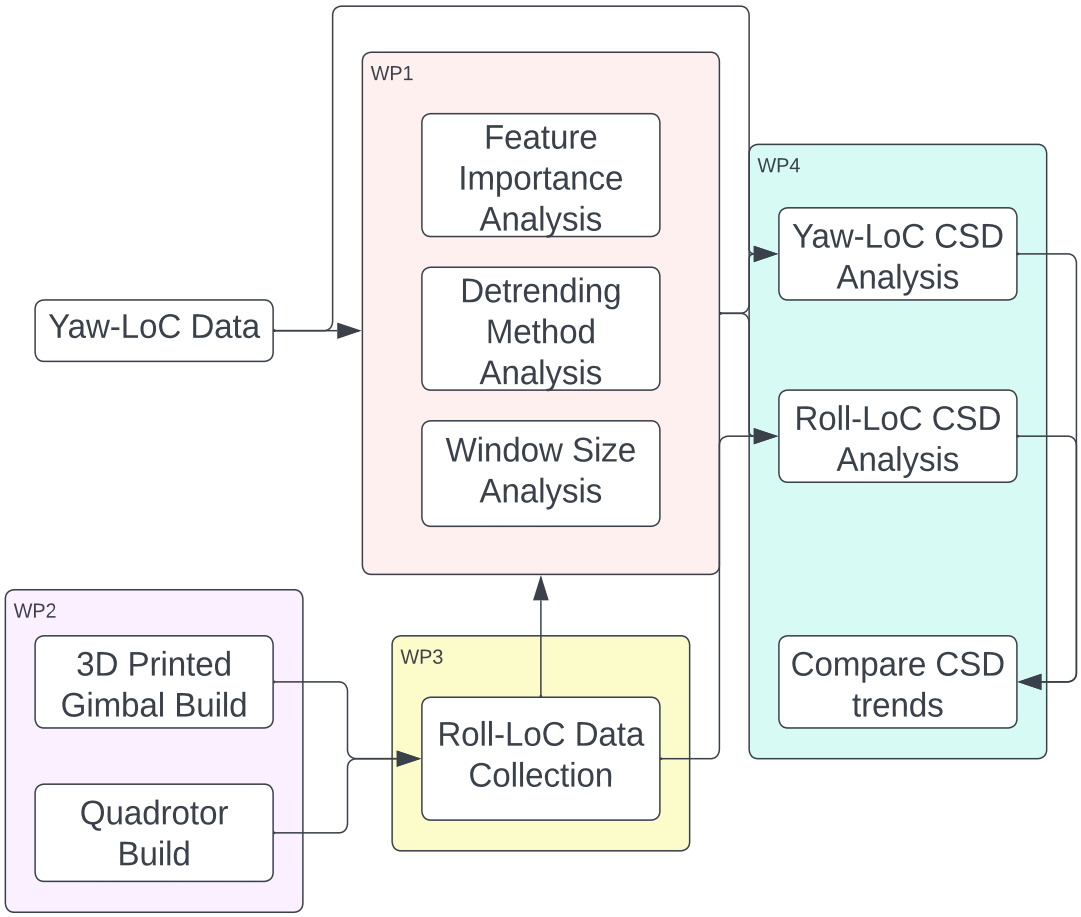

Figure 1.1: Research Strategy Visualization

The structure of the report is as follows. Firstly, Part I presents the scientific paper. Next, Part II presents the Literature Review. Finally, Part III presents the conclusions and recommendations. All code for this research is available in the GitHub repository [1].

[1] https://github.com/chumbuks/LoC_Detection_CSD-FLIS

Part I

Scientific Article

Loss-of-Control Detection of a Quadrotor Using Critical Slowing Down Theory

Chris Y.S. Chung*

Loss-of-Control (LoC) is the primary cause of drone crashes, necessitating efficient onboard prevention systems that are effective in terms of sensor requirements, computing power, and memory. This study introduces a data-driven approach for detecting LoC in quadrotors, using Critical Slowing Down (CSD) theory as an Early Warning Signal (EWS) of approaching a critical transition. This paper employs a Fuzzy Logic Inference System (FLIS) to aggregate the CSD metrics alongside other EWS indicators, such as actuator phase delay, to provide a fuzzy indicator that quantifies the quadrotor's stability. The proposed FLIS is applied to two LoC modes: the first is a yaw-induced LoC event during free-flight of the quadrotor in which growing off-axis instabilities during the maneuver culminate in LoC. The second is a roll-induced LoC event during a gimballed flight of the quadrotor in which growing off-axis instabilities during the maneuver also culminate in LoC. This approach proposes novel EWS indicators and a LoC detector and is generalizable across varying mass/size without needing precise state estimation of the quadrotor, instead only relying on onboard gyro and rotor speed data. Using real flight data from a GEPRO quadrotor, and a custom-built drone mounted on a 3-axis quadrotor gimbal testing rig, this paper demonstrates that various EWS indicators inferred with a FLIS can provide accurate, and timely detection of an upcoming LoC event, regardless of their specific causes or the maneuvers involved. This novel approach significantly enhances LoC detection rates relative to previous studies, and improves detection times, providing crucial additional seconds for corrective action.

I. Introduction

In recent years, quadrotors have gained significant attention due to their versatile applications across various domains. The commercial drone market, predominantly composed of quadrotors, hit a valuation of $19.89 billion in 2022, underscoring their growing prominence. Furthermore, projections indicate a continuous compound annual growth rate of 13.9% from 2023 to 2030 [1], highlighting the increasing demand for these devices. With the expanding user base, there is a pressing need for enhanced safety measures, particularly in situations where drones interact with people. Loss-of-Control (LoC) emerges as the predominant culprit behind quadrotor accidents [2].

Traditionally, LoC in aircraft has been extensively studied, with research primarily focusing on detecting LoC through the determination of the safe flight envelope (SFE), and defining LoC as a deviation from the boundaries of the SFE. Notable studies include Zhang *et al.* [3] [4], who developed a closed-loop flight envelope protection system for dynamic SFE retrieval using damage identification. Van Oort [5] identified the longitudinal SFE for an F-16 under various configurations and conditions. Lombaerts *et al.* [6] introduced an online SFE determination method accounting for model uncertainties in an impaired aircraft. Chongsival *et al.* [7] proposed a LoC prediction method by monitoring aircraft states across five flight envelopes. However, these methods face significant computational challenges, as estimating the SFE through standard Hamilton-Jacobi reachability analyses is known to be computationally intensive due to the curse of dimensionality [8].

For quadrotors, which operate on much tighter time scales, traditional SFE calculation methods are impractical due to these computational constraints. To address the computational constraints, Sun and de Visser [9] investigated the SFE determination for flight envelope protection using a computationally efficient Monte-Carlo approach. Additionally, Kaffa [10] investigated the behavior of the global SFE of a quad-rotor subjected to varying system dynamics, including the effects of longitudinal center of gravity displacements and actuator dynamics, also using a computationally efficient Monte-Carlo to estimate the SFE. However, using the SFE for LoC detection still has two main limitations. Firstly, a key assumption using SFE for LoC prevention is that it presumes that maintaining the existing system SFE is the sole strategy for preventing LoC. However, this overlooks the potential for a poorly designed controller to induce LoC

*MSc Graduate, Faculty of Aerospace Engineering, Kluyverweg 1.

even within the SFE. Secondly, any estimated SFEs emerge from a given (nominal) model of the system which fails to account for dynamic changes or faults in the system.

Consequently, much of the drone-specific research has shifted towards LoC detection and recovery in the presence of faults. Van der Pluijm [11] and Hoppenbrouwer [12] have led efforts in fault-based LoC detection, employing methods such as Critical Slowing Down (CSD) theory and Lyapunov stability analysis. However, results indicate that CSD alone was not suitable for LoC detection as it would result in too many false positives, and Lyapunov stability analysis required precise state-estimation, often necessitating external motion capturing devices such as OptiTrack.

In terms of LoC recovery in the presence of faults, Sun et al. [13] developed a multi-loop hybrid nonlinear controller for high-speed flight (9m/s) of quadrotors even with a single rotor failure (SRF). Additionally, they introduced an Incremental Nonlinear Fault-Tolerant Controller (FTC) for quadrotors with complete loss of two opposing rotors [14]. Hamadi [15] also made contributions to this field by developing a FTC for a UAV under hardware and software failures experiencing wind gusts. Lu et al. [16] designed an FTC that incorporates active fault detection and isolation, enabling real-world application of quadrotor control under rotor failure.

Despite these advancements, research into LoC in undamaged quadrotors remains sparse. One notable study by Altena *et al.* [17], tackled controller-induced LoC through a scenario-specific approach using neural networks for yaw-induced LoC detection and prediction. LoC is defined as the moment in time where the roll or pitch angle exceeds, and continues to exceed, 90 deg after the start of the yawing maneuver. The two limitations here were: firstly, the definition of LoC was very scenario specific, and does not apply in more general contexts where quadrotors perform maneuvers such as flips. Secondly, this attitude based LoC definition often failed to detect even if the dangerous yawing maneuver resulted in a crash, thus having a poor detection rate.

A gap exists in the current literature for a data-driven LoC detection in quadrotors that does not depend on the presence of hardware faults, is adaptable to various LoC scenarios, and remains computationally viable, without relying on precise state estimation of the quadrotor. Considering this gap, the research objective is to improve LoC detection in quadrotors using CSD theory, and actuator phase delay as an EWS, alongside a quantitative definition of LoC. This study aims to examine the roles of CSD, and actuator phase delay in detecting upcoming LoC events, and explore their integration using a Fuzzy Logic Inference System (FLIS) to enhance the accuracy and robustness of LoC detection while minimizing both false positives and negatives. Additionally, the study will assess how different quadrotors (systems) experiencing different modes of loss of control, such as yaw-induced vs roll-induced LoC maneuvers, impact these parameters, aiming to determine if consistent global trends exist across different systems.

The remainder of this paper is structured as follows: section II introduces CSD theory, and section III covers Fuzzy Logic. The methodology for data collection, signal detrending, Early Warning Signal (EWS) calculation, and the Fuzzy logic architecture are presented in section IV. section V presents the results from the CSD analysis and the Fuzzy Logic Inference of the CSD outputs, which are then discussed in section VI. The conclusions are presented in section VII.

II. Critical Slowing Down (CSD) Theory

The behavior of numerous natural and human-made systems is influenced by multiple parameters that evolve over time. Their responses to external perturbations may vary: they may exhibit a smooth, gradual change in the system's state, or they may show little reaction within certain ranges of conditions, responding more as they approach a critical threshold. This situation, where a small perturbation can lead to an abrupt shift in the system's state when it reaches a critical threshold is commonly referred to as *tipping* or *critical transition*. Tipping events can be observed in real-world systems like climate systems [18], ecological systems [19], financial systems [20], and biological systems [21].

Predicting these critical transitions presents significant challenges, as the system's state often exhibits minimal change prior to reaching the tipping point. Furthermore, models of complex systems typically lack the precision required for reliable predictions of where critical thresholds may occur. Interestingly, however, research indicates that various systems typically exhibit common symptoms as they approach a critical point [22].

In particular, several research papers on ecological systems [23][24][25] have emphasized the occurrence of *Critical Slowing Down* (CSD) in empirical/time-series measurements as they approach a tipping point. CSD theory is based on the generic phenomenon that, in the vicinity of various tipping points, the rate at which a system recovers from small perturbations becomes very low [23]. It is important to note that not all transitions can show signs of CSD. This can be visualized in Figure 1, where the transitions are grouped into Rapid Regime Shift (abrupt changes in the state and conditions of system behavior), Bifurcations (qualitative changes in system behavior due to the passing of a threshold in underlying parameters or conditions), and CSD (observed behavior of slow system response to perturbation).

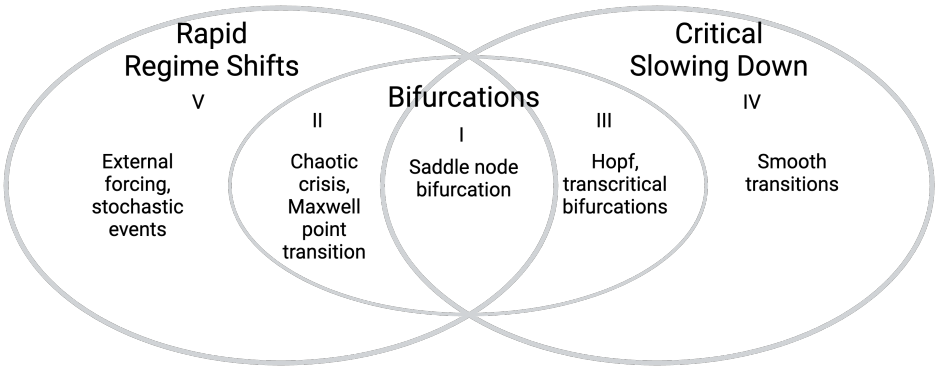

Fig. 1 Intersecting domains of Rapid Regime Shifts, Bifurcations, and Critical Slowing Down. Domain I represents catastrophic bifurcations preceded by CSD, Domain II describes catastrophic bifurcations not preceded by CSD, Domain III represents non-catastrophic bifurcations preceded by CSD, Domain IV represents CSD in the absence of bifurcations or regime shifts, and finally Domain V represents catastrophic regime shifts without bifurcations or CSD. Smaller labels under each domain illustrate specific phenomena occurring within these categories, adapted from [26].

Figure 1 illustrates that not all regime shifts include bifurcations, and bifurcations can exist without exhibiting signs of CSD, and some regime shifts can also exhibit signs of CSD. Consequently, CSD should not be interpreted as a universal predictor of tipping points nor specific to approaching tipping point. Rather, CSD should be regarded as a *broad spectrum* indicator of potential fundamental changes in the current state [23]. It is worth noting that much of the prior work on CSD has been centered around its application in open-loop systems. To the best of our knowledge, the successful application of CSD theory in closed-loop systems remains unexplored, opening new avenues for investigation.

To accurately apply CSD analysis, it is crucial to pre-process the data by detrending the signals. This process involves removing long-term trends or shifts from the time series data, thereby isolating the residual oscillations. Detrending is essential because it minimizes the influence of non-stationary trends on the analysis, allowing for a clearer detection of the slowing down phenomena that precede critical transitions.

Although CSD occurs for a class of different bifurcations (Figure 1), the following explanation will focus on the saddle-node (fold) bifurcation. This type is particularly intuitive to understand due to the *folded* nature of the system and the presence of distinct basins of attraction. To understand why CSD occurs at a fold bifurcation—where the system recovery behavior following a perturbation diminishes and approaches zero—consider the dynamics of a simple system:

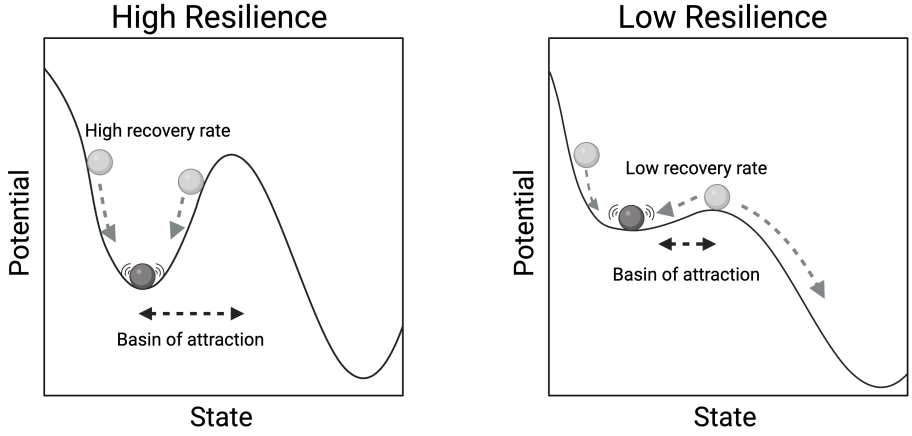

Fig. 2 Reduced system resilience due to Critical Slowing Down (CSD) in a saddle-node bifurcation. The rate of recovery from minor perturbations is lower when the basin of attraction is smaller, and thus lower resilience (right side) compared to when it is larger, and hence higher resilience (left side).

$$\frac{dx}{dt} = r + x^2 \tag{1}$$

In this model, r represents the bifurcation parameter. When r is negative, two equilibrium points exist: $\bar{x}_{stable} = \sqrt{-r}$ and $\bar{x}_{unstable} = \sqrt{-r}$, with one being stable and the other unstable, which can be visualized in the left side figure of Figure 2, where the left basin represents the stable equilibrium and right basin represents the unstable equilibrium. As r approaches zero from the negative side, the two fixed points move closer to each other. At $r = 0$, they merge and annihilate, resulting in a catastrophic bifurcation and an abrupt, irreversible change to a different state. This can be visualized on the right side of Figure 2, as r approaches 0, the system's resilience decreases, indicated by a smaller basin of attraction and the smaller slope of the basin, which effectively slows the rate of recovery.

Now, lets examine the dynamics near the tipping point, to analyze the stability of the equilibrium points, and how perturbations decay, linearizing the system round a fixed point x_0 by introducing a small perturbation ϵ such that $x = x_0 + \epsilon$:

$$\frac{d\epsilon}{dt} = \frac{d}{dt}(x - x_0) = \frac{dx}{dt} = r + (x_0 + \epsilon)^2 \tag{2}$$

Linearizing this equation using a first-order Taylor expansion yields :

$$\frac{d\epsilon}{dt} \approx r + x_0^2 + 2x_0\epsilon \tag{3}$$

Since x_0 is a fixed point, $r + x_0^2 = 0$, and the equation simplifes to:

$$\frac{d\epsilon}{dt} \approx 2x_0\epsilon \tag{4}$$

Here, the coefficient $2x_0$ is essentially the eigenvalue λ for the linearized system around the fixed point x_0. Now, as r approaches zero from the negative side, we can substitute x_{stable} into our expression for the eigenvalue:

$$\lambda = 2x_{stable} = 2(-\sqrt{-r}) \Rightarrow \lim_{r \to 0} \lambda = 2(-0) = 0 \tag{5}$$

Thus, as the system approaches a bifurcation, the parameter r approaches zero, and the eigenvalue associated with the stable fixed point approaches zero. Since the eigenvalue can be interpreted as the rate at which small perturbations to the system's state will grow or decay, this proves that as the system approaches a bifurcation, its rate of recovery will diminish.

The primary implication of CSD is that a decreased recovery rate following a small experimental perturbation may indicate the system's proximity to a bifurcation point [27]. Although it may be impractical to systematically monitor recovery rates after experimental perturbations, the fact that most natural systems regularly encounter such disturbances means that this can serve as an effective early warning signal for impending critical transitions. Notable indicators include increased lag-1 auto-correlation and variance in system fluctuations. This can be explained intuitively; CSD reduces the system's rate of change, causing its current state to more closely resemble its past and effectively increasing its *memory*, as measured by lag-1 auto-correlation [28] [29]. Additionally, as a tipping point approaches, the variance in the system's fluctuations increases due to the decreased rate of recovery. Because the effects of disturbances do not dissipate, there is an increase in the variance of the state variable [30].

This phenomenon can also be shown mathematically demonstrated using a simple first-order auto-regressive model, where repeated disturbances every Δt lead to a progressively slower return to equilibrium, characterized by a recovery rate λ:

$$x_{n+1} - \bar{x} = e^{\lambda \Delta t}(x_n - \bar{x}) + \sigma\epsilon_n \tag{6}$$

$$y_{n+1} = e^{\lambda \Delta t} y_n + \sigma\epsilon_n \tag{7}$$

y_n represents the deviation of the state variable x from its equilibrium, and ϵ_n is zero-mean noise with a standard deviation of σ. Assuming the recovery rate λ and the time interval Δt are independent of y_n, this setup can be effectively described as a first-order auto-regressive (AR(1)) process:

$$y_{n+1} = \alpha y_n + \sigma\epsilon_n \tag{8}$$

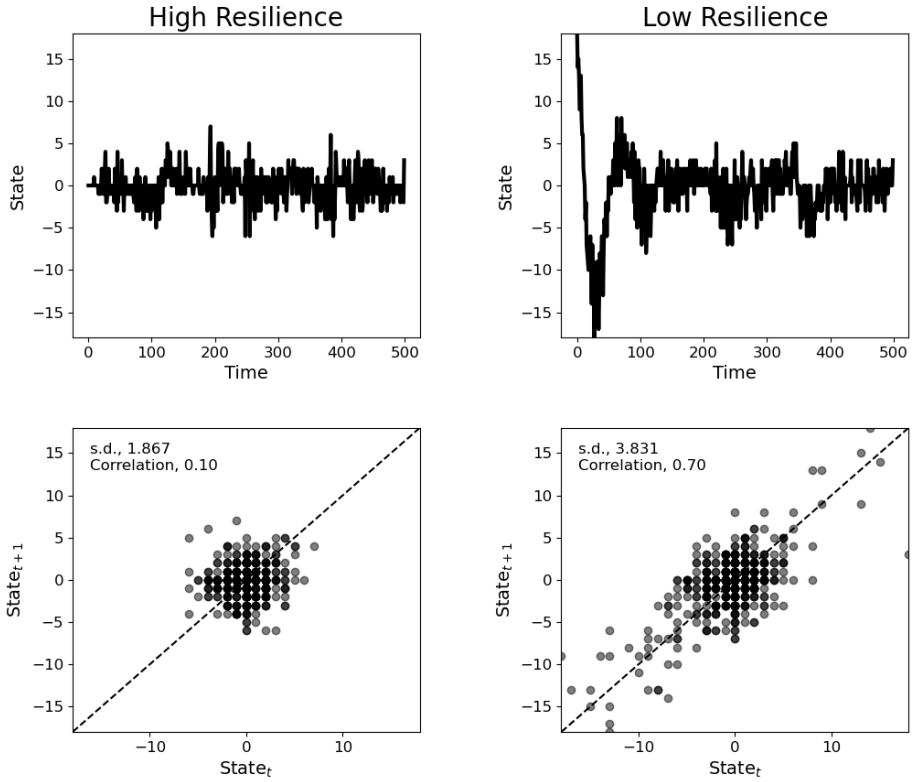

Fig. 3 The effect CSD measured in stochastically induced fluctuations in the state of the system as increased variance and "memory" as shown by lag-1 auto-correlation.

In an AR(1) process, the AR1 coefficient, denoted as $\alpha \equiv e^{\lambda \Delta t}$, quantifies how much each value in the series depends on its immediate predecessor. For white noise, this coefficient is zero, whereas for red (auto-correlated) noise, the coefficient is close to one [31]. The expected value of an AR(1) process $y_{n+1} = c + \alpha y_n + \sigma \epsilon_n$ is [32]:

$$E(y_{n+1}) = c + \alpha E(y_n) + E(\sigma \epsilon_n) \Rightarrow \mu = c + \alpha \mu + 0 \Rightarrow \mu = \frac{c}{1-\alpha} \tag{9}$$

can be understood as a linear combination of the previous value, y_n, adjusted by the coefficients c and α, with an added term ϵ_n for noise scaled by σ [32]. For $c = 0$, the mean equals zero and the variance is found to be :

$$Var(y_{n+1}) = E(y_n^2) - \mu^2 = \frac{\sigma^2}{1-\alpha} \tag{10}$$

Close to the critical point, the return recovery rate to equilibrium decreases, implying that λ approaches zero and the auto-correlation α tends to one. Thus the variance tends to infinity.

While CSD has traditionally been applied to ecological systems, van der Pluijm [11] attempted to explore the application of CSD in predicting critical transitions in quadrotors. He investigated whether EWS could be detected in the time-series data of a damaged quadrotor with a single rotor failure (SRF) using CSD indicators. The objective was to gather data on the steady-state and transient behavior of a damaged quadrotor experiencing various wind speeds, and to examine if there were EWS that could be detected prior to the quadrotor crashing as the wind speed was increased.

The analysis revealed that w_2, and q were the most relevant variables, as angles were found to be less likely to contain EWS, compared to angular rates as stochastic effects are less pronounced in angles. The study's concluded that using CSD theory alone to identify EWS in quad-rotor systems was not feasible due to the unrealistic constraints it placed on wind speed behavior. As the author noted, CSD requires slow and monotonic transitions, whereas damaged quadrotors usually are subject to sudden changes, resembling rapid regime shifts. Observations from Figure 1 indicate that some rapid regime shifts show no signs of slowing down, impacting the effectiveness of CSD based on the rate of

wind speed changes. However, by broadening the scope beyond traditional CSD indicators and incorporating other domain specific indicators such as actuator saturation metric, more reliable EWS could be generated. Furthermore, a sensitivity analysis showed that the most critical parameter is the size of the rolling window used, with larger windows yielding fewer false positives; however, for robotic applications, smaller windows are preferred.

This insight is consistent with a study by Wen *et al.* [33], which emphasizes the sensitivity of EWS detection to several factors, including the intrinsic early warning period (time-scale of LoC event, and window size), data frequency, and the choice of test statistic.

Another insightful example comes from a study by Dakos *et al.* [34], which delved into the potential EWS in a highly stochastic system using a model that describes the shift of a lake from an oligotrophic to a eutrophic state [35]. Initially, it was assumed that in such noisy conditions, CSD might not be relevant, and traditional EWS would not be expected to be effective. However, the research uncovered an unexpected connection. Even though this system did not align with the typical characteristics of a slow-time-varying system, but rather being characterized by strong perturbations leading to rapid regime shifts, or "flickering". The study revealed that the behavior of the system often exhibited rising variance, AR1, and skew. These patterns closely resembled the effects of CSD, challenging traditional assumptions and highlighting the adaptability of CSD Theory in unforeseen, highly stochastic environments. This unexpected finding underscores the versatility of CSD theory in identifying critical transitions, even in systems with dynamic, non-monotonic perturbations.

III. Fuzzy Logic Inference System

A Fuzzy Logic Inference System (FLIS) is the process of formulating the mapping from given input(s) to output(s) using fuzzy logic. Fuzzy Logic is based on the idea that truth can be expressed as a continuous *fuzzy* variable, not a binary output [36]. This approach may be particularly useful in scenarios like predicting the onset of Loss of Control (LoC), where a quadrotor's state might not strictly be binary; but could exhibit varying degrees of LoC.

FLIS is instrumental in various domains, including automatic control, data classification, decision-making, expert systems, robotics, and more, owing to its ability to handle nuances in data and decision-making processes [37]. The system comprises five primary components:

- **Rule Base** - A collection of fuzzy rules that use logical operators such as IF, OR, and NOT.
- **Database** - Defines membership functions for input and output variables.
- **Fuzzification Interface** - Converts crisp inputs into fuzzy inputs through degrees of match with linguistic variables (membership functions).
- **Decision-Making Unit** - Executes inference on the fuzzy rules.
- **Defuzzification Interface** - Transforms fuzzy inference results back into crisp outputs.

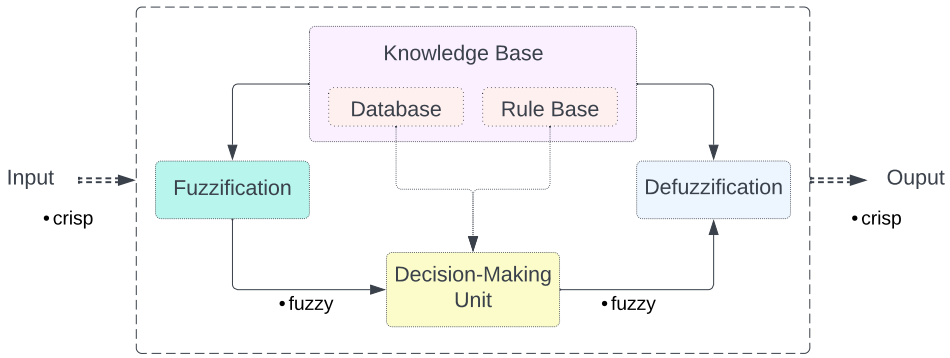

Fig. 4 Structure of a Fuzzy Logic Inference System (FLIS), with the five main components : Rule Base, Database, Fuzzification Interface, Decision-Making Unit, and Defuzzification Interface

The process begins with fuzzification, which transforms specific inputs into degrees of match with membership functions, based on predefined intervals. This transformation results in fuzzified variables, representing the antecedents in the

fuzzy logic system. These variables are then processed by the decision-making unit. This unit employs logical operators (*AND* for the minimum, *OR* for the maximum, and *NOT* for the complement of the membership value) to associate the antecedents' membership functions with a consequent membership function in the output. The choice and application of these logical operators critically influence the degree of match between the antecedents and the consequent, effectively determining the output's membership value. Upon applying all the rules, where each rule connects antecedents to a consequent, the decision-making unit aggregates these consequent values to produce a single, crisp output through defuzzification. This final step typically involves calculating the centroid of the aggregated fuzzy set, culminating in a crisp output value reflective of the fuzzy logic inference.

Fuzzy logic inference systems has various advantages, such as handling uncertainty, inherent interpretability, and most importantly domain expertise. If domain expert knowledge regarding quadrotor LoC are available, this can be leveraged to define appropriate membership functions and fuzzy rules that can reflect the intricacies of the system, such as combining CSD, and actuator phase delay parameters.

IV. Methodology

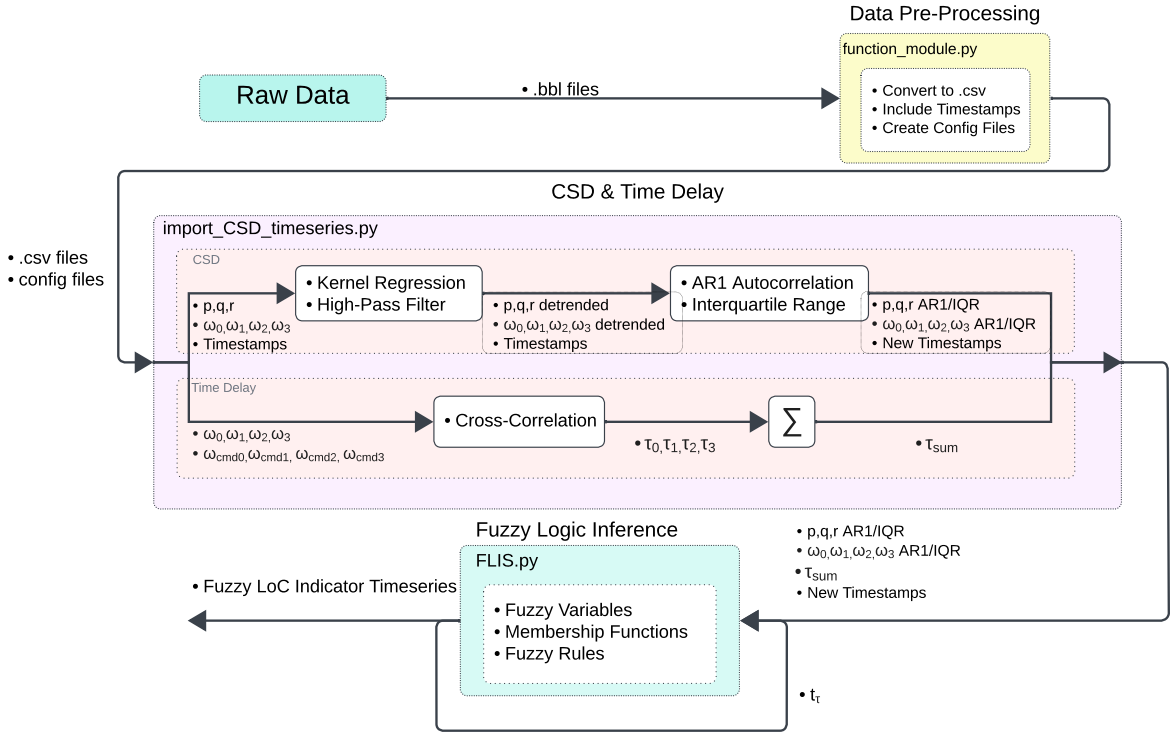

Fig. 5 CSD-FLIS Algorithm Pipeline

To accurately identify the occurrence of a LoC event, the proceeding sections will analyze the gyro rates (p,q,r) along with the measured rotor speeds ($\omega_0, \omega_1, \omega_2, \omega_3$) following the feature importance test from Van der Pluijn's study [11]. The proceeding sections will explain the different components of the pipeline, which includes data collection, signal detrending, EWS indicators, and the FLIS. Figure 5 presents the algorithm pipeline that will be used for the detection of LoC in quadrotors. Note that the algorithm pipeline uses some plotting and data-preprocessing functions from [38].

A. Data Collection

The primary cause of LoC in quadrotors is actuator saturation [39]. When actuators saturate, control authority deteriorates. Without control authority, the ability to counteract external disturbances is lost. Thus, the study will investigate two different modes of LoC induced by actuator saturation, which involve demanding very high roll and

pitch rates. Although both modes experience LoC due to a high rotational rates and consequently actuator saturation, the underlying LoC dynamics differ.

1. Yaw induced LoC

The two yaw datasets were sourced from the research of Altena *et al.* [17]. The datasets comprise two sets of Yaw-LoC Data. The first set is obtained from the GEPRO (CineGo), a quadrotor with a diagonal length of 361 mm and a mass of 265 g (Figure 6). The second set is obtained from the Tiny Whoop (DataCan), a more compact quadrotor with a diagonal length of 75 mm and a weight of 56 g (Figure 7). The details of these quadrotors are summarized in Table 1.

Fig. 6 GEPRO (CineGo) Quadrotor

Fig. 7 Tiny Whoop (DataCan) Quadrotor

The occurrence of a yaw induced Loss of Control (LoC) mode was attributed to a functionality in the BetaFlight flight controller software named "anti-gravity". This feature aims to maintain level flight by measuring the accelerometer's x and y axes. Nevertheless, in instances where the Inertial Measurement Unit (IMU) is not precisely aligned with the center of gravity (c.g.) of the drone, a pure yaw input results in the accelerometer also detecting translational motion. Consequently, the flight controller misinterprets these readings as disturbances in roll and pitch angles, endeavoring to counteract them, which leads to oscillations around the roll and pitch axes. At low rotational rates, or when the actuators are not saturated, these oscillations are manageable, and thus do not lead to LoC. However, at high rotational rates, when the actuators are therefore saturated, these oscillations become unstable. These oscillations tend to increase either due to phase discrepancies between the control actions and the measurements or as a result of the controller fighting itself.

Given that the CineGo dataset is substantially more extensive (96 LoC Flights, 71 System Identification (SysID) Flights) compared to the DataCan dataset (47 LoC Flights), the primary analysis will focus on the CineGo dataset, with the DataCan dataset serving to corroborate and validate the findings.

It is important to note that the yaw SysID flights include maneuvers that are coupled in roll, pitch, and yaw, thereby posing challenges in distinguishing which off-axis gyro-rates to apply the CSD analysis to. This issue will be revisited in subsection IV.D.

2. Roll Induced LoC

The collection of the roll LoC dataset will be conducted as part of this research. Previous attempts by Altena *et al.* [17] to collect roll LoC data were unsuccessful due to space limitations, which prevented the manifestation of LoC. This issue was re-confirmed during free-flight roll LoC maneuver tests conducted in the CyberZoo, a ten-by-ten meter drone testing laboratory at the Faculty of Aerospace Engineering, Delft University of Technology. It was observed that the quadrotor would impact the ground before entering a LoC state.

To address this challenge, a specialized gimbal test-bench (Figure 9), as designed by Sharma & Dijkstra [40], enables achieving a high roll-rate of ±800 deg/s. This required constructing a quadrotor with a custom frame to allow mounting onto the gimbal. The custom quadrotor and the gimbal build are displayed in Figure 8 and Figure 9. The details of the custom quadrotor are summarized in the third column of Table 1.

Table 1 GEPRO CineGo, TinyWhoop DataCan, and Gimbal Drone characteristics

Characteristics	GEPRO CineGO	Tiny Whoop DataCan	Gimbal Drone
Mass incl. batteries	265 [g]	56 [g]	138 [g]
Axis-to-axis diameter	361 [mm]	75 [mm]	117 [mm]
Propeller Diameter	76.5 [mm]	40 [mm]	40 [mm]
Motor	TC0803 with 15,000 [kV]	Emax Eco 1407 with 3300 [kV]	T-MOTOR M0802 with 25000 [kV]
Batteries	One 4S (four cell) Tattu R-Line with 14.8 [V] and 550 [mAh]	Two 1S (one cell) BETAFPV with 4.35 [V] and 300 [mAh]	One 1S (one cell) Tattu R-Line with 3.7 [V] and 500 [mAh]
Flight Controller	MATEKSYS F722-mini 2-8S	JHEMCU SH50 F4 2S with 8.0 MB black box	T-MOTOR F411 6A AIO - ELRS
FC software	Betaflight 4.2	Betaflight 4.2	Betaflight 4.2

Fig. 8 Custom Quadrotor (Gimbal Drone)

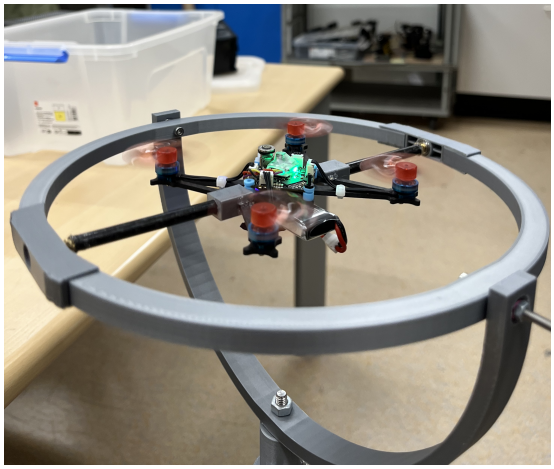

Fig. 9 3D printed Gimbal

During data collection, the quadrotor is deliberately placed in a LoC scenario by demanding a roll rate of ±800 deg/s. Initially, the quadrotor rotates about its x-axis as intended; however, due to quadrotor coupling [41], the rolling motion also introduces yawing, which later induces pitch oscillations. As the LoC maneuver continues, the yaw rate increases. However, in the absence of input for yaw, the gyroscopic sensor detects a gradual increase in yaw rate errors. As the yaw rate drift increases, the error grows until the yaw rate surges to 1000 deg/s, leading to rotor saturation. Subsequently, BetaFlight kills the integrator (I) gain for yaw to prevent integrator wind-up, resulting in the quadrotor ceasing to roll and losing control entirely, unable to rotate around its x-axis.

Addressing the challenge encountered with SysID Yaw data, specifically the coupled maneuvers in SysID flights, the SysID data for Roll-LoC has been segregated into three distinct sets: Roll, Pitch, and Yaw. This separation of maneuver types enables the discrete analysis of gyro AR1 values pertinent to each maneuver.

B. Signal Detrending

CSD applied to a closed-loop stochastic and complex system such as a quadrotor poses numerous challenges. It is important that CSD is applied to signals that can capture the rate of recovery to external perturbations. In the context of quadrotors, this could mean analyzing the oscillations in roll and pitch axes when a yaw maneuver is performed. To

obtain meaningful results from CSD analysis on quadrotors, it is imperative that the long term trends in the data are removed (detrended) to isolate the fluctuations in the residuals. If this is not done properly, the remaining correlations (trends or periodicity in the time series) can lead to false indications.

Detrending is done by finding a function to approximate the signal's long-term trends. Once this can be captured, these long-term trends may be removed from the original signal. The result is a detrended signal, which only contains the residual fluctuations due to external disturbances. During the preliminary phase of the research, four detrending techniques were evaluated: Moving Average, Differencing, Kernel Regression, High Pass Filter. Detailed outcomes of this evaluation are presented in the Appendix A.1.

The investigation revealed that the gyro rate data yielded the most effective EWS through CSD when detrended using Kernel Regression. Conversely, the measured rotor speed is best detrended using a High-Pass Filter.

Kernel regression is a non-parametric technique in statistics used to estimate the conditional expectation of a random variable; it allows the modeling of complex relationships between variables without assuming a predefined form of the relationship. Kernel regression can be used for detrending by fitting a smooth curve to the time-series data, capturing the long-term trends, which is subsequently subtracted from the original time series to obtain the detrended data. In this study, a local constant Gaussian kernel detrending method is used, as widely acknowledged in literature [31] [42][43], the Gaussian kernel function at the point $(t_i - t)$ adjusted by the bandwidth h is formulated as shown in Equation 11:

$$K_h(t_i - t) = \frac{1}{\sqrt{2\pi}h} e^{-\frac{(t_i - t)^2}{2h^2}} \tag{11}$$

$$\hat{y}_t = \frac{\sum_{i=1}^{n} K_h(t_i - t) y_i}{\sum_{i=1}^{n} K_h(t_i - t)} \tag{12}$$

Here, $(t_i - t)$ is the distance in time between the data point t_i and the estimation point t. The kernel function K_h uses this distance to compute how much influence the data point y_i should have on the estimation at t. h represents the bandwidth of the kernel, which influences the kernel's width. In essence, the computation of the kernel function results in a weight that decreases exponentially as the distance $(t_i - t)$ increases, more rapidly so as the bandwidth h decreases. The bandwidth influences the smoothness of the trend line, a larger bandwidth resulting in a smoother estimate, while a smaller bandwidth captures more local fluctuations. The choice of bandwidth is critical; as it must be adequate to remove long-term trends while preserving residual dynamics.

Each data point y_i at a time t_i is then multiplied by this weight, and the sum of those weighted values forms the numerator of the kernel regression estimate. The denominator is the sum of the weights, normalizes these weights to ensure that the weights sum to 1. The entire equation for the kernel regression estimate $(\hat{y}_t)$ can be seen in Equation 12. By aggregating all $\hat{y}_t$ values into a time series, a time-series which contains only the long-term trends (trend-line) is generated.

Similarly, the High-Pass Filter (HPF) has been utilized to remove the long-term trends (low frequency oscillations), as indicated by references [44] [45]. This filter is represented in the frequency domain by the transfer function $H(s)$ given by Equation 13

$$H(s) = \frac{s^n}{s^n + \sum_{k=1}^{n} c_k s^{n-k}} \tag{13}$$

Here, $s = j\omega$ represents the complex frequency, with ω being the angular frequency of the signal, c_k the coefficients determined by the filter design, and n the filter order. Now the actual implementation of the filter requires a discrete time filter, which approximates the continuous time filter in Equation 13.

$$y[n] = \frac{1}{a_0}\left(\sum_{k=0}^{M} b_k \cdot x[n-k] - \sum_{i=1}^{N} a_i * y[n-i]\right) \tag{14}$$

To achieve this, the filter is applied in the time domain using pre-calculated coefficients designed to mimic the desired frequency domain effects. This typically involves a two-step process where a Butterworth filter first determines the necessary coefficients a_i, b_k, that define the filter's behavior. Once the filter coefficients have been obtained, the filter is applied directly to the discrete-time signal, through the use of a difference equation as shown in Equation 14. Here, $y[n]$ is the current filtered output, $(\sum_{k=0}^{M} b_k \cdot x[n-k])$ is the input signal component, which computes a weighted sum of the current and past input samples, and the coefficients b_k defines the filter's zeros. $(\sum_{i=1}^{N} a_i * y[n-i])$ is the feedback component, which calculates a weighted sum of the past filtered outputs, and the coefficients a_i set the poles.

M and N represent the filter orders, which indicates the number of b and a coefficients, respectively, and n indicates the current sample index. In this study, a fifth-order HPF will be utilized. The bandwidth and the cutoff frequency used for the detrending methods can be obtained in Table 2.

Table 2 Detrending Parameters

Data	CineGo		DataCan		Gimbal	
Variables	Gyro Rates	Debug	Gyro Rates	Debug	Gyro Rates	Debug
Bandwidth/Cutoff	0.01 [-]	7 [Hz]	0.004 [-]	5 [Hz]	0.015 [-]	15 [Hz]

C. Early Warning Signals

1. CSD indicators

The two CSD indicators that are used in this paper are the lag-1 auto-correlation coefficient (AR1) and the inter-quartile range (IQR). Note that instead of the variance, this study uses IQR as the measure of variability, since variance is sensitive to extreme values. Thus, IQR is selected to provide a consistent measure of variability across the time-series data [46] [47] [48].

The AR1 is the most commonly used indicator to measure CSD. An increase in auto-correlation usually refers to an increase in short-term memory (correlation at low lags) of the system, which indicates that the rate of return/recovery following a perturbation has been reduced.

To approximate AR1, the correlation between consecutive values (i.e., its lag-1 value) in a time series is computed using a moving window offset by a single time step (step size). The auto-correlation is computed as follows:

$$R_{XY} = \frac{\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2}\sqrt{\sum(Y_i - \bar{Y})^2}} \tag{15}$$

where X_i and Y_i are the i^{th} data point in windows X and Y respectively. $\bar{X}$ and $\bar{Y}$ are the means of the data within windows X and Y, and N is the window size. To use AR1 as an EWS indicator, two hyper-parameters are tuned: the window size and step size.

The selection of window size critically influences the false positive rate of EWS given by CSD. Opting for a large window size allows for a more smooth, and general capture of the persistent effects of AR1 peaks, at the expense of delayed detection. Opting for a smaller window allows for faster detection but increases the risk false positives. An optimal window size is determined heuristically by testing various window sizes on a subset of the data to find the minimum window size that minimizes false positives without compromising detection time, which was found to be 450 as detailed in Table 2. The step size, less critical than the window size, was set at 1 to maximize the detection of EWS.

An important consideration is the impact of the bandwidth/cutoff frequency used in the detrending method, which can sometimes yield a *Not a Number* (NaN) value for the AR1. To address this, a checking mechanism has been implemented to detect if the resulting AR1 time-series contains any NaN values, prompting an incremental adjustment to the bandwidth/cutoff and a subsequent recalculation of the AR1 time-series in an iterative manner. The algorithm for this iterative detrending and AR1 calculation can be found in the Appendix A.2.

The IQR is the second CSD indicator that will be used. The IQR is calculated as the difference between the third quartile (Q3) and the first quartile (Q1). $Q3$ represents the value below which 75% of the data falls, and Q1 represents the value at which 25% of the data falls. The method for calculating these quartiles and consequently the IQR is given by :

$$\text{IQR} = Q3 - Q1 \tag{16}$$

The data is first sorted in ascending order. This is a fundamental step because percentiles are values that divide the sorted dataset into parts. To find the percentile value Q3 and Q1, the following formula can be used to find to position/index k of the p-th percentile in a dataset with N samples:

$$k = \frac{p}{100}(N + 1) \tag{17}$$

If k is not an integer, the percentile value lies between two data points. NumPy uses linear interpolation to estimate this value. For the IQR, the only hyper-parameter that can be tuned is the window size. From initial analysis, it was found that IQR is not as sensitive to false positives occurring, therefore a smaller window size of 50 was chosen heuristically. The bandwidth, cutoff, and window size parameters used can be obtained in Table 2.

Additionally, recall from section II, CSD theory posits that as a system approaches a bifurcation, its eigenvalues tend to zero, implying that the rate of recovery goes to infinity. However, a quadrotor is not a linear system and does not have globally defined eigenvalues. In our approach, given that we are analyzing very short time scales (0.5 seconds for the AR1 windows), the quadrotor dynamics may be considered approximately linear, thereby allowing us to implicitly linearize the system around a moving operating point. This adaptation is critical in applying CSD theory to the complex dynamics of quadrotors effectively.

2. Time Delay indicator

Van der Pluijm's research [11] recommends the implementation of case-specific measures to address the challenges of false positives and the limitations of CSD in detecting EWS. Building on this, our study hypothesizes that LoC often arises from phase lags due to actuator and sensor saturation, particularly under high rotational demands. To effectively monitor these phase lags, the Time Delay Indicator (τ), is introduced as a case-specific measure. The cross-correlation between two signals can be computed using:

$$R_{xy}(\tau) = \sum x[n] \cdot y[n+\tau] \tag{18}$$

where $x[n]$ and $y[n]$ are the measured and commanded RPM values, τ is the lag, or shift applied to the signal y relative to x. The cross-correlation is calculated for various lag values in a predefined window of 50 time-steps.

Choosing an optimal window size is pivotal; a smaller window size provides higher resolution and quicker response to data changes but is more susceptible to noise, potentially leading to misleading correlations. Conversely, a larger window might detect irrelevant changes due to external factors like throttle adjustments. From a preliminary sensitivity analysis, it was determined that an optimal window size of 50 provides a good balance between fast detection, and minimizing false positives.

The lag that yields the highest correlation is identified and recorded as the time-delay coefficient τ. To translate this lag into real-time units, dividing by the sampling rate yields the τ in seconds. Subsequently, the τ values from all four rotors are summed to have a single indicator of phase delays, offering insights into the cumulative impact of phase lags across multiple rotors.

This unified τ contributes to constructing another indicator, t_τ, which quantifies the prolonged detection of lags within the time series. It serves as a temporal metric to gauge the duration and magnitude of the lag, employing an algorithm that increments a counter based on the presence and magnitude of the lag, resetting after 50 consecutive lag-free time-steps. The methodology for calculating the t_τ variable is detailed in the Appendix A.2. The strategy for determining the increment magnitude balances the reduction of false positives while maintaining a threshold low enough to facilitate early detection.

D. Fuzzy Logic Architecture

Various methods exist for interpreting results from the CSD analysis. Previous research focusing on applying CSD to quadrotors [11], utilized the Kendall Rank correlation coefficient to evaluate the statistical significance of EWS identified via CSD. This method, however, falls short in enabling an integrated analysis combining the outputs from multiple parameters.

To address this limitation, this paper adopts a Fuzzy Logic Inference System (FLIS) for inferring the CSD outputs. While there are various tools available for constructing a FLIS, this paper utilizes the SciKit Fuzzy Logic Toolbox (Version 0.4.2) for Python (Version 3.11.5). Leveraging this toolbox, the main effort is primarily focused on the database and rule base construction.

The Database's first step involves setting the expected value ranges for both antecedent and consequent variables, allowing the assignment of membership functions to specific value ranges. For example, AR1 values are naturally constrained between [-1, 1], easing their fuzzification process. Nevertheless, the IQR's lesser sensitivity to outliers, coupled with fuzzy logic's unique feature of allowing overlaps between membership functions enhances the system's threshold-setting flexibility, which aids in establishing a practical analysis range. Membership functions are then designated to these variables, sorting them into linguistic categories such as 'Stable', 'Mid', 'EWS', and 'LoC', based on

their value ranges. Notably, these membership functions can take shapes such as triangular or trapezoidal, popular for their simplicity and capacity to depict qualitative distinctions clearly [49]. Note that the choice of membership function and the overlap between them greatly determines the shape of the Fuzzy Logic output values, and will be further discussed in subsection V.C. Furthermore the details regarding each membership functions' shape, threshold, overlap can be found in Appendix A.3

Simplicity remains a key objective in our data-driven approach. From initial analysis, it was observed that for simple uncoupled flights, Gyro AR1 was sufficient at detecting LoC. However, for coupled maneuvers like those in the Yaw-LoC SysID flights, Gyro-AR1 rules are inapplicable due to the stick neutrality condition (which will be elaborated on further below), we need to increase the complexity of the FLIS by including other CSD indicators such as Gyro IQR, measured rotor speed AR1, and Time delay indicator (tau).

Therefore, this study employs two main rule groups for inference: Gyro-Rate AR1 based rules (GYRO-AR1), and CSD/Time Delay (CSD-τ) based rules. The Gyro-AR1 rules are the primary set of rules to identify a LoC scenario, only inferring the AR1 time-series of gyro rates, when no stick inputs have been given for the past 450 time steps. This condition of stick neutrality over 450-time-step is due to the nature of a rolling window analysis for generating AR1 values. As the AR1 is calculated on the previous 450 time-steps, the impact of a stick input may persist within the AR1 calculation window. The CSD-τ rules serve as an auxiliary measure to enhance LoC detection rates and address the limitations of the stick neutrality condition when using Gyro-AR1 rules. Since the GYRO-AR1 rules are contingent on the absence of stick inputs, the CSD-τ rules provide a supplementary indication of stability when GYRO-AR1 rules are inapplicable, such as during maneuvers that involve stick inputs in all three axes. These rules utilize Gyro IQR, measured rotor speed's AR1, and the time delay duration variable t_τ to assess the quadrotor's risk of LoC. Overall, the study employs three FLIS models: a Gyro-AR1 based model, a CSD-τ based model, and a combined model. The combined model gives priority to Gyro-AR1 rules, but switches to CSD-τ rules when the Gyro-Ar1 is inapplicable due to violations of the stick-neutrality condition.

Within these rule groups, rules are categorized into Stable (STA), Mid Stable (MID), Early Warning Signal (EWS), and Loss-of-Control (LoC). STA rules determine the quadrotor's stability under normal conditions. MID rules assess the stability during nominal maneuvers. EWS rules are indicative of diminishing stability potentially leading to an LoC, while LoC rules establish the threshold for identifying when a quadrotor is undergoing an LoC event.

Comprehensive details on each variable, their membership functions, and the fuzzy rules are available in Appendix A.3. Once all rules have been applied, the corresponding output membership function values can be combined to determine the overall aggregated output membership value.

V. Results

The flight data were first analyzed using CSD, proving AR1 and IQR outputs for gyro rates, and measured rotor RPMs. These outputs were then used as inputs to the FLIS, as shown in Figure 5. The following sections will discuss the results obtained from the CSD and FLIS, separately for the two different LoC scenarios.

A. CSD Results - Yaw LoC

Figure 10 and Figure 11 present the CSD analysis conducted on the Gyro Rates for the Yaw-LoC (CineGo) datasets.

From Figure 11 it is evident that gyro rates do exhibit a decrease in rate of recovery to external perturbations, as AR1 and IQR increase as the LoC event is approached.

Additionally, the detrended signals also reveal minor oscillations in roll and pitch prior to the yaw maneuver. Although these oscillations appear minimal in the detrended series, the AR1 outputs highlight a significant increase due to these small movements.

This is reflective of the sensitivity of the CSD indicators to the chosen detrending method and subsequent hyper-parameters. This observation points to the sensitivity of CSD analysis towards the chosen detrending method, emphasizing the importance of accurately removing all long-term trends to avoid false positives. Adjusting the bandwidth could be a solution, but this risks eliminating valuable residual data. An innovative approach involves incorporating stick inputs into the CSD inference. When stick inputs for roll, pitch and yaw remain at a neutral ($1500[\mu s]$) position for over 450 time steps, a drop in AR1 is noted. This can be observed in the orange overlay on top of the AR1 time series in Figure 11. This is a binary indicator for stick input neutrality over the last 450 time-steps : a '0' indicating absence of stick inputs, and '1' indicating presence. Hence, gyro rate AR1 values should only be considered when no stick input has been registered in the preceding 450 time steps. Implementing this criterion facilitates the use of the AR1 indicator as an informative CSD, provided the detrending method adequately eliminates (most) long-term trends.

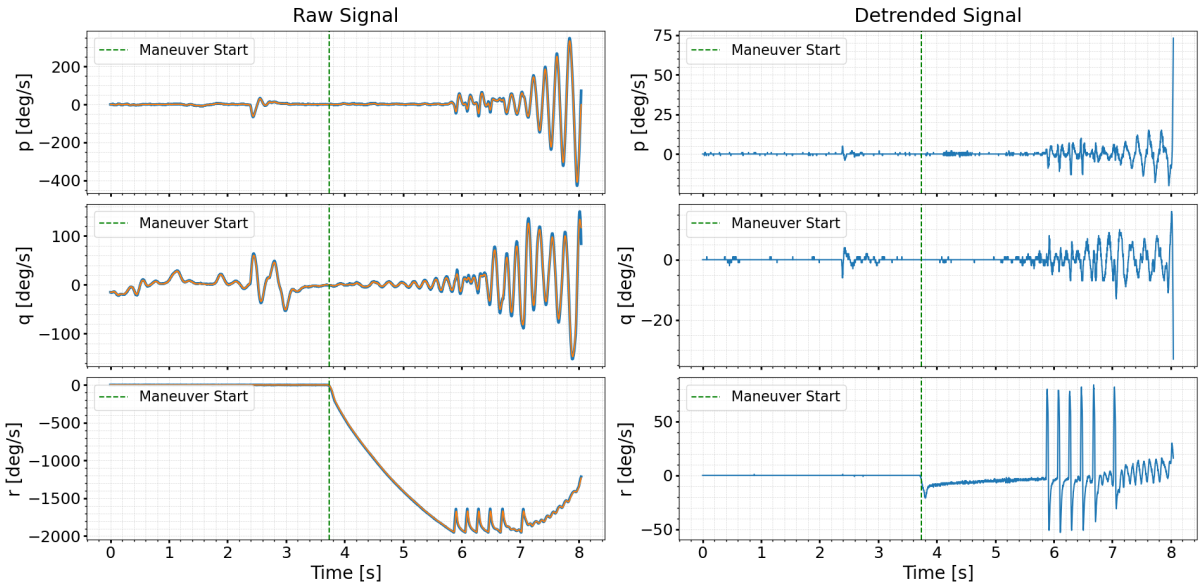

Fig. 10 Yaw-LoC (CineGo) gyro rates detrending results. Each row denotes a rotational rate axis, the first column depicts the original signal in blue, overlaid with the estimated long-term trend in orange. The second column depicts the detrended signal. Both column features a green dotted line indicating the commencement of the LoC maneuver.

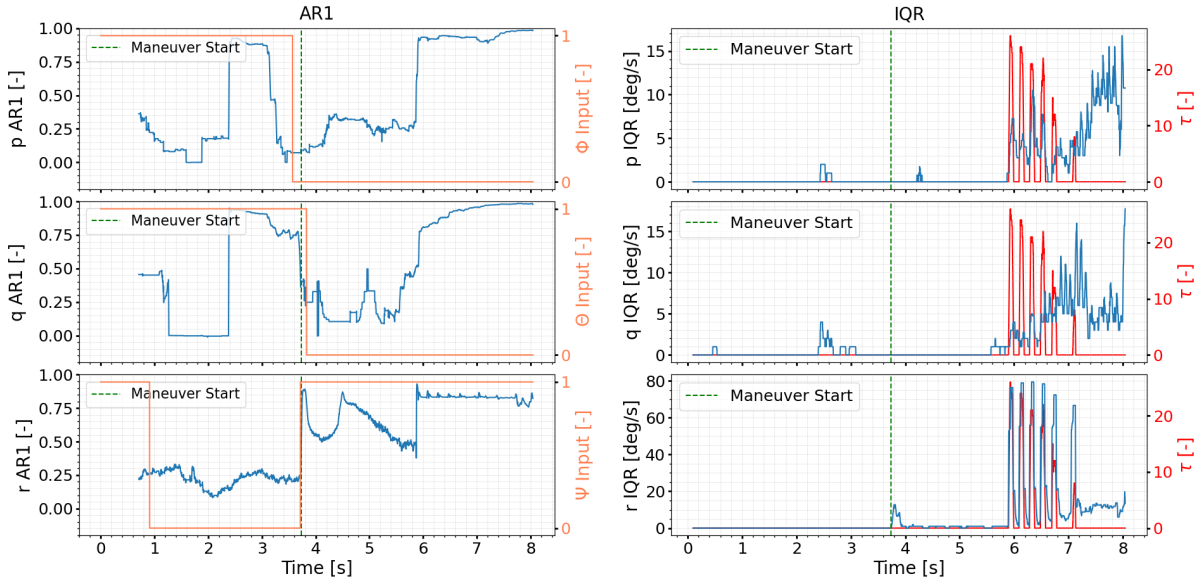

Fig. 11 Yaw-LoC (CineGo) gyro rates CSD results. Each row denotes a rotational rate axis, the first column depicts the AR1 time-series in blue, overlaid with the stick neutrality condition in orange. The second column depicts the IQR time series in blue, overlaid with the lag indicator signal τ. Both column features a green dotted line indicating the commencement of the LoC maneuver.

IQR also appears to exhibit EWS, utilizing a smaller window and showing less sensitivity to the detrending method. Unlike AR1, where minor oscillations in roll and pitch lead to sharp increases, IQR remains stable until the onset of LoC, highlighting its effectiveness in EWS detection. However, the main limitation of IQR is its inconsistent range. Unlike AR1, which has a defined range of $[-1, 1]$, IQR exhibits a variable range, making the setting of precise thresholds prior to observing the LoC maneuver challenging without comprehensive understanding of its dynamics during the LoC maneuver. Thus in this research, AR1 is prioritized as the primary indicator due to its fixed range and direct correlation

with LoC events, and the IQR serves as a supportive measure combined with measured rotor speed's CSD results, to enhance accuracy by reducing the likelihood of false positives.

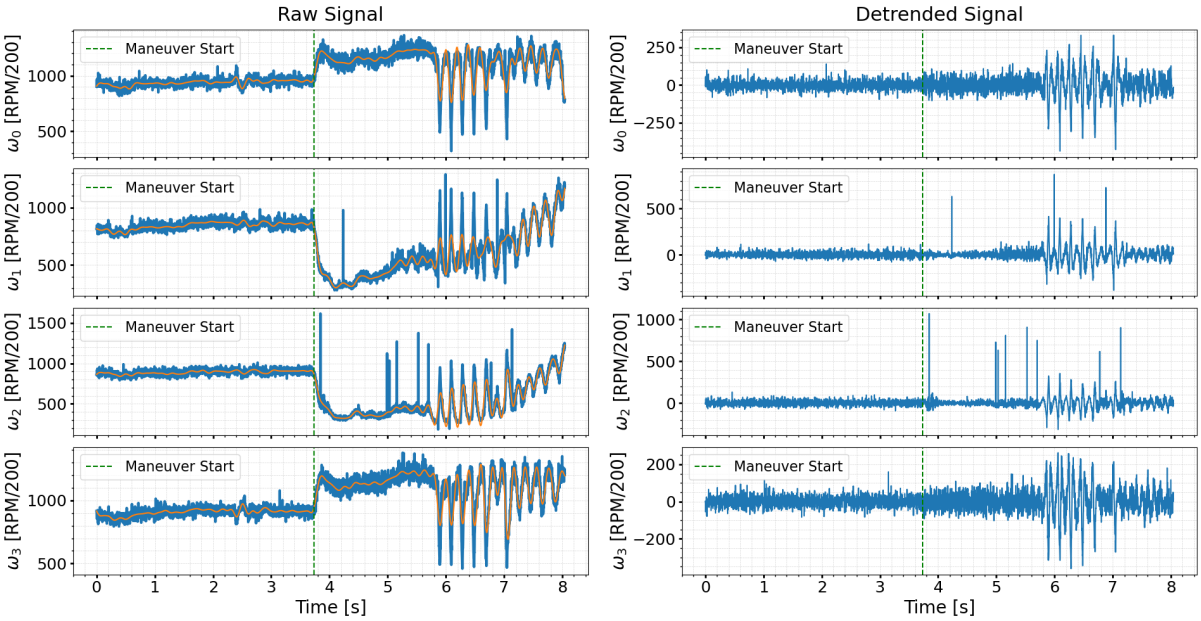

Fig. 12 Yaw-LoC (CineGo) measured rotor RPMs detrending results. Each row denotes a rotor, the first column depicts the original signal in blue, overlaid with the estimated long-term trend in orange. The second column depicts the detrended signal. Both column features a green dotted line indicating the commencement of the LoC maneuver.

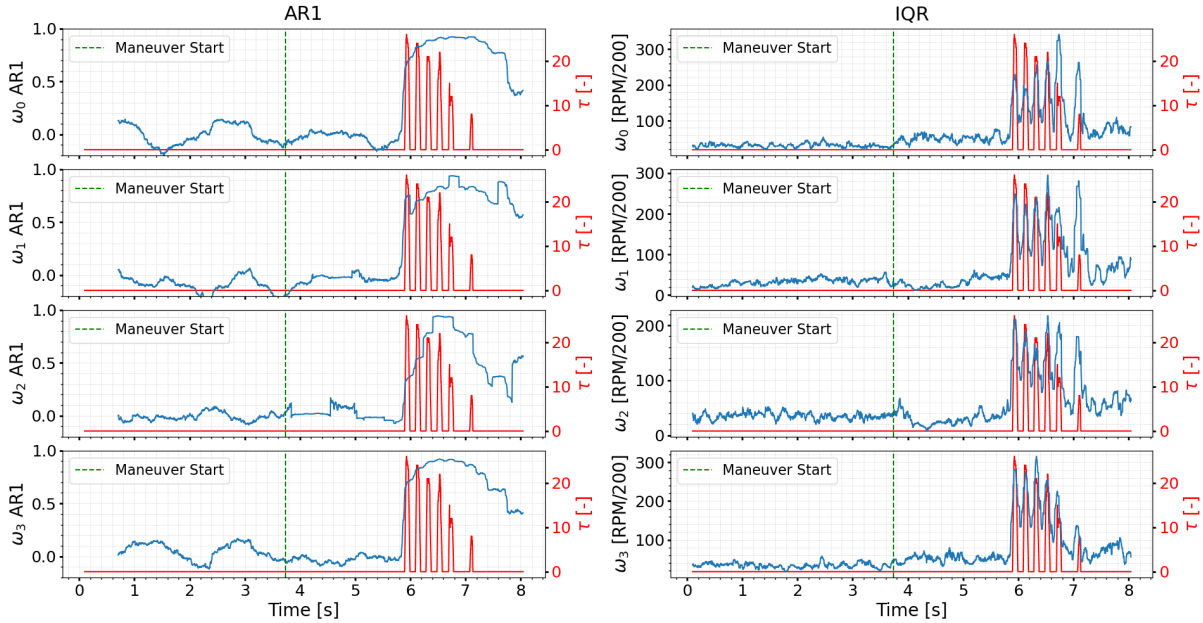

Fig. 13 Yaw-LoC (CineGo) measured rotor RPMs CSD results. Each row denotes a rotor, the first column depicts the AR1 time-series in blue. The second column depicts the IQR time series in blue. Both column features a green dotted line indicating the commencement of the LoC maneuver, and a red line showing the lag indicator signal τ.

The measured rotor speed's CSD analysis illustrated in Figure 12 and Figure 13 reveals these signals also exhibit an increase in AR1 as a LoC event is approached. While measured rotor speed values can be considered for LoC detection, their suitability as an EWS is less clear. This is primarily because, the increase in AR1 tends to be more abrupt. Consequently, this study gives precedence to gyro rates as the main indicator for EWS due to their more gradual increase in AR1. measured rotor speed values, on the other hand, are utilized as supplementary indicator to improve LoC detection rates.

The trends in Figure 11 suggest a sensitivity to non-LoC maneuvers. Thus, Figure 14 presents the CSD analysis on the SysID flights, revealing two significant findings. The first observation concerns AR1 spikes during non-LoC maneuvers. Similar to the LoC flights, the detrending approach appears insufficient in completely eliminating long-term trends. In addition, the SysID flights includes maneuvers that were coupled in roll, pitch, and yaw, resulting in elevated AR1 values for all axes. Implementing the strategy of ensuring stick neutrality over the preceding 450 time-steps as a means to filter out false positives, as demonstrated by the orange overlay in the third column of Figure 14, helps in mitigating this issue. Moreover, the IQR values during SysID flights do not increase as significantly as observed during LoC flights. This smaller increase in variability is also aids in distinguishing between false positives in nominal and LoC maneuvers.

The second observation regards the absence of phase lag in the nominal SysID Flights, supporting the utility of the actuator phase lag indicator. This consistency in lag detection, correlating with increasing AR1 noted in LoC flights (Figure 11 and Figure 13). Moreover, the lag relates the system dynamics to traditional notions of stability, such as phase margin and delay margin, thus providing a robust measure against false positives. These two key results lend credibility to the actuator indicator.

Nonetheless, a notable deviation emerges when applying the time-delay indicator to the DataCan flights, marked by an absence of detected lag during LoC events. This absence of lag suggests a potential link between the actuator size and τ, suggesting that larger (and slower) actuators would inherently exhibit higher latency. The CSD outputs for the DataCan flights can be found in the Appendix A.4. A similar behavior was observed in the Roll-LoC, and will be elaborated further in subsection V.B.

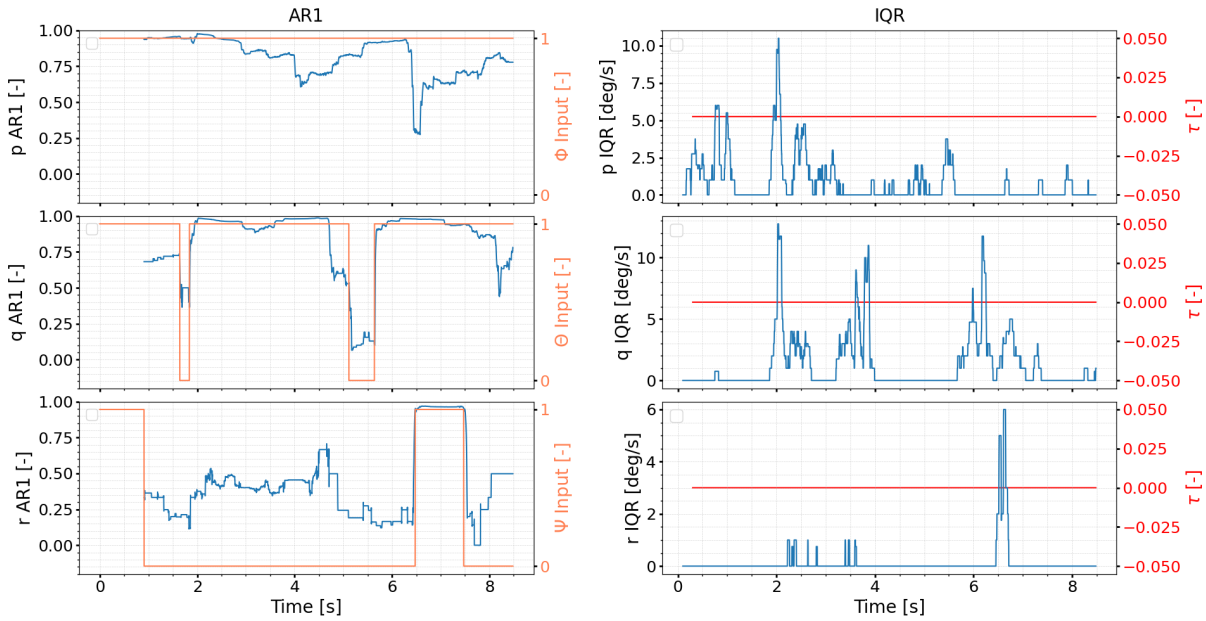

Fig. 14 Yaw-SysID (CineGo) gyro rates CSD results. Each row denotes a rotational rate axis, the first column depicts the AR1 time-series in blue, overlaid with the stick neutrality condition in orange. The second column depicts the IQR time series in blue, overlaid with the lag indicator signal τ in red.

B. CSD Results - Roll LoC

Figure 15 and Figure 16 show the gyro CSD analysis from a Gimballed-LoC flight-test, demonstrating a more consistent AR1 increase compared to the Yaw-LoC CSD outcomes, without detecting false positives before the maneuver. This difference is primarily due to the quadrotor's gimbal setup, which eliminates the need for attitude corrections, or the impact of minor perturbations on the gyro rates. Additionally, the absence of detected phase lag supports the insights regarding actuator size and detectability of phase lag (described in subsection V.A).

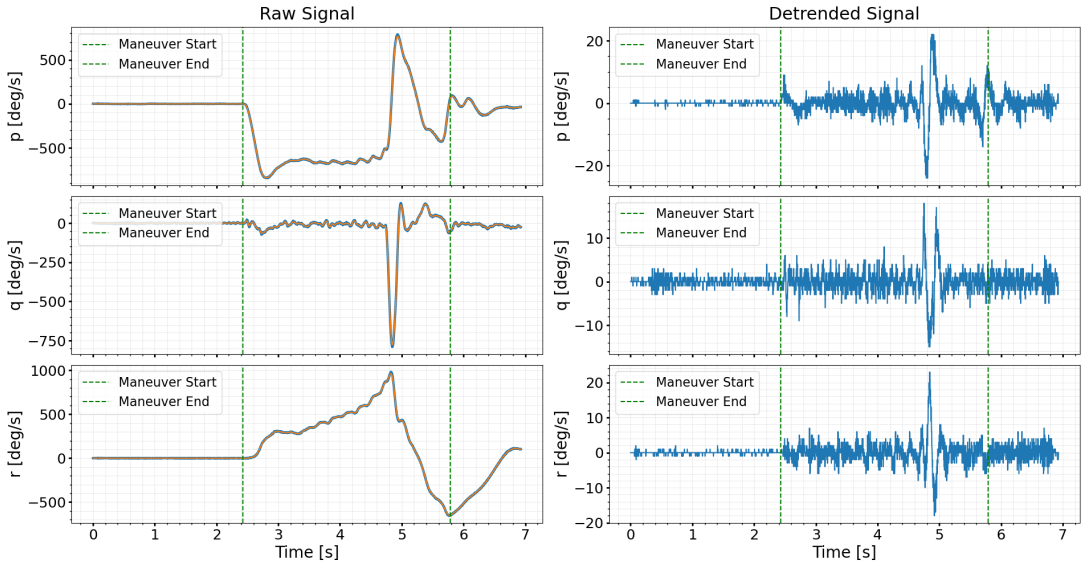

Fig. 15 Roll-LoC gyro rates detrending results. Each row denotes a rotational rate axis, the first column depicts the original signal in blue, overlaid with the estimated long-term trend in orange. The second column depicts the detrended signal. Both column features a green dotted line indicating the beginning and the end of the LoC maneuver.

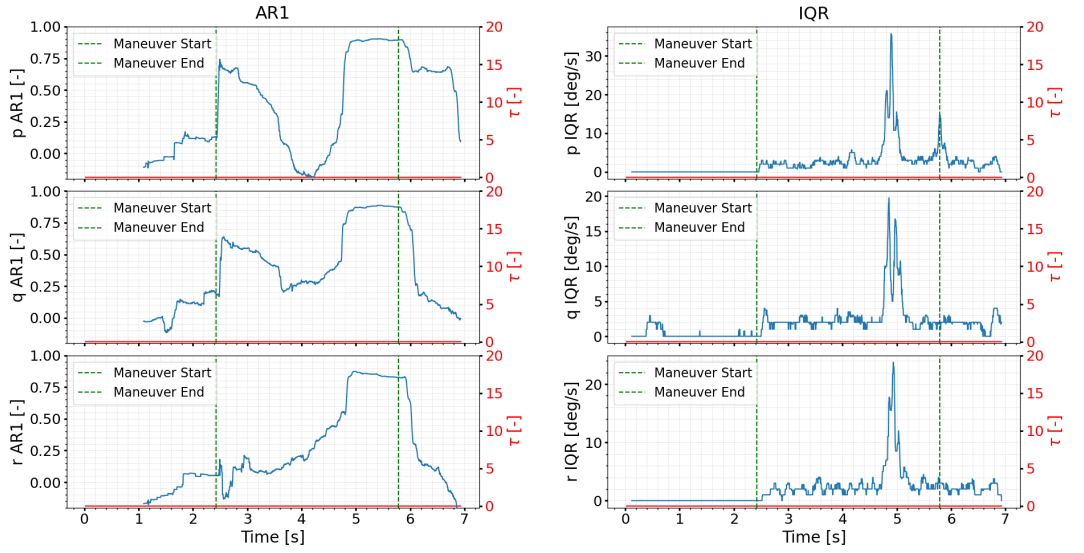

Fig. 16 Roll-LoC gyro rates CSD results. Each row denotes a rotational rate axis, the first column depicts the AR1 time-series in blue. The second column depicts the IQR time series in blue. Both column features a green dotted line indicating the beginning and the end of the LoC maneuver, and a red line depicting the lag indicator signal τ.

Similarly to the Yaw-LoC scenario, there is a consistent pattern where AR1 initially increases before experiencing a

decrease shortly after the start of the LoC maneuver. This pattern is visible in the AR1 time series for pitch and yaw rates, shown in the second and third rows of Figure 16, respectively, suggesting coupling effects. In the Roll-induced LoC, these coupling effects appear more pronounced, as illustrated by the detrended time series in the second column. There is a significant coupling effect between the roll and yaw axes, initially maintaining a steady yaw rate that begins to escalate after one second. This escalation in yaw rate further intensifies pitch oscillations, leading to a peak three seconds after the maneuver onset. Following this peak, the flight controller kills the integrator (I) gain for yaw to prevent integrator wind-up, effectively halting the rolling maneuver, as evidenced by the drop in roll rate in Figure 15. The distinctions in LoC dynamics between roll and yaw will be explored in greater detail in subsection VI.A.

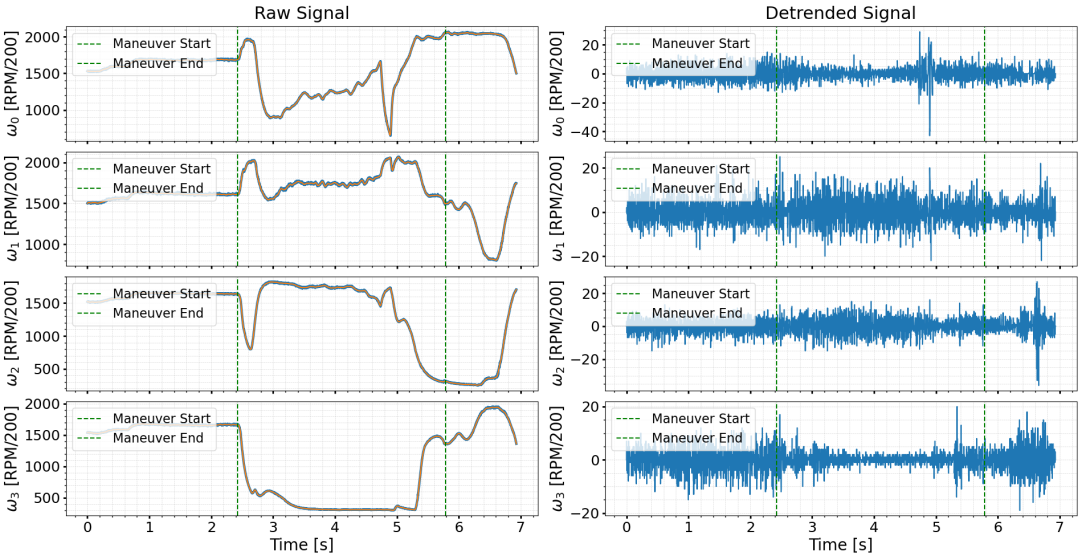

Fig. 17 Roll-LoC measured rotor speed's detrending results. Each row denotes a rotor, the first column depicts the original signal in blue, overlaid with the estimated long-term trend in orange. The second column depicts the detrended signal. Both column features a green dotted line indicating the beginning and the end of the LoC maneuver.

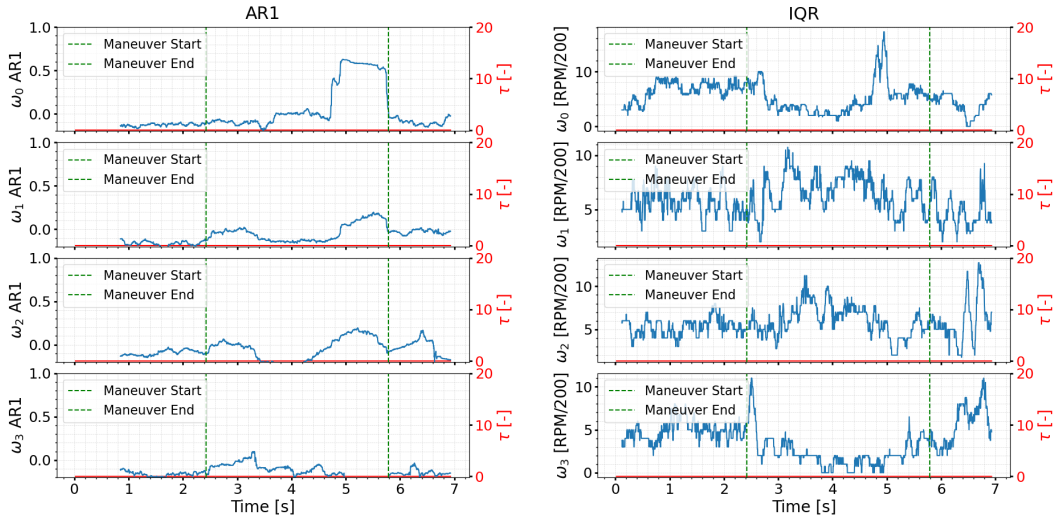

Fig. 18 Roll-LoC measured rotor speed's CSD results. Each row denotes a rotor, the first column depicts the AR1 time-series in blue. The second column depicts the IQR time series in blue. Both column features a green dotted line indicating the beginning and the end of the LoC maneuver, and a red line depicting the lag indicator signal τ.

Figure 17 and Figure 18 presents the detrending and CSD results on the measured rotor speeds. Unlike the Yaw-LoC scenarios, the measured rotor speed in Roll-LoC does not effectively reflect LoC. This discrepancy may stem from the nature of Roll-LoC, compared to Yaw-LoC. In Yaw-LoC, the controller typically induces oscillations as a corrective measure, resulting in rapid throttle changes. However, in Roll-LoC, the oscillations arise not from the controller actions, but from the coupling between roll and yaw. Consequently, there are fewer throttle changes in the roll LoC data, which means that the rotor speeds do not accurately reflect LoC. Additionally, compared to the Yaw-LoC, Roll-LoC uses a higher cutoff frequency (f_c) of 15$[Hz]$, which may lead to the removal of some residual oscillations during detrending.

Figure 20 and Figure 22 present the CSD analysis for the nominal rolling SysID maneuver. It is important to note that the AR1 time series begins during the maneuver, making the high initial correlations in the pitch rate unsurprising. The CSD results demonstrate its ability to distinguish false positives in nominal and LoC maneuvers, evidenced by the pitch and yaw AR1 values rising only to $[0.7, 0.4]$ respectively. For the IQR, observations show that in the SysID flights, gyro rates show little variability, as reflected by lower IQR values relative to those in the LoC flights. The CSD results on the gyro rates highlights it's effectiveness in distinguishing between false positives in nominal and LoC maneuvers.

Additionally, while the measured rotor speeds do not exhibit any increase in AR1, their IQR values seem to increase similarly to those observed in the LoC flights. This suggests that the IQR of measured rotor speeds may not be an adequate indicator of LoC.

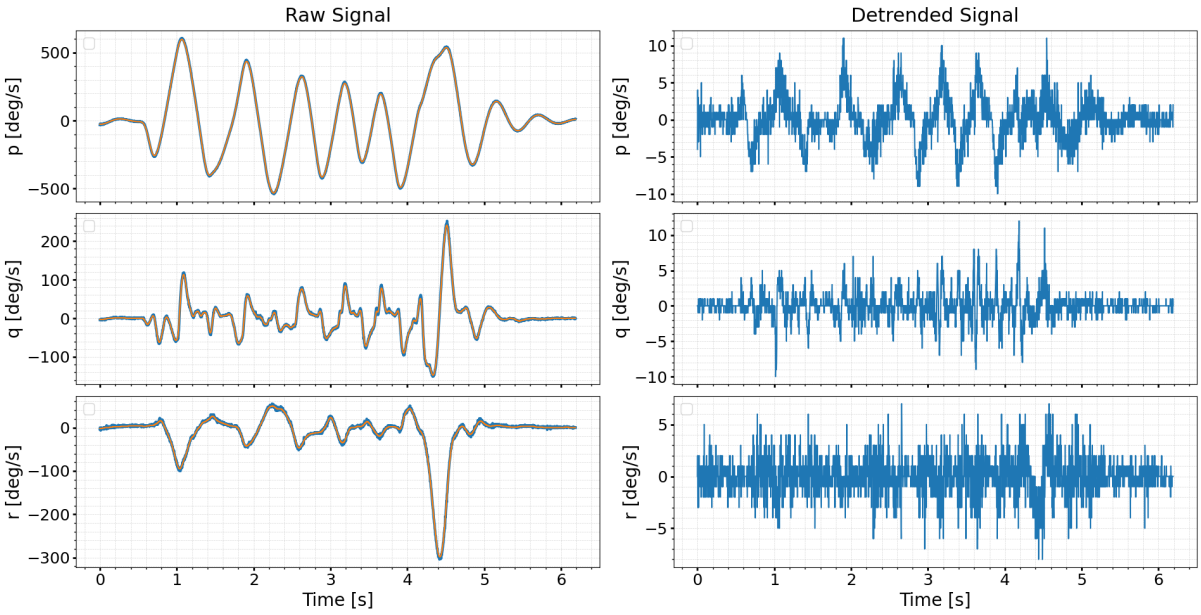

Fig. 19 Roll-SysID gyro rates detrending results. Each row denotes a rotational rate axis, the first column depicts the original signal in blue, overlaid with the estimated long-term trend in orange. The second column depicts the detrended signal.

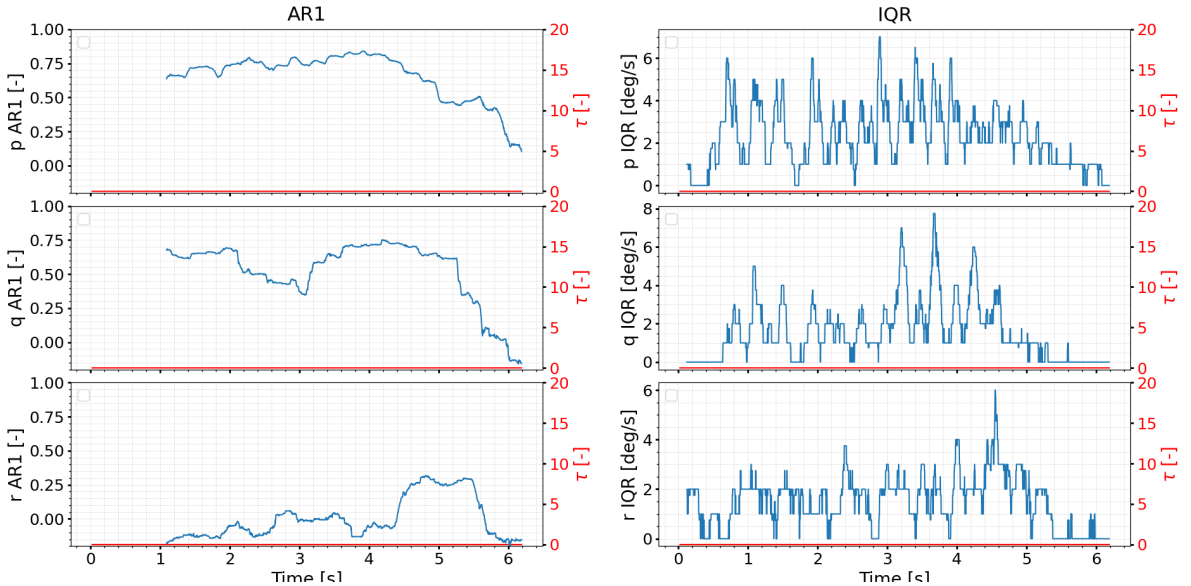

Fig. 20 Roll-SysID gyro rates CSD results. Each row denotes a rotational rate axis, the first column depicts the AR1 time-series in blue. The second column depicts the IQR time series in blue. Both column features a red line depicting the lag indicator signal τ.

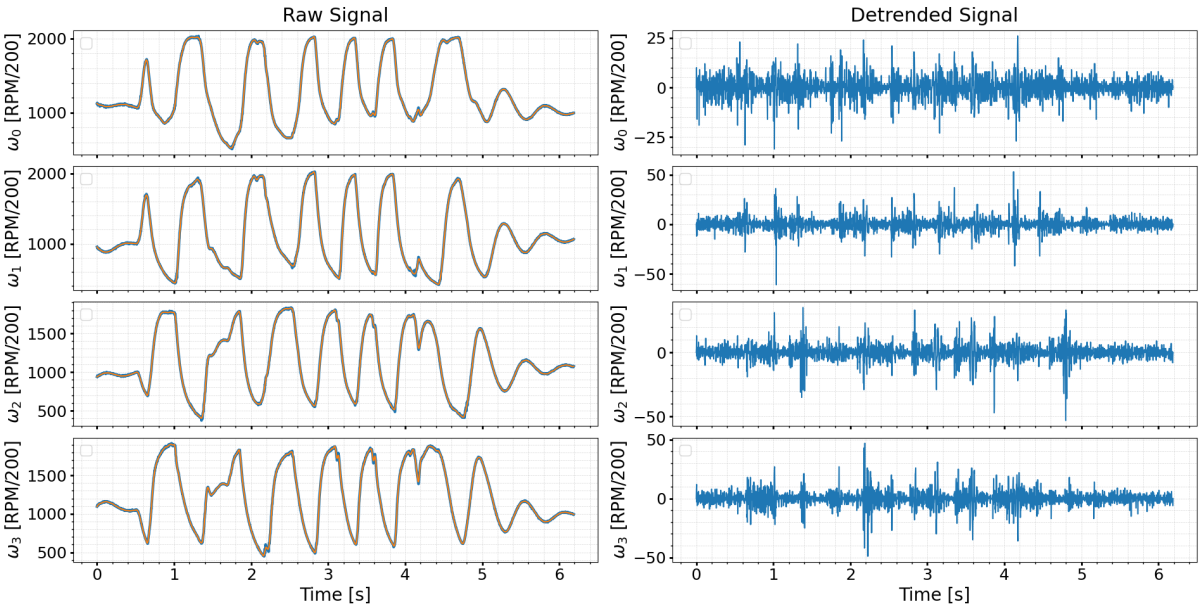

Fig. 21 Roll-SysID measured rotor speed's detrending results. Each row denotes a rotor, the first column depicts the original signal in blue, overlaid with the estimated long-term trend in orange. The second column depicts the detrended signal.

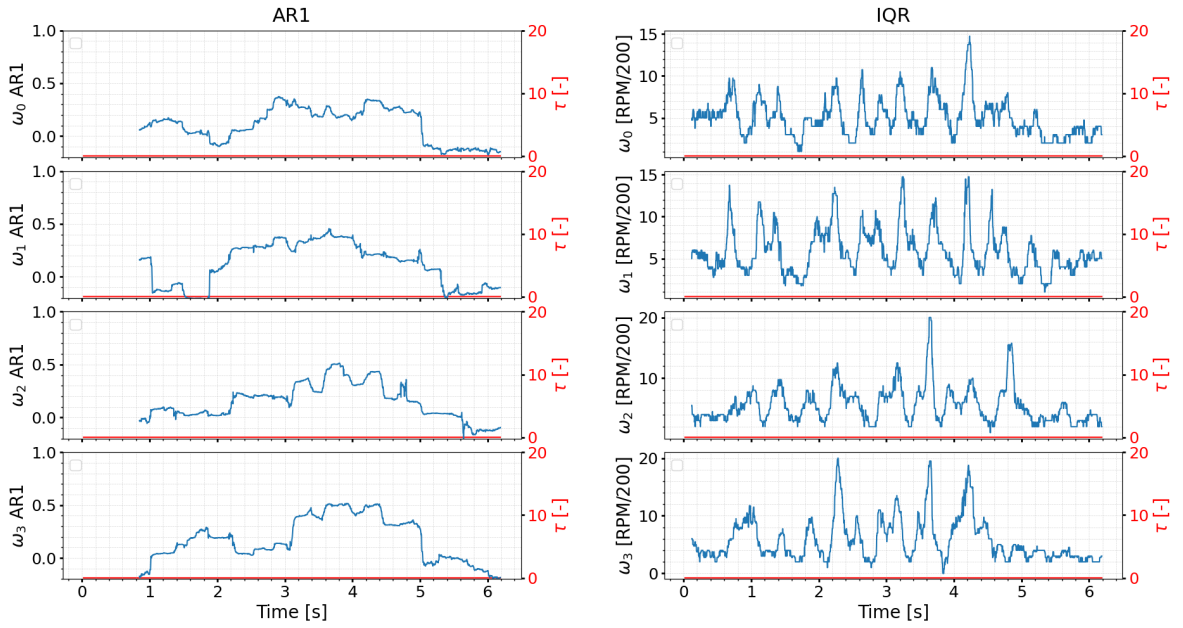

Fig. 22 Roll-SysID measured rotor speed's CSD results. Each row denotes a rotor, the first column depicts the AR1 time-series in blue. The second column depicts the IQR time series in blue. Both column features a red line depicting the lag indicator signal τ.

C. FLIS Results - Yaw LoC

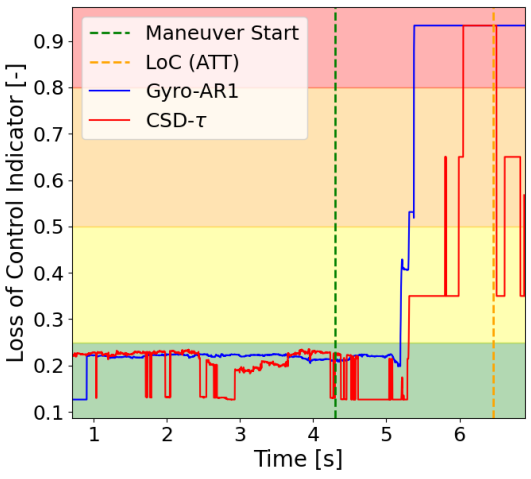

Fig. 23 Fuzzy LoC Indicator for Yaw-LoC (CineGo). The color-coded regions indicate stability and risk levels: green (stable), yellow (mid-level stability), orange (early warning signal), and red (LoC). The maneuver initiation is marked by the vertical green dashed line. The blue line depicts the Fuzzy LoC Indicator based on Gyro-AR1 rules, while the red line is based on CSD-τ rule set. The onset of ATT-based LoC detection is indicated by the vertical orange dashed line.

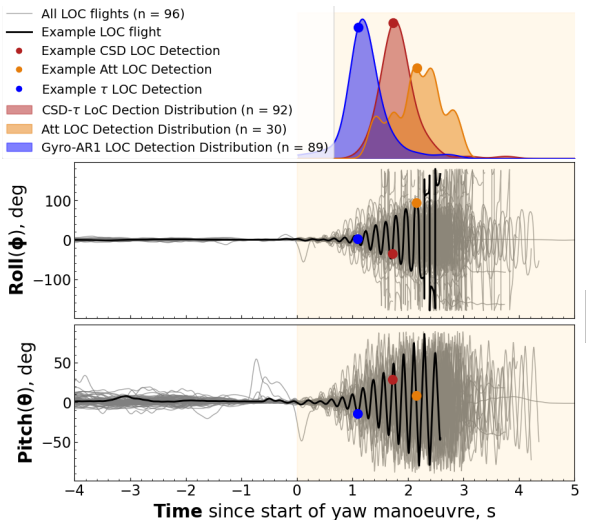

Fig. 24 LoC detection method comparison for Yaw-LoC (CineGo). Displayed are: Gyro-AR1 (blue), CSD-τ (red), and attitude-based definition of [17]. The first row visualizes LoC detection timing relative to the start of the maneuver. The second and third row overlay roll and pitch responses of 96 flights in grey, with a sample LoC flight in black. Colored dots indicate the moment LoC was detected by each method.

Figure 23 displays the FLIS output, demonstrating the FLIS' capability to quantify the quadrotor's proximity to an upcoming LoC event. Three distinct LoC definitions are depicted:

1) **Gyro-AR1 based LoC detection** (Blue) - LoC defined as the moment Gyro AR1 exceeds 0.8, provided that the stick-neutrality condition is met.
2) **CSD-τ based LoC detection** (Red) - LoC defined as the moment t_τ exceeds 50.
3) **ATT based LoC detection** (Orange) - LoC defined as the time when roll or pitch angle exceeds, and continues to exceed, 90 degrees after the start of the yawing maneuver[17]

As expected, prior to the maneuver, the LoC indicator remains in the stable (green) region. Upon initiation of the maneuver, the indicator swiftly transitions into the mid-stable (yellow) region, progressively increasing until it reaches the EWS (orange) region, and finally to the LoC (red) region. Generally, the Gyro-AR1 method exhibits earlier detection compared to the CSD-τ based method.

This demonstrates the FLIS's ability to effectively interpret multiple indicators, as shown by the CSD-τ method, which integrates Gyro IQR, measured rotor speed AR1, and time delay duration t_τ. This combination of indicators reduces the likelihood of false positives (FP) and improves missed detection (false negatives).

Using the Gyro-AR1 based LoC detection, the quadrotor transitions from a stable to an LoC state in 1.09 seconds from the onset of the yaw maneuver. The CSD-τ based definition identifies LoC after 1.72 seconds from the maneuver's start, and the Attitude based definition identifies LoC after 2.15 seconds from the maneuver start.

Additionally, Figure 24 compares detection times across all 96 (CineGo) LoC flights for the three different LoC definitions. These results reveal two main advantages of the Gyro-AR1 and CSD-τ based LoC definitions over the ATT (Attitude) based definition. Firstly, a significant improvement in detection rate is observed, with the ATT-based definition achieving a 31% detection rate, while the Gyro-AR1 and CSD-τ based LoC definitions achieve detection rates of 93% and 96%, respectively. The lower detection rate of the attitude-based approach is attributed to its reliance on a 90-degree attitude criterion, which is often unmet before a drone crash.

Secondly, both the Gyro-AR1 and CSD-τ based methods offer earlier LoC detection by an average of 0.97 seconds and 0.4 seconds, respectively, compared to the ATT-based definition. Considering the rapid dynamics of quadrotors, this additional time for recovery is crucial, potentially making the difference between a drone's recovery and crash. This extension of available recovery time, as well as a high detection rate, underscores the effectiveness of CSD as an EWS indicator.

A notable constraint of the Gyro-AR1 based rules is their sensitivity to the detrending methodology used. To counteract this, a stick-neutrality condition was integrated when applying the Gyro-AR1 rules. Nevertheless, this approach is challenged by SysID flight data, involving nominal maneuvers that engage stick inputs across all three axes, rendering the Gyro-AR1 rules inapplicable. To address this limitation, the FLIS was employed to combine both the Gyro-AR1 with the CSD-τ based rules, thereby enabling its application to the SysID data. The combined model gives priority to Gyro-AR1 rules, but switches to CSD-τ rules when the Gyro-Ar1 is inapplicable due to violations of the stick-neutrality condition. These results are summarized in Table 3

Subsequent observations from this combined method indicated a 99% detection rate and an 8% false positive rate, underscoring the effectiveness of integrating both rule sets within the FLIS framework for enhanced LoC detection.

Table 3 LoC Detection Time Distribution Comparison between Gyro-AR1 versus CSD-τ versus ATT based LoC Definitions

Gyro-AR1				**CSD-τ**				**ATT**			
Mean	Var	TP	FP	Mean	Var	TP	FP	Mean	Var	TP	FP
1.205 [s]	0.027 [s^2]	93%	8%	1.764 [s]	0.032 [s^2]	96%	8%	2.172 [s]	0.197 [s^2]	31%	0%

D. FLIS Results - Roll LoC

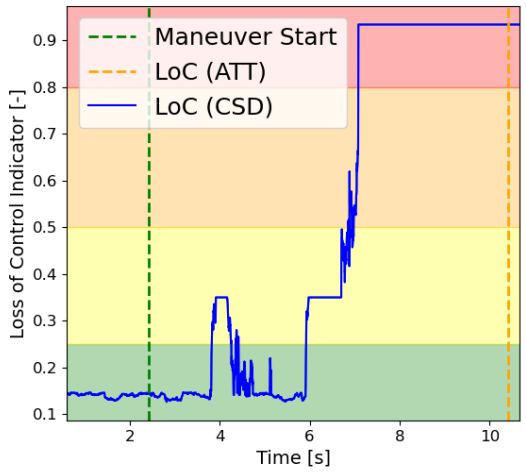

Fig. 25 Fuzzy LoC Indicator for Roll-LoC. The color-coded regions indicate stability and risk levels: green (stable), yellow (mid-level stability), orange (early warning signal), and red (LoC). The maneuver initiation is marked by the vertical green dashed line. The blue line depicts the Fuzzy LoC Indicator based on CSD rules (p, q, r, ω_i). The onset of ATT-based LoC detection is indicated by the vertical orange dashed line.

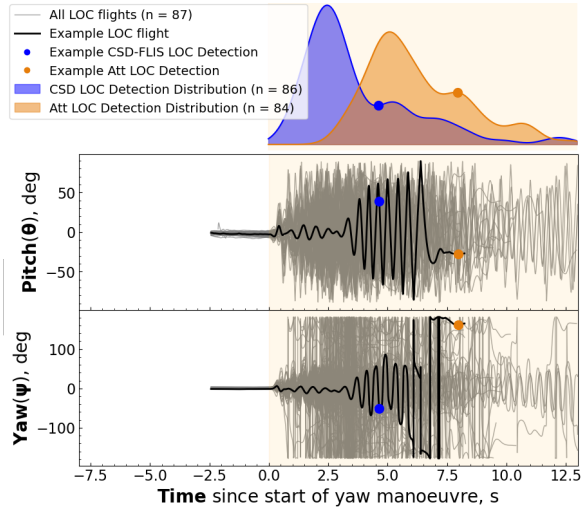

Fig. 26 LoC detection method comparison for Roll-LoC. Displayed are: CSD (blue), and attitude-based definition of [17]. The first row visualizes LoC detection timing relative to the start of the maneuver. The second and third row overlay pitch and yaw responses of 87 flights in gray, with a sample LoC flight in black. Colored dots indicate the moment LoC was detected by each method.

As highlighted in subsection V.B, the Roll-LoC maneuver exhibits no discernible lag between commanded and actual RPMs, leading to the application of only two LoC definitions for Roll-LoC: a CSD-based definition and an ATT-based definition from [17].

Figure 25 demonstrates the FLIS output for a Roll-LoC flight, where the quadrotor initially stays within the stable region. Upon the initiation of the roll maneuver, it temporarily transitions into the mid-region due to the initial coupling effect observed in subsection V.B, but soon returns to the stable green region. However, as the yaw rate begins to escalate, the FLIS output steadily increases, advancing into the EWS region. The increasing yaw rate amplifies the pitch oscillations until both approach instability, prompting the FLIS to signal a transition into the LoC region. The FLIS identifies a LoC event 4.65 seconds after the roll maneuver begins, whereas the ATT-based definition identifies it 8.98 seconds after the roll maneuver begins.

Figure 26 presents the detection performance of the two LoC definitions on Roll-LoC dataset. In the Yaw-LoC (CineGo) dataset, the application of CSD significantly improved detection rates relative to the ATT-based definition. The ATT-based definition for Roll-LoC already achieves a 97% detection rate. This difference in detection rates, for the same LoC definition, stems from the incorporation of the gimbal. Unlike in Yaw-LoC scenarios where drones often crash before reaching the 90-degree attitude criterion, the gimbal ensures the quadrotor meets the LoC criteria without crashing. Moreover, the distribution of detection times for Roll-LoC exhibit a broader range compared to Yaw-LoC, ranging from 2 to 8 seconds, which is due to the dependency of time to LoC with max yaw rate. This will be elaborated further in subsection VI.A

Table 4 LoC Detection Time Distribution Comparison between CSD versus ATT based LoC Definitions

CSD				ATT			
Mean	Var	TP	FP	Mean	Var	TP	FP
3.500 [s]	3.604 [s^2]	99%	9%	6.326 [s]	4.364 [s^2]	96%	0%

VI. Discussion

Combining the results outlined in the previous section, this section will discuss the fundamental differences in Roll vs Yaw induced LoC dataset, analyzing τ and CSD as an EWS indicator, and finally, Fuzzy Logic as an EWS inference system.

A. Roll vs Yaw induced LoC

The results from subsection V.A and subsection V.B highlight that both roll-induced and yaw-induced LoC share similar CSD trends, despite distinct differences in their LoC dynamics. The dynamics of these LoC events differ, particularly in how error accumulates. Recall that the essence of CSD lies in its ability to detect a system approaching a critical transition, reflected on its slowed recovery after a small perturbation.

In the case of yaw-induced LoC, perturbations are introduced in roll and pitch oscillations stemming from the "anti-gravity" feature of BetaFlight (the flight controller), which induces oscillations due to a misalignment between the center of gravity (c.g) and the Inertial Measurement Unit (IMU). The high yaw rates results in translational accelerometer measurements due to the c.g and IMU misalignment. Consequently, the flight controller misinterprets these readings as disturbances in roll and pitch angles, endeavoring to counteract them, which induces oscillations around the roll and pitch axes. At low rotational rates, or when the actuators are not saturated, these oscillations remain small and manageable, not leading to LoC. However, at high rotational rates or under actuator saturation, oscillations intensify and become unstable, thereby tying this mode of LoC to controller-induced LoC.

Conversely, roll-induced LoC introduces perturbations in yaw through the coupling of roll and yaw axis due to gyroscopic precession. Gyroscopic precession, observed in rotating bodies, occurs where an applied force causes the body to move in a direction perpendicular to the force's direction [50]. This effect results from the conservation of angular momentum and is commonly seen in gyroscopes and spinning objects like bicycle wheels, tops, and quadrotors rotors [51].

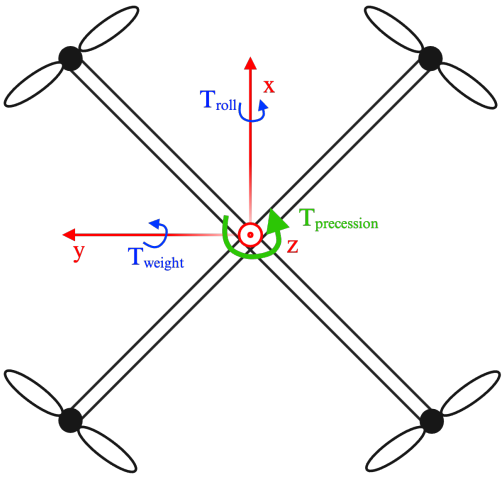

Fig. 27 Free Body Diagram of Quadrotor experiencing Gyroscopic Precession

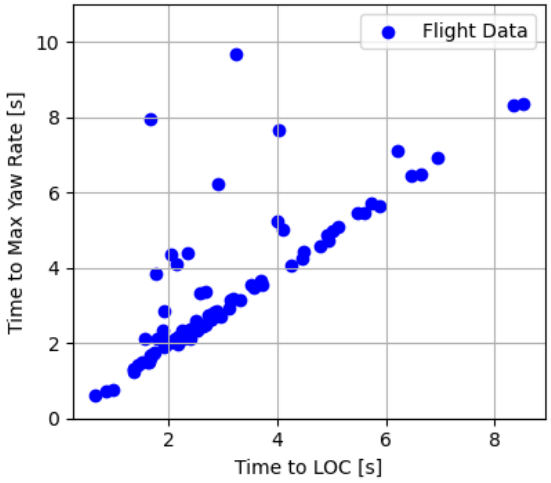

Fig. 28 Time to Max Yaw Rate vs Time to LoC

The visual representation of this effect is provided in Figure 27. In an ideal scenario, a quadrotor mounted on a gimbal and executing a pure roll maneuver would have the resultant torque vector T_{roll} placed in the x direction, and as roll rate increases, this would increase the angular momentum in the x axis direction, without causing any translational movement since the quadrotor is placed on a gimbal. However, in reality, c.g. misalignment occur, such as an improperly placed battery, which introduces a new torque vector T_{weight} in the y direction. This torque, when combined with the angular momentum from the rolling maneuver, redirects this angular momentum around the z axis, causing a yawing motion ($T_{precession}$). As yawing intensifies, it also triggers pitch oscillations, culminating in LoC when BetaFlight deactivates the I term to avoid integrator windup, halting the roll motion. Although roll-induced LoC also ultimately occurs due to the controller killing the I term, the reasons differ: for Yaw-LoC, the anti-gravity feature induces error

accumulation due to corrective measures, whereas for Roll-LoC, the error accumulates due to the coupling of roll and yaw.

Further analysis explored the relationship between time to LoC and the time taken for the quadrotor to achieve the maximum yaw rate, as depicted in Figure 28. A linear correlation is observed between the time to max yaw rate and time to LoC, solidifying our insights that gyroscopic precision-induced yawing is the primary driver of the roll-induced LoC dynamics. This can also be used to explain the wider spread in LoC detection times between roll-induced and yaw-induced LoC found in Figure 26. Variations in the c.g., due to battery placement after each flight run, affect the torque magnitude, contributing to the spread in LoC detection times shown.

Integrating these findings from the Roll-induced and Yaw-induced LoC maneuvers, a key takeaway emerges: despite the different dynamics behind each LoC type, CSD analysis consistently identifies trends that appear to apply across various LoC scenarios. This outcome highlights the effectiveness and broad applicability of CSD as a method for detecting potential LoC events, regardless of their specific causes or the maneuvers involved.

B. Time Delay Indicator

Saturation causes non-linear actuator responses, leading to disproportionate control inputs, phase lags, and system instability. When saturated, actuators may fail to adjust, causing oscillations or deviations from intended trajectories, resulting in erratic behavior. Given these effects, especially when dealing with LoC due to high rotational rates, indicators of actuator saturation or phase lag become effective for defining a LoC event. For the GEPRO (CineGo) quadrotor, CSD analysis from subsection V.A confirmed that the time-delay variable τ is an effective definition of LoC. Even with some false positives in SysID flights, the use of τ to obtain the delay duration variable t_τ, defining LoC as a phase lag persisting over 50 time-steps, has proven to improve its reliability as a LoC definition.

However, the TinyWhoop (DataCan) and Gimbal drone analyses from subsection V.A and subsection V.B indicated no (measureable) phase lag in smaller drones, pointing to a correlation between actuator size and τ detection. This seems plausible, since larger actuators, with their mass distributed away from the rotation axis, have a higher moment of inertia and thus exhibit slower response times. In contrast, smaller actuators exhibit lower inertia and quicker response times.

Considering this, increasing the sampling rate for smaller drones could offer a finer resolution for detecting shorter lags, given their rapid response capabilities. Ideally, CSD would serve as an EWS indicator and LoC detector, with t_τ acting as a redundancy layer to minimize false positives, and increase detection rates. Future research should delve deeper into the time delay variable, exploring how variations in sampling rate or different controller designs might influence lag detection efficacy. Additionally, testing Roll-LoC maneuvers on larger drones with gimbal setups could substantiate the assumption that τ detection correlates with actuator size and not LoC mode, ensuring that the results are applicable across various drone configurations.

C. Critical Slowing Down as an Early Warning Signal

CSD has emerged as an effective EWS for detecting quadrotor LoC. Its straightforward application, computational efficiency, and reliance solely on data from onboard IMUs, without the need for precise state estimation via external devices like OptiTrack, make it a practical choice. Table 8 highlights the comparative effectiveness of different LoC definitions: CSD-based, τ-based, and ATT-based. The insight from the table lead to several conclusions:

Table 5 LoC Detection Time Distribution Comparison between CSD-FLIS versus ATT based LoC Definitions

	Gyro-AR1				CSD-τ				ATT			
	Mean [s]	Var [s^2]	TP [%]	FP [%]	Mean [s]	Var [s^2]	TP [%]	FP [%]	Mean [s]	Var [s^2]	TP [%]	FP [%]
Yaw LoC	1.205	0.027	93	8	1.764	0.032	96	8	2.172	0.197	31	0
Roll LoC	3.500	3.604	99	9	-	-	-	-	6.326	4.364	96	0

1) **Detection Rates** - The Gyro-AR1 and CSD-τ based definitions demonstrate high detection rates, with Gyro-AR1 achieving detection rates of 93% and 99% for yaw and roll induced LoC scenarios, respectively. This marks a significant improvement over the ATT-based definition [17], which was foundational for exploring the use of

neural networks to predict time-to-LoC in quadrotors. Incorporating a consistent and non-binary LoC definition could potentially enhance the generalizability and performance of these neural networks.

2) **Prediction Time** - For yaw-induced LoC, the ATT-based definition detects LoC on average 2.172 seconds post-maneuver. The CSD-τ based approach improves this by 0.4 seconds, averaging detection at 1.764 seconds. The Gyro-AR1 methodology advances this even further, reducing the detection time to just 1.205 seconds, an entire second earlier than the ATT-based definition. This additional reaction time could mean the difference between a drone's recovery or crash.
3) **Generalizability** - CSD has shown to be broadly applicable, effectively serving as an EWS for LoC events across various quadrotor size/weight and LoC modes.
4) **Evaluation of CSD Indicators** - Gyro AR1 appears to provide the best EWS among the investigated indicators, offering the earliest detection of LoC, ease of use due to its natural bound between [-1,1], and consistency across different systems.

Nonetheless, a notable limitation exists: in scenarios where stick inputs occur across all axes simultaneously, the system's current approach falters, unable to interpret any of the CSD results. An ideal solution would use stick inputs to detrend gyro rates, ensuring that CSD analysis is focused on external system influences.

Theoretically, gyro error should only reflect oscillations due to external perturbations, as intentional maneuvers should not generate error. However, in practice, this is not the case. For instance, a step input will always be perceived as a large error due to the quadrotor's inability to meet sudden reference changes instantly. This discrepancy occurs because the sampling rate for gyro errors often exceeds the quadrotor's natural response rate to control inputs, resulting in transient differences between intended and actual gyro rates being recorded as errors. This complicates distinguishing between disturbances and control reference changes.

This complexity indicates the need for either improved detrending methods to distinguish between control reference changes and oscillations due to external disturbances, or to combine multiple metrics in order to describe the quadrotor's stability in various flight scenarios. Future research should aim to address these challenges, refining the accuracy and reliability of CSD as a predictive tool for LoC in quadrotors.

D. Fuzzy Logic as Early Warning Signal Inference System

Fuzzy logic has been employed in this research as a method to interpret CSD outputs. Its main advantages include simplicity in setup, the ability to aggregate multiple EWS indicators, and the conversion of complex AR1 and IQR values into more understandable linguistic variables. Furthermore, fuzzy logic's allowance for overlapping membership functions provides a margin of error when interpreting AR1/IQR values, offering flexibility near critical thresholds.

One of the biggest advantages of FLIS is the ability to implement domain expertise, particularly in two areas. First, insights from the CSD analysis underscore the necessity of the stick-neutrality condition when inferring the Gyro-AR1 results. Second, it's acknowledged that an AR1 value of 1 signals a tipping point, whereas a value of 0 indicates stability; however, when noise is considered, tipping is likely just below 1. This enables the precise placement of LoC membership functions slightly under an AR1 value of 1. Therefore, these qualitative domain knowledge can be incorporated into a quantitative metric.

The benefit of using FLIS to interpret multiple EWS can be showcased in Table 8. As can be seen, the detection times remain the same because the Gyro-AR1 rules usually detect LoC before the CSD-τ based rules; however, the main benefit of this is the generalizability to non-LoC maneuvers as well. The FLIS allows the analysis of additional flight data containing nominal SysID maneuvers that are coupled in all three axis. By combining both sets of data, the FLIS has a 99% detection rate and an 8% false positive rate from 96 LoC flights, and 67 SysID flights.

Since the FLIS was developed using SciKit's Fuzzy logic toolbox, the main tasks involved defining the Database and Rule Base. These define the membership functions, their thresholds, and the fuzzy rules. The database translates an AR1 value of 0.2, into a 'stable' linguistic value, aiding in verifying the accuracy of set thresholds through the FLIS outputs. Correctly set thresholds should reflect the quadrotor's actual behavior, indicating stability prior to LoC and transitioning from stable to LoC post-maneuver.

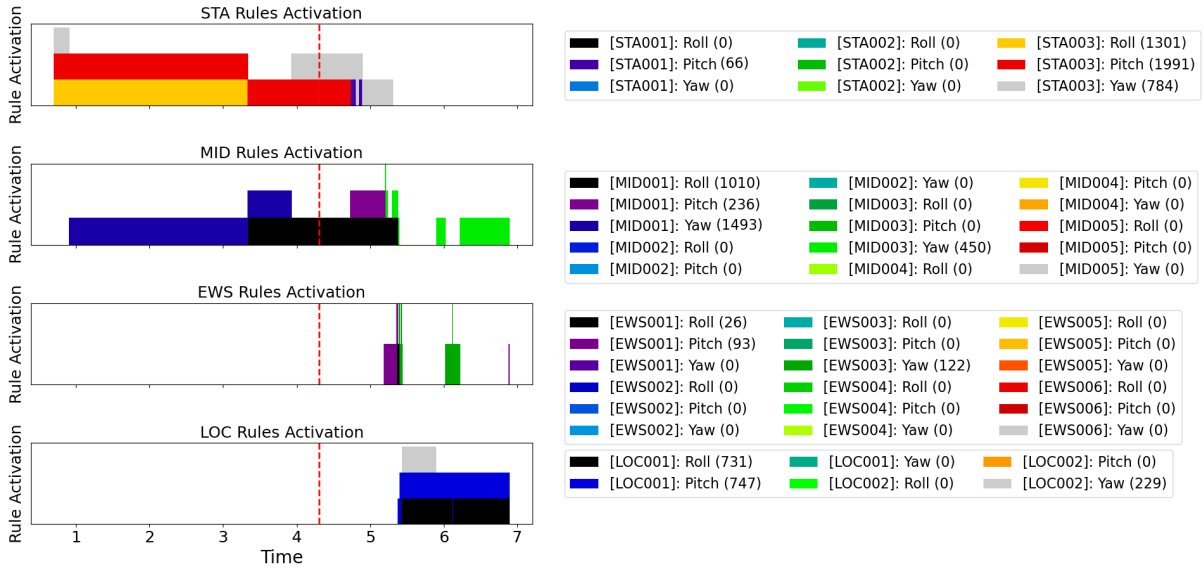

Fig. 29 Output of the Rule Activation Detection Algorithm output for a Yaw-LoC Flight (CineGo). The four subplots display the temporal distribution of rule activation within the time-series, with a red vertical line marking the start of the yaw maneuver. The legend on the right depicts each rule with its color coding and the total number of activation through the dataset.

For the fuzzy rules, two aspects need verification : the timing of rule activation in flight data, and the interrelationship of rules to ensure no conflicts. However, the pre-made toolbox treats FLIS somewhat as a black-box, not providing any insights into rule activation. To address this, two key measures were implemented: a rule activation detection algorithm to track the activation of rules, and a cross-correlation analysis to investigate the relationships between all the rules.

The rule activation detection algorithm is designed to verify that the fuzzy rules trigger at appropriate moments in the flight data timeline. Specifically, Stable (STA) rules should activate before the maneuver begins, Mid rules before and slightly after the start of the maneuver, and Early Warning Signal (EWS)/Loss of Control (LoC) rules activate only after the maneuver has started. By creating a dictionary containing all rules, their antecedents, and corresponding membership values, each FLIS iteration's membership states are calculated to determine the active rules. An example of the outcomes for an arbitrary LoC flight is depicted in Figure 29. Indeed, it can be seen that the rules are activating at the correct phases, aligning with the intended criteria. A comprehensive distribution analysis of rule activation for all the flight data, detailed in Appendix A.5, further confirms that each rule activates at the appropriate stage relative to the maneuver's initiation.

Subsequently, the relationship between each rule was examined through cross-correlation analysis. This analysis aims to understand how rules interact with each other, where a high correlation between two rules suggests simultaneous activation. Ideally, rules within the same group should exhibit high correlation, while rules from different groups should show low or negative correlation. This pattern is visible in the Figure 30, especially near the diagonal where highly correlated rules signify intra-group consistency, while lower off-diagonal correlations indicate proper segregation between different rule groups. It's noteworthy that the bottom left corner of the figure shows a relatively high correlation between EWS004(Yaw) and MID002(Yaw), despite them belonging to different rule groups. This anomaly can be attributed to certain Mid rules being designed to trigger immediately post-maneuver, coupled with the inherent overlap in fuzzy membership functions allowing antecedents to fall within two categories. This overlap explains why some MID and EWS rules exhibit unexpectedly high correlations. Similar correlation results can be seen for the Yaw-LoC, which can be found in Appendix A.5.

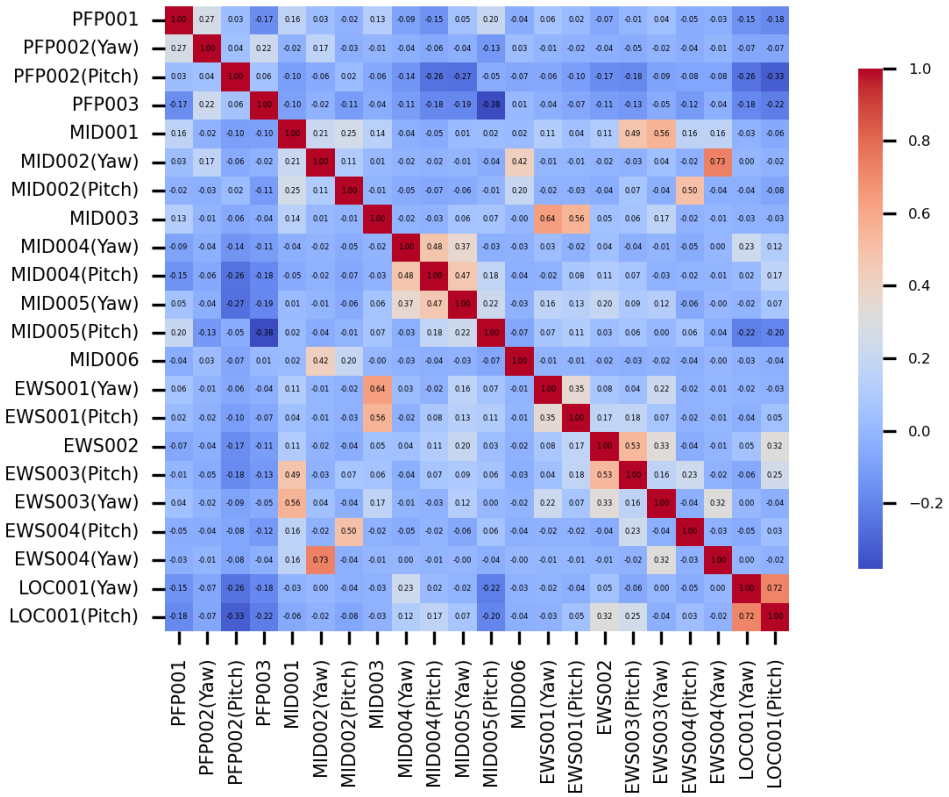

Fig. 30 Correlation Matrix of Fuzzy Rules for Roll-LoC

These methodologies for analyzing rule activation and inter-rule correlations offer deep insights into the fuzzy rules' operational effectiveness and the absence conflicts between rules. Overall, the FLIS demonstrates a robust capacity for interpreting CSD outputs, evidenced by the performance metrics in Table 8. This approach not only complements but also extends van der Pluijm's [11] research on using Kendall Tau for AR1 value interpretation, showcasing the FLIS's ability to integrate multiple variables, set nuanced boundaries for different AR1 levels, and effectively prevent false positives.

VII. Conclusion

This research embarked on addressing the critical challenge of Loss-of-Control (LoC) in quadrotors, a prevalent cause of drone failures, through the use of Critical Slowing Down (CSD) theory and a Fuzzy Logic Inference System (FLIS). The study focused on two different LoC scenarios induced by demanding high rotational rates: roll induced LoC (Roll-LoC) and yaw induced LoC (Yaw-LoC). The Yaw-LoC data were collected from a real flight of a GEPRO quadrotor, and the Roll-LoC data generated through experiments on a custom-built drone mounted on a 3-axis quadrotor gimbal testing rig. Initial findings identified gyro rates and rotor rpm values as highly indicative CSD, thus chosen for the subsequent analysis, while kernel regression and high pass filters effectively eliminated long-term trends, yielding accurate residual data for CSD analysis.

Comparison of CSD outcomes for two different drones undergoing distinct LoC maneuvers revealed similar CSD patterns, despite differences in LoC characteristics, as well as drone weight and size. Besides CSD, time delay, an indicator reflecting the actuator phase lag, served as an additional LoC indicator. However, the detection of time delays was only observed in larger quadrotors, suggesting a correlation between phase lag detection and actuator size. Future work should seek to investigate the relationship between actuator size and the detectability of phase lag, if any.

CSD findings were then inferred using FLIS, translating variables such as AR1 and IQR from gyro rates and measured rotor speeds, along with the time delay variable, into a single fuzzy value [0,1] indicative of the quadrotor's proximity to a LoC event. The CSD-FLIS framework proved effective as a general LoC detection method, with high

detection rates (True Positive: 96% for Yaw-LoC, 98% for Roll-LoC) and low missed detection rates (False Positive: 8% for Yaw-LoC, 9% for Roll-LoC), applicable across different quadrotors and LoC modes. Moreover, the CSD-FLIS framework was able to improve the detection rate compared to the previous ATT-based definition, which had a detection rate of 31%.

Beyond detection rates, the CSD-FLIS framework also excelled in prediction time compared to the ATT-based definition. On average, for Yaw-LoC, the CSD-FLIS detected a LoC event 1.2 seconds after the maneuver, whereas the ATT-based definition detected it 2.2 seconds later. For Roll-LoC, the framework detected a LoC event 3.5 seconds after the maneuver, compared to 6.3 seconds with the ATT-based definition. These earlier detection times enable more LoC experiments to be conducted without risking quadrotor crashes, facilitating a more *sustainable* approach to generating LoC data.

The main limitation of this method is its sensitivity to the chosen detrending method, impacting its vulnerability to false positives. Two potential solutions were proposed: firstly, use of stick inputs to define the overall trend for signal detrending, to only capture the residual oscillations attributed to movements in axes other than the one being directly manipulated. Secondly, further exploration into time-delay variable is recommended, to enhance phase lag detection in smaller quadrotors, possibly through increased sampling rates.

Appendix

A.1 - Detrending Analysis

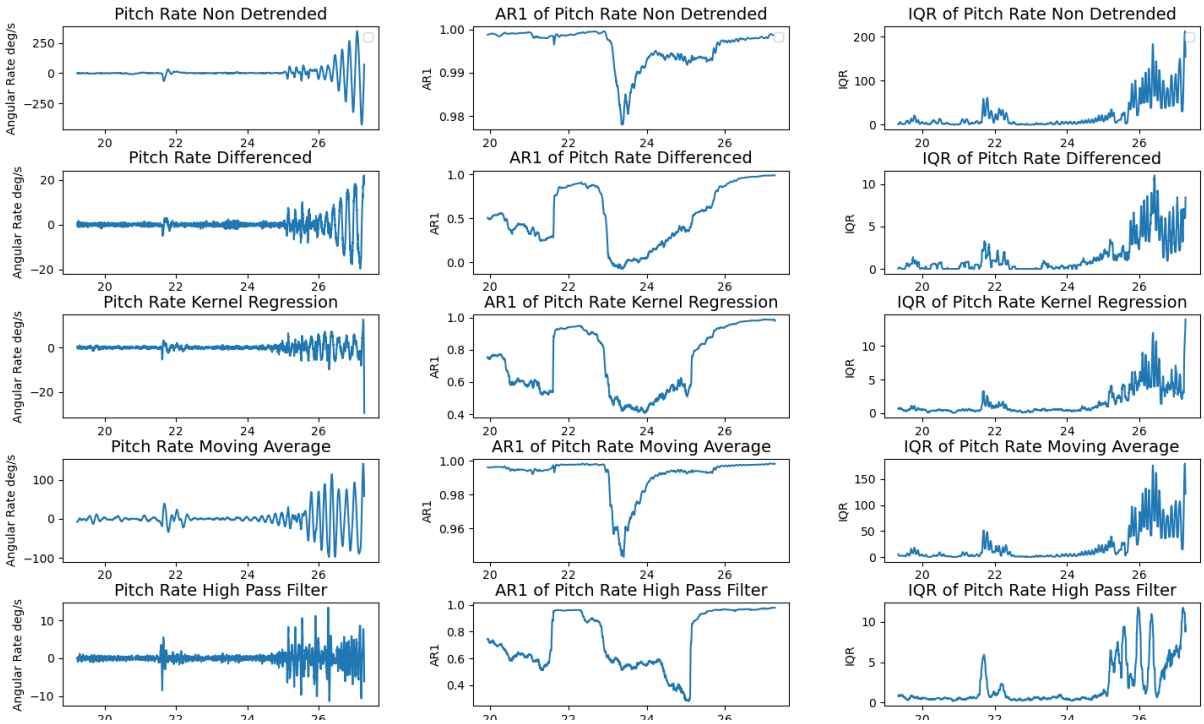

Fig. 31 Comparison of various detrending methods applied to gyro rates and their corresponding AR1 and IQR time-series

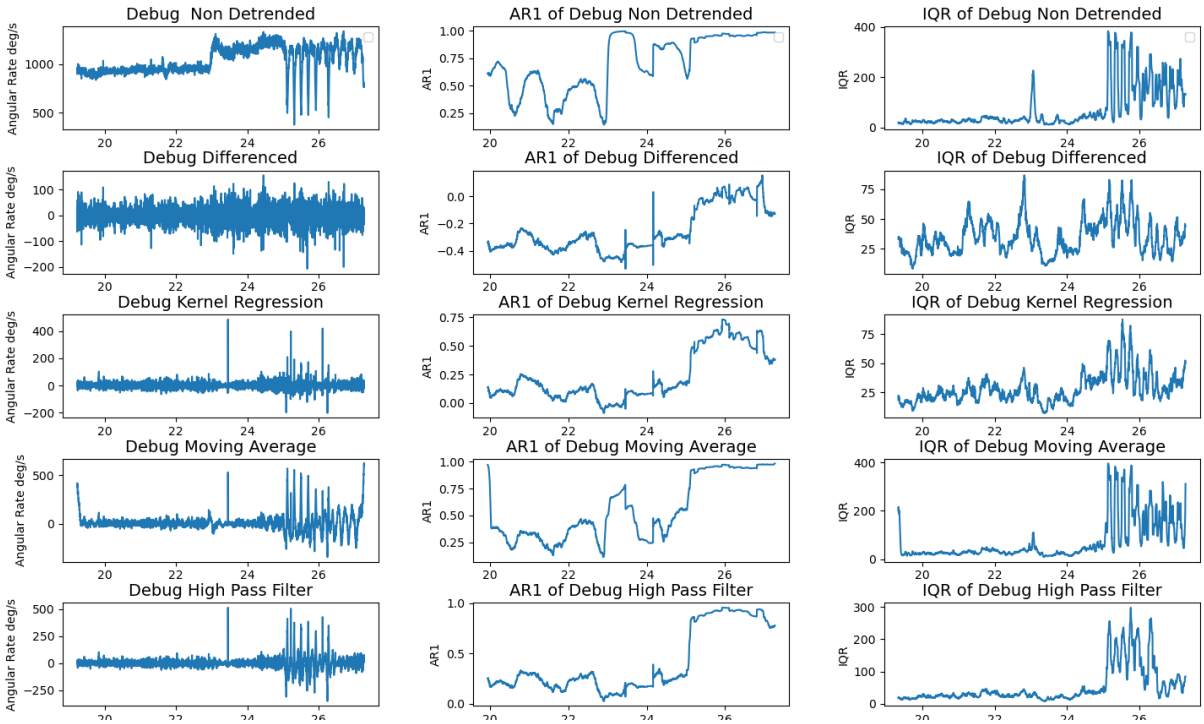

Fig. 32 Comparison of various detrending methods applied to measured RPM (debug) and their corresponding AR1 and IQR time-series

A.2 - Algorithms

Algorithm 1 Update Delay Duration

```
function UpdateDelayDuration(tau, delay_duration, zero_delay_count)
    if time_delay = 0 then
        zero_delay_count ← zero_delay_count + 1      ▷ Check if tau is 0 and update zero_delay_count or reset it
    else
        zero_delay_count ← 0                         ▷ Reset counter if time_delay is not 0
    end if
    if zero_delay_count > 50 then
        delay_duration ← 0
        zero_delay_count ← 0                         ▷ Reset the zero delay count for next time
    else if time_delay > 4 then                      ▷ Update delay_duration based on tau value
        if time_delay < 10 then
            delay_duration ← delay_duration + 1
        else if 10 ≤ time_delay < 20 then
            delay_duration ← delay_duration + 2
        else if time_delay ≥ 20 then
            delay_duration ← delay_duration + 3
        end if
    end if
    return delay_duration, zero_delay_count
end function
```

Algorithm 2 Kernel Regression and AR(1) Calculation

function KR_AND_AR1(y, $time$, bw, $windowSize$, $step$, var_name, $logger$, $bw\_inc = 0.001$, $max\_iter = 500$)
  $y\_detrend \leftarrow$ empty_like(y)
  $y\_trend \leftarrow$ empty_like(y)
  $ar1\_values \leftarrow$ empty(($y.shape[0]$, max(0, (y.shape[1] $-$ $windowSize$)//$step$)))
  for $i = 0$ **to** $y.shape[0] - 1$ **do**
    $bw\_adjusted \leftarrow bw[i]$
    $detrended\_series \leftarrow$ array([])
    $trend\_series \leftarrow$ array([])
    $ar1\_series \leftarrow$ array([])
    for $current\_position = 0$ **to** y.shape[1] $-$ $windowSize$ **step** $step$ **do**
      if $current\_position + windowSize >$ y.shape[1] **then**
        continue ▷ Ensure we have a full window
      end if
      $windowed\_data \leftarrow y[i, current\_position : current\_position + windowSize]$
      $windowed\_time \leftarrow time[current\_position : current\_position + windowSize]$
      $kReg \leftarrow$ KernelReg($endog = windowed\_data, exog = windowed\_time, var\_type = c, bw = [bw\_adjusted], reg\_type = lc$)
      $fitted, \_ \leftarrow kReg.fit(windowed\_time)$
      $window\_trend \leftarrow fitted$ ▷ Windowed Trend Line
      $window\_detrend \leftarrow windowed\_data - window\_trend$ ▷ Windowed Detrend Line
      if $current\_position = 0$ **then** ▷ Keep everything for first window
        $detrended\_series \leftarrow$ concatenate($detrended\_series, window\_detrend$)
        $trend\_series \leftarrow$ concatenate($trend\_series, window\_trend$)
      else ▷ Only append last term for proceeding windows
        $detrended\_series \leftarrow$ append($detrended\_series, window\_detrend[-1]$)
        $trend\_series \leftarrow$ append($trend\_series, window\_trend[-1]$)
      end if
      if $current\_position + windowSize + step \leq$ y.shape[1] **then** ▷ Last window
        $next\_window \leftarrow y[i, current\_position + step : current\_position + windowSize + step] - window\_trend$
        $ARCoeff \leftarrow$ corr($window\_detrend, next\_window$)
        $ar1\_series \leftarrow$ append($ar1\_series, ARCoeff$)
      end if
    end for
    $ar1\_values[i, : len(ar1\_series)] \leftarrow ar1\_series[:$ len($ar1\_values[i]$)$]$
    $y\_detrend[i, : len(detrended\_series)] \leftarrow detrended\_series$
    $y\_trend[i, : len(trend\_series)] \leftarrow trend\_series$
  end for
  return $y\_detrend$, y_trend, $ar1\_values$
end function

Algorithm 3 HPF and AR(1) Calculation

function HPF_AND_AR1(y, $time$, $cutoff$, $windowSize$, $step$, fs, var_name, $logger$, $order = 5$, $bw\_inc = 0.001$, $max\_iter = 10$)
  $y\_filtered \leftarrow$ empty_like(y)
  $y\_trend \leftarrow$ empty_like(y)
  $ar1\_values \leftarrow$ empty($y.shape[0]$)
  for $i = 0$ **to** $y.shape[0] - 1$ **do**
    $iteration \leftarrow 0$
    $cutoff\_adjusted \leftarrow cutoff[i]$
    $nan \leftarrow$ True
    while $iteration < max\_iter$ **and** nan **do**
      $filtered\_data, trend\_data \leftarrow$ highpass_filter($y[i,:], cutoff\_adjusted, fs, order$)
      $y\_filtered[i,:] \leftarrow filtered\_data$
      $y\_trend[i,:] \leftarrow trend\_data$
      $ar1 \leftarrow$ indicatorsCSD($y\_filtered[i,:], windowSize, step$)
      if not $np.isnan(ar1).any()$ **then**
        $ar1\_values[i] \leftarrow ar1$
        $nan \leftarrow$ False
      else
        $cutoff\_adjusted \leftarrow cutoff\_adjusted + bw\_inc$
        $iteration \leftarrow iteration + 1$
      end if
    end while
    if $iteration == max\_iter$ **then**
      print("Warning: Max iterations reached for series ", i, ". AR1 autocorrelation may be NaN.")
    else
      print("Cutoff frequency adjusted to ", $cutoff\_adjusted$, " for series ", i, " with AR1 autocorrelation ", $ar1$, ".")
      logger.info("Chosen Cutoff Freq for ", var_name, " KR: ", $cutoff\_adjusted$)
    end if
  end for
  if $y\_filtered.shape[0] == 1$ **then**
    $y\_filtered \leftarrow y\_filtered.reshape(-1)$
    $y\_trend \leftarrow y\_trend.reshape(-1)$
  end if
  return $y\_filtered, y\_trend, ar1\_values$
end function

A.3 - Fuzzy Logic Inference System Architecture

Table 6: FLIS-Yaw Variable Description: membership functions, function shape, overlap intervals

Variable	Range	Step	Membership	Function Shape	Interval
(p, q, r) AR1	[-0.2, 1.0]	0.01	STABLE	Trapezoidal	[-0.4, -0.2, 0, 0.25]
			MID	Triangular	[0.05, 0.4, 0.6]
			EWS	Trapezoidal	[0.55, 0.6, 0.79, 0.79]
			LoC	Trapezoidal	[0.80, 0.80, 1, 1]
$(\omega_0, \omega_1, \omega_2, \omega_3)$ AR1	[-0.6, 1.0]	0.01	STABLE	Trapezoidal	[-0.6, -0.6, 0.4, 0.5]
			EWS	Trapezoidal	[0.4, 0.5, 1, 1]
(p, q, r) IQR	[0, 500]	1	STABLE	Trapezoidal	[0, 0, 0, 0]
			EWS	Trapezoidal	[1, 1, 500, 500]
(Φ, Θ, Ψ) Inputs	[0, 1]	0.1	STABLE	Trapezoidal	[0, 0, 0.3, 0.3]
			LoC	Trapezoidal	[0.7, 0.7, 1, 1]
t_τ	[0, 1000]	0.1	STABLE	Trapezoidal	[0, 0, 0, 0]
			EWS	Triangular	[0.8, 40, 40]
			LoC	Trapezoidal	[41, 50, 1000, 1000]
Consequent	[0, 1]	0.01	STABLE	Trapezoidal	[0, 0, 0.2, 0.3]
			MID	Trapezoidal	[0.2, 0.3, 0.4, 0.5]
			EWS	Trapezoidal	[0.45, 0.5, 0.8, 0.8]
			LoC	Triangular	[0.8, 1, 1]

Table 7: FLIS-Roll Variable Description: membership functions, function shape, overlap intervals

Variable	Range	Step	Membership	Function Shape	Interval
(p, q, r) AR1	[-0.2, 1.0]	0.01	STABLE	Trapezoidal	[-0.4, -0.4, 0, 0.25]
			MID	Triangular	[0.2, 0.4, 0.6]
			EWS	Trapezoidal	[0.5, 0.6, 0.7, 0.8]
			LoC	Trapezoidal	[0.8, 0.8, 1, 1]
$(\omega_0, \omega_1, \omega_2, \omega_3)$ AR1	[-0.6, 1.0]	0.01	STABLE	Trapezoidal	[-0.6, -0.6, 0.3, 0.32]
			EWS	Trapezoidal	[0.3, 0.4, 1, 1]
(p, q, r) IQR	[0, 500]	1	STABLE	Trapezoidal	[0, 0, 4, 5]
			EWS	Trapezoidal	[4, 5, 10, 15]
			LoC	Trapezoidal	[10, 15, 500, 500]
(Φ, Θ, Ψ) Inputs	[0, 1]	0.1	STABLE	Trapezoidal	[0, 0, 0.3, 0.3]
			LoC	Trapezoidal	[0.7, 0.7, 1, 1]
Consequent	[0, 1]	0.01	STABLE	Trapezoidal	[0, 0, 0.2, 0.3]
			MID	Trapezoidal	[0.2, 0.3, 0.4, 0.5]
			EWS	Trapezoidal	[0.45, 0.5, 0.8, 0.85]
			LoC	Triangular	[0.8, 1, 1]

Table 8: Fuzzy Rules for Yaw-LoC (Gyro-AR1 & CSD-τ). Columns: (1) Rule ID – identifies each rule, (2) Logic Operator – the logical connector used within the rule ('&' for 'AND', '∼' for 'NOT', '|' for 'OR'), (3 & 4) Antecedents & Membership Values – the variables and their respective membership value to activate the rule, (5) Consequent Membership Value – the outcome's degree of truth, (6) Rule Description – an elaboration of the rule's application.

Rule ID	Condition	Antecedents	Membership	Consequent	Description
STA001	AND	Φ Input	stable	STABLE	If stick input is neutral, AND gyro AR1 is stable, THEN output is stable.
	AND	p AR1	stable		
STA001	AND	Θ Input	stable	STABLE	If stick input is neutral, AND gyro AR1 is stable, THEN output is stable.
	AND	q AR1	stable		
STA001	AND	Ψ Input	stable	STABLE	If stick input is neutral, AND gyro AR1 is stable, THEN output is stable.
	AND	r AR1	stable		
STA002	AND	Φ Input	LoC	STABLE	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND ALL measured rotor speed's AR1 is stable, and no lag, THEN output is stable.
	AND	p IQR	stable		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	stable		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
STA002	AND	Θ Input	LoC	STABLE	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND ALL measured rotor speed's AR1 is stable, and no lag, THEN output is stable.
	AND	q IQR	stable		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	stable		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input \ r AR1	$\sim$ (stable&LoC)		
STA002	AND	Ψ Input	LoC	STABLE	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND ALL measured rotor speed's AR1 is stable, and no lag, THEN output is stable.
	AND	r IQR	stable		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	stable		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Θ Input & q AR1	$\sim$ (stable&LoC)		
STA003	AND	Φ Input	LoC	STABLE	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND ALL measured rotor speed's AR1 is stable, and no lag,THEN output is stable.
	AND	p IQR	EWS		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	stable		
	OR	Θ Input q AR1	$\sim$ (stable&LoC)		
		Ψ Input r AR1	$\sim$ (stable&LoC)		

Continued on next page

Table 8: Fuzzy Rules for Yaw-LoC (Gyro-AR1 & CSD-τ). Columns: (1) Rule ID – identifies each rule, (2) Logic Operator – the logical connector used within the rule ('&' for 'AND', '∼' for 'NOT', '|' for 'OR'), (3 & 4) Antecedents & Membership Values – the variables and their respective membership value to activate the rule, (5) Consequent Membership Value – the outcome's degree of truth, (6) Rule Description – an elaboration of the rule's application. (Continued)

STA003	AND	Θ Input	LoC	STABLE	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND ALL measured rotor speed's AR1 is stable, and no lag,THEN output is stable.
	AND	q IQR	EWS		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	stable		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input r AR1	$\sim$ (stable&LoC)		
MID001	AND	Φ Input	stable	MID	If stick input is neutral, AND gyro AR1 is MID, THEN output is mid.
	AND	p AR1	mid		
MID001	AND	Θ Input	stable	MID	If stick input is neutral, AND gyro AR1 is MID, THEN output is mid.
	AND	q AR1	mid		
MID001	AND	Ψ Input	stable	MID	If stick input is neutral, AND gyro AR1 is MID, THEN output is mid.
	AND	r AR1	mid		
MID002	AND	Φ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND one of the measured rotor speed's AR1 is EWS, and no lag,THEN output is mid.
	AND	p IQR	stable		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	stable		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
MID002	AND	Θ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND one of the measured rotor speed's AR1 is EWS, and no lag,THEN output is mid.
	AND	q IQR	stable		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	stable		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
MID002	AND	Ψ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND one of the measured rotor speed's AR1 is EWS, and no lag,THEN output is mid.
	AND	r IQR	stable		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	Stable		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Θ Input & q AR1	$\sim$ (stable&LoC)		

Continued on next page

Table 8: Fuzzy Rules for Yaw-LoC (Gyro-AR1 & CSD-τ). Columns: (1) Rule ID – identifies each rule, (2) Logic Operator – the logical connector used within the rule ('&' for 'AND', '∼' for 'NOT', '|' for 'OR'), (3 & 4) Antecedents & Membership Values – the variables and their respective membership value to activate the rule, (5) Consequent Membership Value – the outcome's degree of truth, (6) Rule Description – an elaboration of the rule's application. (Continued)

MID003	AND	Φ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND one of the measured rotor speed's AR1 is EWS, and no lag,THEN output is mid.
	AND	p IQR	EWS		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	stable		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
MID003	AND	Θ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND one of the measured rotor speed's AR1 is EWS, and no lag,THEN output is mid.
	AND	q IQR	EWS		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	stable		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
MID003	AND	Ψ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND one of the measured rotor speed's AR1 is EWS, and no lag,THEN output is mid.
	AND	r IQR	EWS		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	stable		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
MID004	AND	Φ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is EWS,THEN output is mid.
	AND	p IQR	stable		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	EWS		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
MID004	AND	Θ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is EWS,THEN output is mid.
	AND	q IQR	stable		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	EWS		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		

Continued on next page

Table 8: Fuzzy Rules for Yaw-LoC (Gyro-AR1 & CSD-τ). Columns: (1) Rule ID – identifies each rule, (2) Logic Operator – the logical connector used within the rule ('&' for 'AND', '∼' for 'NOT', '|' for 'OR'), (3 & 4) Antecedents & Membership Values – the variables and their respective membership value to activate the rule, (5) Consequent Membership Value – the outcome's degree of truth, (6) Rule Description – an elaboration of the rule's application. (Continued)

MID004	AND	Ψ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is EWS,THEN output is mid.
	AND	r IQR	stable		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	EWS		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Θ Input & q AR1	$\sim$ (stable&LoC)		
MID005	AND	Φ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is EWS,THEN output is mid.
	AND	p IQR	EWS		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	EWS		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
MID005	AND	Θ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is EWS,THEN output is mid.
	AND	q IQR	EWS		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	EWS		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
MID005	AND	Ψ Input	LoC	MID	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is EWS,THEN output is mid.
	AND	r IQR	EWS		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	EWS		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Θ Input & q AR1	$\sim$ (stable&LoC)		
EWS001	AND	Φ Input	stable	EWS	While other axis is not LoC, If stick input is neutral, AND gyroAR1 is MID, THEN output is EWS.
	AND	p AR1	EWS		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
EWS001	AND	Θ Input	stable	EWS	While other axis is not LoC, If stick input is neutral, AND gyroAR1 is MID,THEN output is EWS.
	AND	q AR1	EWS		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		

Continued on next page

Table 8: Fuzzy Rules for Yaw-LoC (Gyro-AR1 & CSD-τ). Columns: (1) Rule ID – identifies each rule, (2) Logic Operator – the logical connector used within the rule ('&' for 'AND', '∼' for 'NOT', '|' for 'OR'), (3 & 4) Antecedents & Membership Values – the variables and their respective membership value to activate the rule, (5) Consequent Membership Value – the outcome's degree of truth, (6) Rule Description – an elaboration of the rule's application. (Continued)

EWS001	AND	Ψ Input	stable	EWS	While other axis is not LoC, If stick input is neutral, AND gyroAR1 is MID,THEN output is EWS.
	AND	r AR1	EWS		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Θ Input & q AR1	$\sim$ (stable&LoC)		
EWS002	AND	Φ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND one of the measured rotor speed's AR1 is EWS, AND Delay Duration is EWS,THEN output is EWS.
	AND	p IQR	stable		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	EWS		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
EWS002	AND	Θ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND one of the measured rotor speed's AR1 is EWS, AND Delay Duration is EWS,THEN output is EWS.
	AND	q IQR	stable		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	EWS		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
EWS002	AND	Ψ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND one of the measured rotor speed's AR1 is EWS, AND Delay Duration is EWS,THEN output is EWS.
	AND	r IQR	stable		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	EWS		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Θ Input & q AR1	$\sim$ (stable&LoC)		
EWS003	AND	Φ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND one of the measured rotor speed's AR1 is EWS, AND Delay Duration is EWS,THEN output is EWS.
	AND	p IQR	EWS		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	EWS		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
EWS003	AND	Θ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND one of the measured rotor speed's AR1 is EWS, AND Delay Duration is EWS,THEN output is EWS.
	AND	q IQR	EWS		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	EWS		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		

Continued on next page

Table 8: Fuzzy Rules for Yaw-LoC (Gyro-AR1 & CSD-τ). Columns: (1) Rule ID – identifies each rule, (2) Logic Operator – the logical connector used within the rule ('&' for 'AND', '∼' for 'NOT', '|' for 'OR'), (3 & 4) Antecedents & Membership Values – the variables and their respective membership value to activate the rule, (5) Consequent Membership Value – the outcome's degree of truth, (6) Rule Description – an elaboration of the rule's application. (Continued)

EWS003	AND	Ψ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND one of the measured rotor speed's AR1 is EWS, AND Delay Duration is EWS,THEN output is EWS.
	AND	r IQR	EWS		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	EWS		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Θ Input & q AR1	$\sim$ (stable&LoC)		
EWS004	AND	Φ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is LoC,THEN output is EWS.
	AND	p IQR	stable		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	LoC		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
EWS004	AND	Θ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is LoC,THEN output is EWS.
	AND	q IQR	stable		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	LoC		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
EWS004	AND	Ψ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is LoC,THEN output is EWS.
	AND	r IQR	stable		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	LoC		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Θ Input & q AR1	$\sim$ (stable&LoC)		
EWS005	AND	Φ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is LoC,THEN output is EWS.
	AND	p IQR	EWS		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	LoC		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		

Continued on next page

Table 8: Fuzzy Rules for Yaw-LoC (Gyro-AR1 & CSD-τ). Columns: (1) Rule ID – identifies each rule, (2) Logic Operator – the logical connector used within the rule ('&' for 'AND', '∼' for 'NOT', '|' for 'OR'), (3 & 4) Antecedents & Membership Values – the variables and their respective membership value to activate the rule, (5) Consequent Membership Value – the outcome's degree of truth, (6) Rule Description – an elaboration of the rule's application. (Continued)

EWS005	AND	Θ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is LoC,THEN output is EWS.
	AND	q IQR	EWS		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	LoC		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
EWS005	AND	Ψ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND ALL measured rotor speed's AR1 is stable, AND Delay Duration is LoC,THEN output is EWS.
	AND	r IQR	EWS		
	AND	$(\omega_0\&\omega_1\&\omega_2\&\omega_3)$ AR1	stable		
	AND	t_τ	LoC		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Θ Input & q AR1	$\sim$ (stable&LoC)		
EWS006	AND	Φ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND one of the measured rotor speed's AR1 is EWS, AND Delay Duration is LoC,THEN output is EWS.
	AND	p IQR	stable		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	LoC		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
EWS006	AND	Θ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND one of the measured rotor speed's AR1 is EWS, AND Delay Duration is LoC,THEN output is EWS.
	AND	q IQR	stable		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	LoC		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
EWS006	AND	Ψ Input	LoC	EWS	While Gyro-AR1 rules are not applicable, AND Gyro IQR is stable, AND one of the measured rotor speed's AR1 is EWS, AND Delay Duration is LoC,THEN output is EWS.
	AND	r IQR	stable		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	LoC		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Θ Input & q AR1	$\sim$ (stable&LoC)		

Continued on next page

Table 8: Fuzzy Rules for Yaw-LoC (Gyro-AR1 & CSD-τ). Columns: (1) Rule ID – identifies each rule, (2) Logic Operator – the logical connector used within the rule ('&' for 'AND', '∼' for 'NOT', '|' for 'OR'), (3 & 4) Antecedents & Membership Values – the variables and their respective membership value to activate the rule, (5) Consequent Membership Value – the outcome's degree of truth, (6) Rule Description – an elaboration of the rule's application. (Continued)

LOC001	AND	Φ Input	stable	LOC	If stick input is neutral, AND gyro AR1 is LoC,THEN output is LoC.
	AND	p AR1	LoC		
LOC001	AND	Θ Input	stable	LOC	If stick input is neutral, AND gyro AR1 is LoC,THEN output is LoC.
	AND	q AR1	LoC		
LOC001	AND	Ψ Input	stable	LOC	If stick input is neutral, AND gyro AR1 is LoC,THEN output is LoC.
	AND	r AR1	LoC		
LOC002	AND	Φ Input	LoC	LOC	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND one of the measured rotor speed's AR1 is EWS, AND Delay Duration is LoC,THEN output is LoC.
	AND	p IQR	EWS		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	LoC		
	OR	Θ Input & q AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
LOC002	AND	Θ Input	LoC	LOC	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND one of the measured rotor speed's AR1 is EWS, AND Delay Duration is LoC,THEN output is LoC.
	AND	q IQR	EWS		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	LoC		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Ψ Input & r AR1	$\sim$ (stable&LoC)		
LOC002	AND	Ψ Input	LoC	LOC	While Gyro-AR1 rules are not applicable, AND Gyro IQR is EWS, AND one of the measured rotor speed's AR1 is EWS,AND Delay Duration is LoC,THEN output is LoC.
	AND	r IQR	EWS		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
	AND	t_τ	LoC		
	OR	Φ Input & p AR1	$\sim$ (stable&LoC)		
		Θ Input & q AR1	$\sim$ (stable&LoC)		

Table 9: Fuzzy Rules for Roll-LoC. Columns: (1) Rule ID – identifies each rule, (2) Logic Operator – the logical connector used within the rule ('&' for 'AND', '∼' for 'NOT', '|' for 'OR'), (3 & 4) Antecedents & Membership Values – the variables and their respective membership value to activate the rule, (5) Consequent Membership Value – the outcome's degree of truth, (6) Rule Description – an elaboration of the rule's application.

Rule ID	**Condition**	**Antecedents**	**Membership**	**Consequent**	**Description**
STA001	AND	Φ Input	LoC	stable	If both gyro AR1 is mid, AND ALL measured rotor speed's AR1 is stable, THEN output is stable.
	AND	q AR1	mid		
	AND	r AR1	mid		
	AND	$(\omega_0 \& \omega_1 \& \omega_2 \& \omega_3)$ AR1	stable		
STA002	AND	Φ Input	LoC	stable	If both gyro AR1 is mid/stable, AND ALL measured rotor speed's AR1 is stable, THEN output is stable.
	AND	q AR1	mid		
	AND	r AR1	stable		
	AND	$(\omega_0 \& \omega_1 \& \omega_2 \& \omega_3)$ AR1	stable		
STA002	AND	Φ Input	LoC	stable	If both gyro AR1 is mid/stable, AND ALL measured rotor speed's AR1 is stable, THEN output is stable.
	AND	q AR1	stable		
	AND	r AR1	mid		
	AND	$(\omega_0 \& \omega_1 \& \omega_2 \& \omega_3)$ AR1	stable		
STA003	AND	Φ Input	LoC	stable	If both gyro AR1 is stable, AND ALL measured rotor speed's AR1 is stable, THEN output is stable.
	AND	q AR1	stable		
	AND	r AR1	stable		
	AND	$(\omega_0 \& \omega_1 \& \omega_2 \& \omega_3)$ AR1	stable		
MID001	AND	Φ Input	LoC	MID	If both gyro AR1 is mid, AND one of the measured rotor speed's AR1 is EWS, THEN output is mid.
	AND	q AR1	mid		
	AND	r AR1	mid		
	AND	$(\omega_0 \| \omega_1 \| \omega_2 \| \omega_3)$ AR1	EWS		
MID002	AND	Φ Input	LoC	MID	If both gyro AR1 is mid/stable, AND one of the measured rotor speed's AR1 is EWS, THEN output is mid.
	AND	q AR1	mid		
	AND	r AR1	stable		
	AND	$(\omega_0 \| \omega_1 \| \omega_2 \| \omega_3)$ AR1	EWS		
MID002	AND	Φ Input	LoC	MID	If both gyro AR1 is mid/stable, AND one of the measured rotor speed's AR1 is EWS, THEN output is mid.
	AND	q AR1	stable		
	AND	r AR1	mid		
	AND	$(\omega_0 \| \omega_1 \| \omega_2 \| \omega_3)$ AR1	EWS		
MID003	AND	Φ Input	LoC	MID	If both gyro AR1 is mid, AND both gyro IQR are EWS, THEN output is mid.
	AND	q AR1	mid		
	AND	r AR1	mid		
	AND	q IQR	EWS		
	AND	r IQR	EWS		

Table 9: Fuzzy Rules for Roll-LoC. Columns: (1) Rule ID – identifies each rule, (2) Logic Operator – the logical connector used within the rule ('&' for 'AND', '∼' for 'NOT', '|' for 'OR'), (3 & 4) Antecedents & Membership Values – the variables and their respective membership value to activate the rule, (5) Consequent Membership Value – the outcome's degree of truth, (6) Rule Description – an elaboration of the rule's application. (Continued)

MID004	AND	Φ Input	LoC	MID	If gyro AR1 is LoC, and ALL measured rotor speed's AR1 is stable, THEN output is mid
	AND	q AR1	LoC		
	AND	$(\omega_0 \& \omega_1 \& \omega_2 \& \omega_3)$ AR1	stable		
MID004	AND	Φ Input	LoC	MID	If gyro AR1 is LoC, and ALL measured rotor speed's AR1 is stable, THEN output is mid
	AND	r AR1	LoC		
	AND	$(\omega_0 \& \omega_1 \& \omega_2 \& \omega_3)$ AR1	stable		
MID005	AND	Φ Input	LoC	MID	If gyro AR1 is EWS, and ALL measured rotor speed's AR1 is stable, THEN output is mid
	AND	q AR1	EWS		
	AND	$(\omega_0 \& \omega_1 \& \omega_2 \& \omega_3)$ AR1	stable		
MID005	AND	Φ Input	LoC	MID	If gyro AR1 is EWS, and ALL measured rotor speed's AR1 is stable, THEN output is mid
	AND	r AR1	EWS		
	AND	$(\omega_0 \& \omega_1 \& \omega_2 \& \omega_3)$ AR1	stable		
MID006	AND	Φ Input	LoC	MID	If both gyro AR1 is mid, AND one of the measured rotor speed's AR1 is EWS THEN output is mid.
	AND	q AR1	mid		
	AND	r AR1	mid		
	AND	$(\omega_0 \| \omega_1 \| \omega_2 \| \omega_3)$ AR1	EWS		
EWS001	AND	Φ Input	LoC	EWS	If both gyro AR1 is mid/EWS, AND both gyro IQR are EWS, THEN output is EWS.
	AND	q AR1	mid		
	AND	r AR1	EWS		
	AND	q IQR	EWS		
	AND	r IQR	EWS		
EWS001	AND	Φ Input	LoC	EWS	If both gyro AR1 is mid/EWS, AND both gyro IQR are EWS, THEN output is EWS.
	AND	q AR1	EWS		
	AND	r AR1	mid		
	AND	q IQR	EWS		
	AND	r IQR	EWS		
EWS002	AND	Φ Input	LoC	EWS	If both gyro AR1 is EWS, AND one of the measured rotor speed's AR1 is EWS, THEN output is EWS.
	AND	q AR1	EWS		
	AND	r AR1	EWS		
	AND	$(\omega_0 \| \omega_1 \| \omega_2 \| \omega_3)$ AR1	EWS		
EWS003	AND	Φ Input	LoC	EWS	If both gyro AR1 is EWS/mid, AND one of the measured rotor speed's AR1 is EWS, THEN output is EWS.
	AND	q AR1	EWS		
	AND	r AR1	mid		
	AND	$(\omega_0 \| \omega_1 \| \omega_2 \| \omega_3)$ AR1	EWS		

Table 9: Fuzzy Rules for Roll-LoC. Columns: (1) Rule ID – identifies each rule, (2) Logic Operator – the logical connector used within the rule ('&' for 'AND', '∼' for 'NOT', '|' for 'OR'), (3 & 4) Antecedents & Membership Values – the variables and their respective membership value to activate the rule, (5) Consequent Membership Value – the outcome's degree of truth, (6) Rule Description – an elaboration of the rule's application. (Continued)

EWS003	AND	Φ Input	LoC	EWS	If both gyro AR1 is EWS/mid, AND one of the measured rotor speed's AR1 is EWS, THEN output is EWS.
	AND	q AR1	mid		
	AND	r AR1	EWS		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
EWS004	AND	Φ Input	LoC	EWS	If both gyro AR1 is EWS/stable, AND one of the measured rotor speed's AR1 is EWS, THEN output is EWS.
	AND	q AR1	EWS		
	AND	r AR1	stable		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
EWS004	AND	Φ Input	LoC	EWS	If both gyro AR1 is EWS/stable, AND one of the measured rotor speed's AR1 is EWS, THEN output is EWS.
	AND	q AR1	stable		
	AND	r AR1	EWS		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
LOC001	AND	Φ Input	LoC	LoC	If gyro AR1 is LoC, AND one of the measured rotor speed's AR1 is EWS, THEN output is LoC.
	AND	q AR1	LoC		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		
LOC001	AND	Φ Input	LoC	LoC	If gyro AR1 is LoC, AND one of the measured rotor speed's AR1 is EWS, THEN output is LoC.
	AND	r AR1	LoC		
	AND	$(\omega_0\|\omega_1\|\omega_2\|\omega_3)$ AR1	EWS		

A.4 - CSD Analysis for DataCan(TinyWhoop) Flights

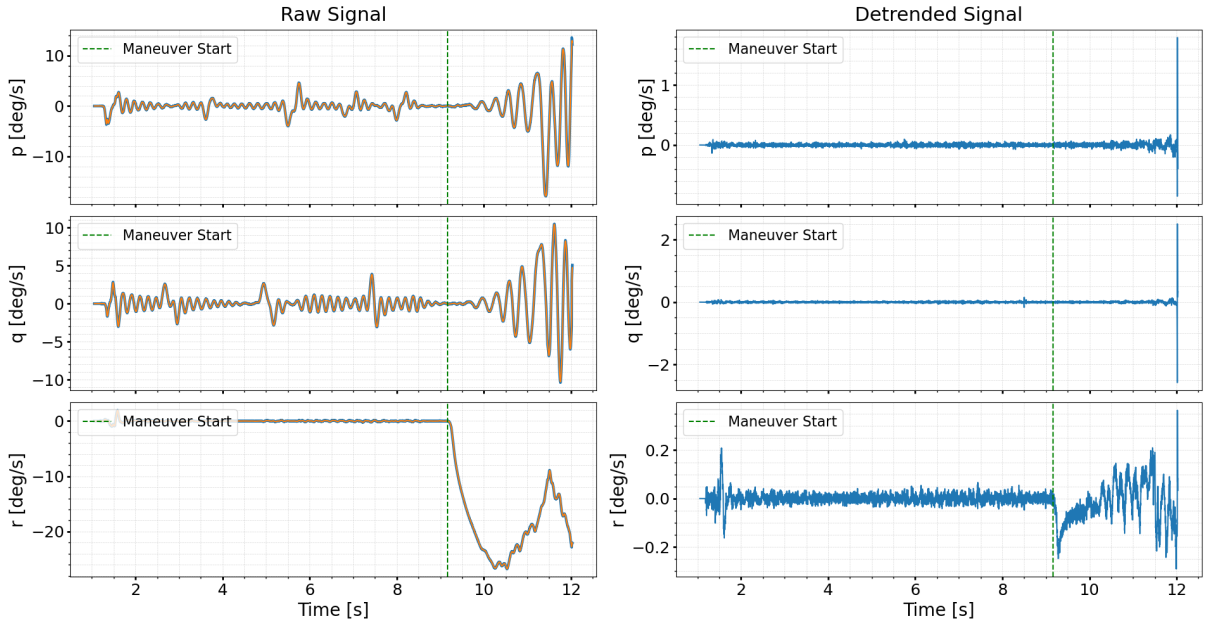

Fig. 33 Yaw-LoC (DataCan) gyro rates detrending results. Each row denotes a rotational rate axis, the first column depicts the original signal in blue, overlaid with the estimated long-term trend in orange. The second column depicts the detrended signal. Both column features a green dotted line indicating the commencement of the LoC maneuver.

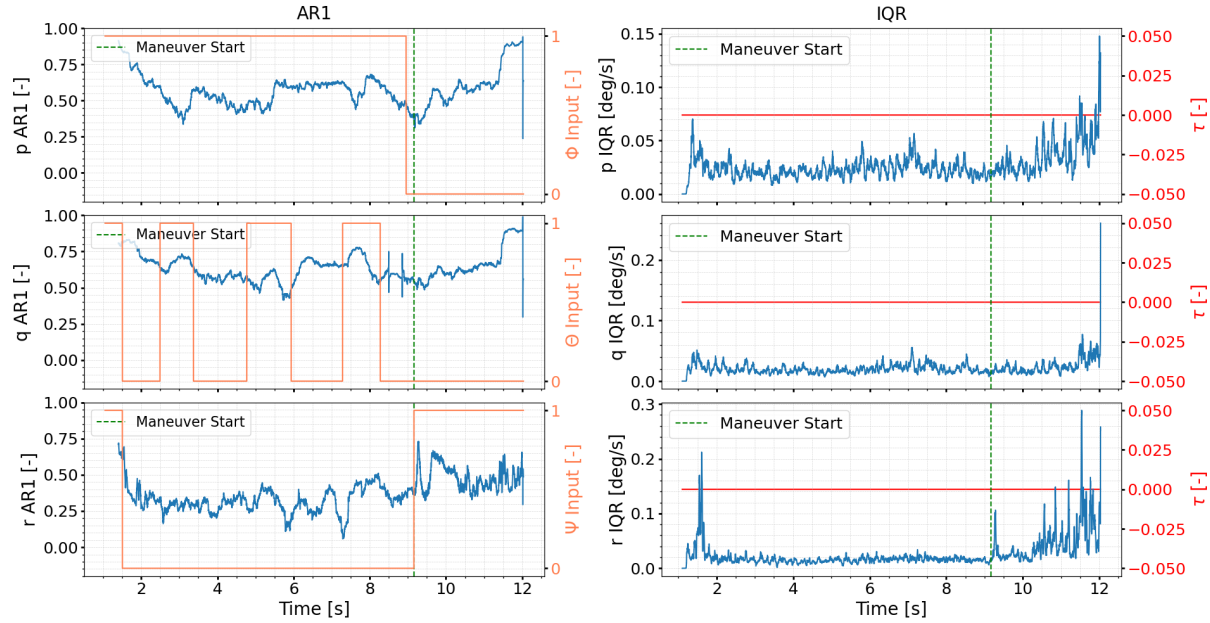

Fig. 34 Yaw-LoC (DataCan) gyro rates CSD results. Each row denotes a rotational rate axis, the first column depicts the AR1 time-series in blue, overlaid with the stick neutrality condition in orange. The second column depicts the IQR time series in blue, overlaid with the lag indicator signal τ. Both column features a green dotted line indicating the commencement of the LoC maneuver.

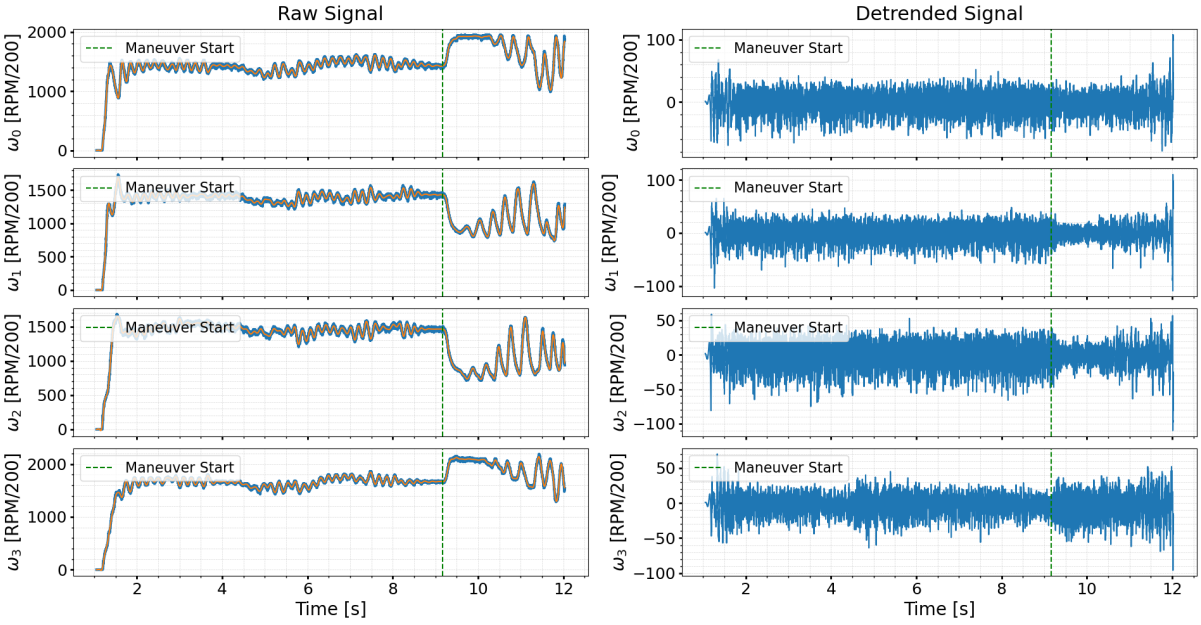

Fig. 35 Yaw-LoC (DataCan) measured rotor RPMs detrending results. Each row denotes a rotor, the first column depicts the original signal in blue, overlaid with the estimated long-term trend in orange. The second column depicts the detrended signal. Both column features a green dotted line indicating the commencement of the LoC maneuver.

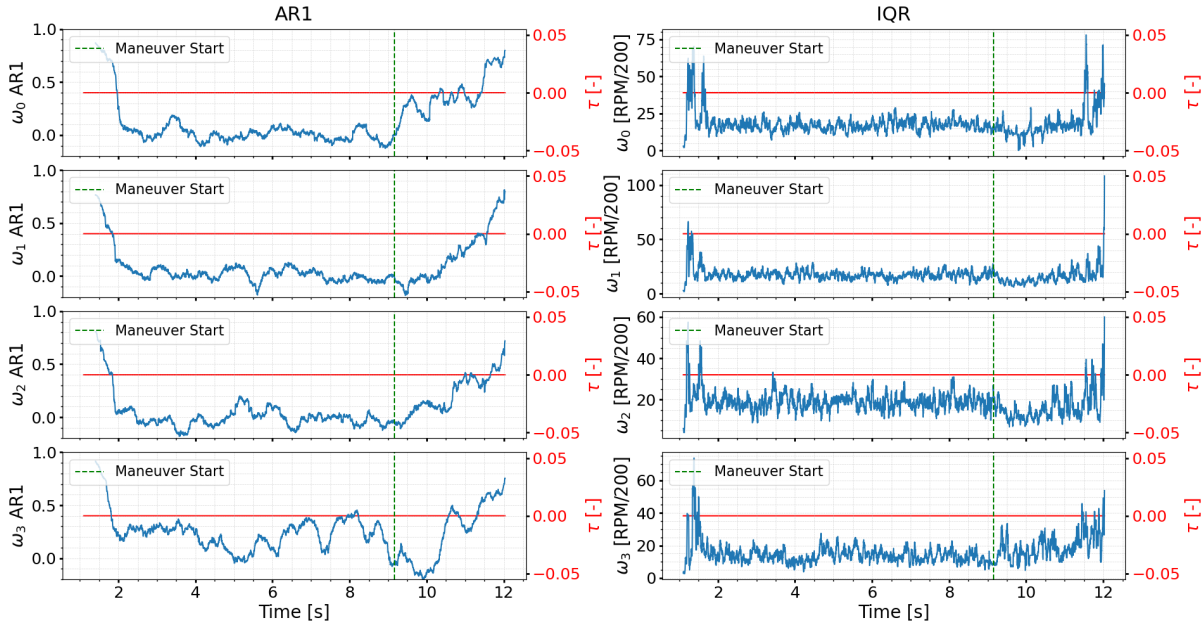

Fig. 36 Yaw-LoC (DataCan) measured rotor RPMs CSD results. Each row denotes a rotor, the first column depicts the AR1 time-series in blue. The second column depicts the IQR time series in blue. Both column features a green dotted line indicating the commencement of the LoC maneuver, and a red line showing the lag indicator signal τ.

A.5 - Fuzzy Logic Inference System Verification

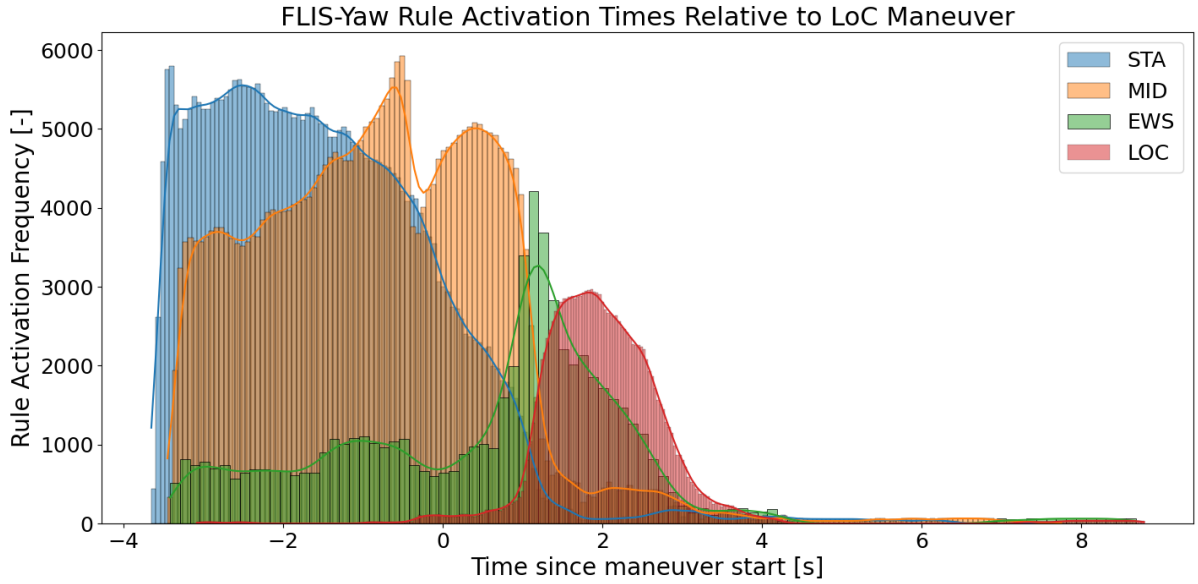

Fig. 37 Histogram of FLIS-Yaw activation times relative to the initiation of the yaw maneuver. Each rules are categorized into four groups: Stable (STA), Mid Stable (MID), Early Warning Signal (EWS), Loss of Control (LoC).

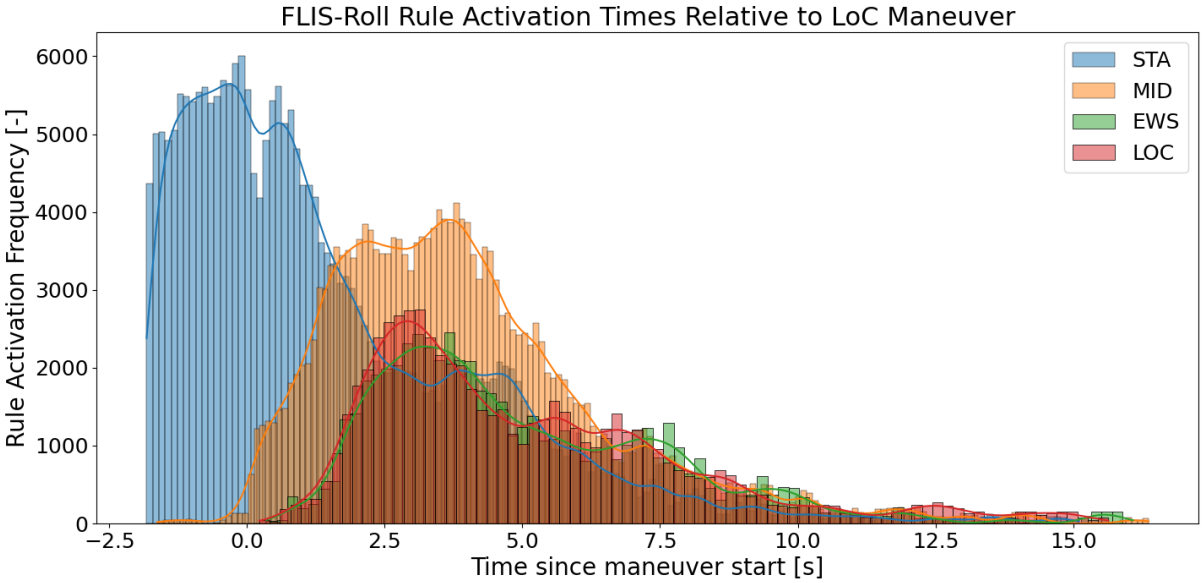

Fig. 38 Histogram of FLIS-Roll activation times relative to the initiation of the yaw maneuver. Each rules are categorized into four groups: Stable (STA), Mid Stable (MID), Early Warning Signal (EWS), Loss of Control (LoC).

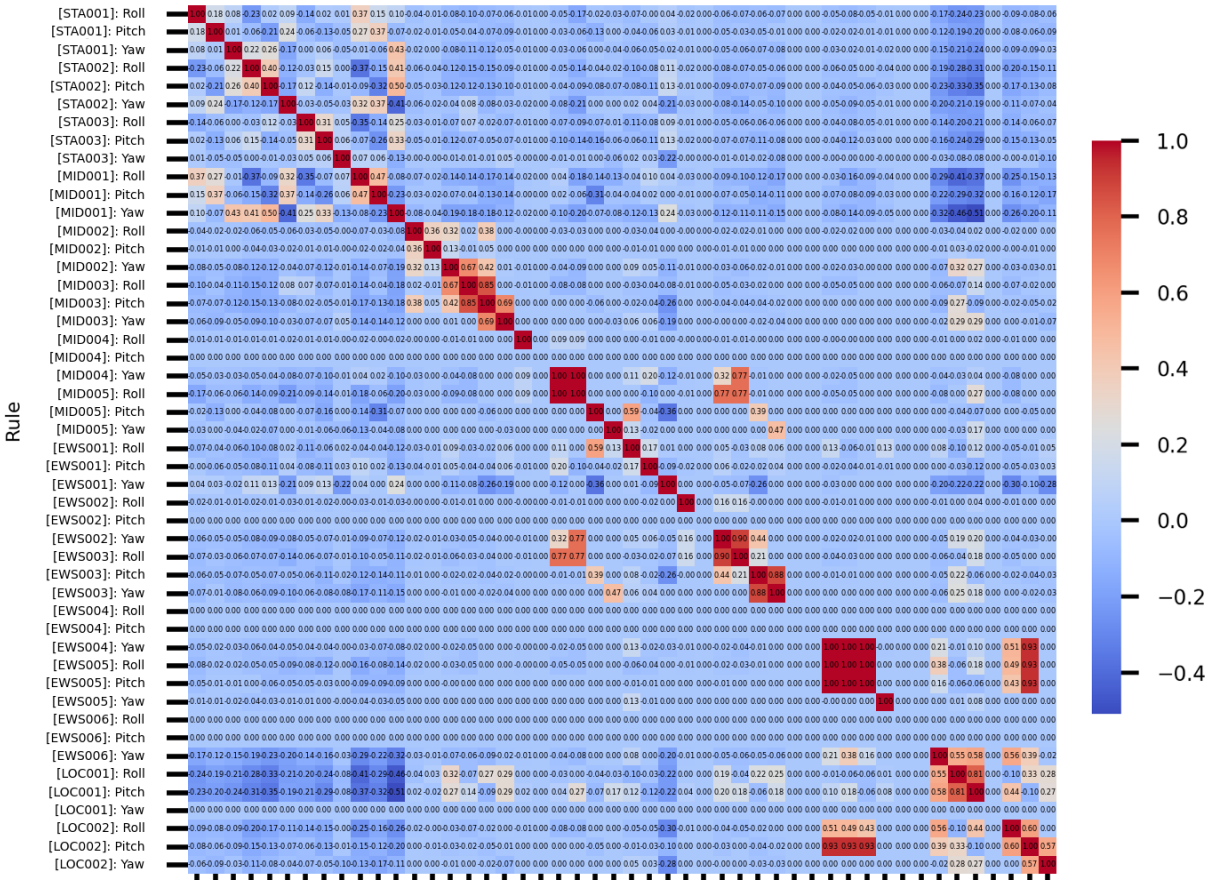

Fig. 39 Correlation Matrix of Fuzzy Rules for Yaw-LoC

References

[1] "Commercial Drone Market Size, Share & Trends Analysis Report By Product, By Application, By End-use, By Propulsion Type, By Range, By Operating Mode, By Endurance, By Region, And Segment Forecasts, 2023 - 2030," Technical Report 978-1-68038-482-6, Grand View Research, Inc., ???? URL `https://www.grandviewresearch.com/industry-analysis/global-commercial-drones-market`.

[2] Wild, G., Murray, J., and Baxter, G., "Exploring Civil Drone Accidents and Incidents to Help Prevent Potential Air Disasters," *Aerospace*, Vol. 3, No. 3, 2016, p. 22. https://doi.org/10.3390/aerospace3030022, URL http://www.mdpi.com/2226-4310/3/3/22.

[3] Zhang, Y., De Visser, C. C., and Chu, Q. P., "Database Building and Interpolation for a Safe Flight Envelope Prediction System," *2018 AIAA Information Systems-AIAA Infotech @ Aerospace*, American Institute of Aeronautics and Astronautics, Kissimmee, Florida, 2018. https://doi.org/10.2514/6.2018-1635, URL https://arc.aiaa.org/doi/10.2514/6.2018-1635.

[4] Zhang, Y., Huang, Y., Chu, Q., and de Visser, C. C., "Database-Driven Safe Flight-Envelope Protection for Impaired Aircraft," *Journal of Aerospace Information Systems*, Vol. 18, No. 1, 2021, pp. 14–25. https://doi.org/10.2514/1.I010846, URL https://arc.aiaa.org/doi/10.2514/1.I010846, publisher: American Institute of Aeronautics and Astronautics.

[5] van Oort, E. R., Chu, Q. P., and Mulder, J. A., "Maneuver Envelope Determination through Reachability Analysis," *Advances in Aerospace Guidance, Navigation and Control*, edited by F. Holzapfel and S. Theil, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 91–102.

[6] Lombaerts, T., Schuet, S., Wheeler, K., Acosta, D. M., and Kaneshige, J., "Safe Maneuvering Envelope Estimation based on a Physical Approach," *AIAA Guidance, Navigation, and Control (GNC) Conference*, American Institute of Aeronautics and Astronautics, Boston, MA, 2013. https://doi.org/10.2514/6.2013-4618, URL https://arc.aiaa.org/doi/10.2514/6.2013-4618.

[7] Chongvisal, J., Tekles, N., Xargay, E., Talleur, D., Kirlik, A., and Hovakimyan, N., "Loss-of-Control Prediction and Prevention for NASA's Transport Class Model," *AIAA Guidance, Navigation, and Control Conference*, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2014. https://doi.org/10.2514/6.2014-0784, URL https://arc.aiaa.org/doi/10.2514/6.2014-0784.

[8] Lu, Z., Hong, H., Gerdts, M., and Holzapfel, F., "Flight Envelope Prediction via Optimal Control-Based Reachability Analysis," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 1, 2022, pp. 185–195. https://doi.org/10.2514/1.G006219, URL https://doi.org/10.2514/1.G006219, publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.G006219.

[9] Sun, S., and De Visser, C. C., "Quadrotor Safe Flight Envelope Prediction in the High-Speed Regime: A Monte-Carlo Approach," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, San Diego, California, 2019. https://doi.org/10.2514/6.2019-0948, URL https://arc.aiaa.org/doi/10.2514/6.2019-0948.

[10] Kaffa, L., "Safe Flight Envelope Prediction and Experimental Validation of Multirotor UAV: An Analysis on the Effects of Longitudinal Center of Gravity Position and Actuator Dynamics on Quadcopter Flight Envelopes," Master's thesis, Delft University of Technology, Delft, Jul. 2023. URL https://repository.tudelft.nl/islandora/object/uuid%3A5d3f5432-3801-4b5b-9dcf-6db1ce2cb150.

[11] van der Pluijm, A., "Early Warning Signals for Loss of Control Prediction of a Damaged Quadcopter," Master's thesis, Delft University of Technology, Delft, Jan. 2020.

[12] Hoppenbrouwer, T., "Loss of Control Prediction for Quadrotors with Single Rotor Failure," Master's thesis, Delft University of Technology, Delft, Jun. 2023.

[13] Sun, S., Sijbers, L., Wang, X., and de Visser, C., "High-Speed Flight of Quadrotor Despite Loss of Single Rotor," *IEEE Robotics and Automation Letters*, Vol. 3, No. 4, 2018, pp. 3201–3207. https://doi.org/10.1109/LRA.2018.2851028, URL https://ieeexplore.ieee.org/document/8398406, conference Name: IEEE Robotics and Automation Letters.

[14] Sun, S., Wang, X., Chu, Q., and Visser, C. d., "Incremental Nonlinear Fault-Tolerant Control of a Quadrotor With Complete Loss of Two Opposing Rotors," *IEEE Transactions on Robotics*, Vol. 37, No. 1, 2021, pp. 116–130. https://doi.org/10.1109/TRO.2020.3010626, URL https://ieeexplore.ieee.org/document/9160894, conference Name: IEEE Transactions on Robotics.

[15] Hamadi, H., "Fault-tolerant control of a multirotor unmanned aerial vehicle under hardware and software failures," PhD Thesis, Université de Technologie de Compiègne, Jul. 2020.

[16] Lu, P., and van Kampen, E.-J., "Active fault-tolerant control for quadrotors subjected to a complete rotor failure," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4698–4703. https://doi.org/10.1109/IROS.2015.7354046.

[17] Altena, A. V., van Beers, J. J., and de Visser, C. C., "Loss-of-Control Prediction of a Quadcopter Using Recurrent Neural Networks," *Journal of Aerospace Information Systems*, Vol. 20, No. 10, 2023, pp. 648–659. https://doi.org/10.2514/1.I011231, URL https://arc.aiaa.org/doi/10.2514/1.I011231, publisher: American Institute of Aeronautics and Astronautics.

[18] Lenton, T. M., Held, H., Kriegler, E., Hall, J. W., Lucht, W., Rahmstorf, S., and Schellnhuber, H. J., "Tipping elements in the Earth's climate system," *Proceedings of the National Academy of Sciences*, Vol. 105, No. 6, 2008, pp. 1786–1793. https://doi.org/10.1073/pnas.0705414105, URL https://www.pnas.org/doi/full/10.1073/pnas.0705414105, publisher: Proceedings of the National Academy of Sciences.

[19] Scheffer, M., Carpenter, S., Foley, J. A., Folke, C., and Walker, B., "Catastrophic shifts in ecosystems," *Nature*, Vol. 413, No. 6856, 2001, pp. 591–596. https://doi.org/10.1038/35098000, URL https://www.nature.com/articles/35098000, number: 6856 Publisher: Nature Publishing Group.

[20] Sornette, D., and Johansen, A., "Large financial crashes," *Physica A: Statistical Mechanics and its Applications*, Vol. 245, No. 3, 1997, pp. 411–422. https://doi.org/10.1016/S0378-4371(97)00318-X, URL https://www.sciencedirect.com/science/article/pii/S037843719700318X.

[21] Venegas, J. G., Winkler, T., Musch, G., Vidal Melo, M. F., Layfield, D., Tgavalekos, N., Fischman, A. J., Callahan, R. J., Bellani, G., and Scott Harris, R., "Self-organized patchiness in asthma as a prelude to catastrophic shifts," *Nature*, Vol. 434, No. 7034, 2005, pp. 777–782. https://doi.org/10.1038/nature03490, URL https://www.nature.com/articles/nature03490, number: 7034 Publisher: Nature Publishing Group.

[22] Schroeder, M., *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*, Dover Publications, Mineola, N.Y, 2009.

[23] Scheffer, M., Carpenter, S. R., Lenton, T. M., Bascompte, J., Brock, W., Dakos, V., Van De Koppel, J., Van De Leemput, I. A., Levin, S. A., Van Nes, E. H., Pascual, M., and Vandermeer, J., "Anticipating Critical Transitions," *Science*, Vol. 338, No. 6105, 2012, pp. 344–348. https://doi.org/10.1126/science.1225244, URL https://www.science.org/doi/10.1126/science.1225244.

[24] Zeng, C., and Wang, H., "Noise and large time delay: Accelerated catastrophic regime shifts in ecosystems," *Ecological Modelling*, Vol. 233, 2012, pp. 52–58. https://doi.org/10.1016/j.ecolmodel.2012.03.025, URL https://www.sciencedirect.com/science/article/pii/S030438001200141X.

[25] Dakos, V., van Nes, E. H., Donangelo, R., Fort, H., and Scheffer, M., "Spatial correlation as leading indicator of catastrophic shifts," *Theoretical Ecology*, Vol. 3, No. 3, 2010, pp. 163–174. https://doi.org/10.1007/s12080-009-0060-6, URL https://doi.org/10.1007/s12080-009-0060-6.

[26] Boettiger, C., Ross, N., and Hastings, A., "Early warning signals: the charted and uncharted territories," *Theoretical Ecology*, Vol. 6, No. 3, 2013, pp. 255–264. https://doi.org/10.1007/s12080-013-0192-6, URL http://link.springer.com/10.1007/s12080-013-0192-6.

[27] Van Nes, E. H., and Scheffer, M., "Slow recovery from perturbations as a generic indicator of a nearby catastrophic shift," *The American Naturalist*, Vol. 169, No. 6, 2007, pp. 738–747. Publisher: The University of Chicago Press.

[28] Livina, V. N., and Lenton, T. M., "A modified method for detecting incipient bifurcations in a dynamical system," *Geophysical research letters*, Vol. 34, No. 3, 2007. Publisher: Wiley Online Library.

[29] Dakos, V., Scheffer, M., van Nes, E. H., Brovkin, V., Petoukhov, V., and Held, H., "Slowing down as an early warning signal for abrupt climate change," *Proceedings of the National Academy of Sciences*, Vol. 105, No. 38, 2008, pp. 14308–14312. https://doi.org/10.1073/pnas.0802430105, URL https://www.pnas.org/doi/full/10.1073/pnas.0802430105, publisher: Proceedings of the National Academy of Sciences.

[30] Carpenter, S. R., and Brock, W. A., "Rising variance: a leading indicator of ecological transition," *Ecology letters*, Vol. 9, No. 3, 2006, pp. 311–318. Publisher: Wiley Online Library.

[31] Diks, C., Hommes, C., and Wang, J., "Critical slowing down as an early warning signal for financial crises?" *Empirical Economics*, Vol. 57, No. 4, 2019, pp. 1201–1228. https://doi.org/10.1007/s00181-018-1527-3, URL https://doi.org/10.1007/s00181-018-1527-3.

[32] Ives, A. R., "Measuring resilience in stochastic systems," *Ecological Monographs*, Vol. 65, No. 2, 1995, pp. 217–233. Publisher: Wiley Online Library.

[33] Wen, H., Ciamarra, M. P., and Cheong, S. A., "How one might miss early warning signals of critical transitions in time series data: A systematic study of two major currency pairs," *PLOS ONE*, Vol. 13, No. 3, 2018, p. e0191439. https://doi.org/10.1371/journal.pone.0191439, URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0191439, publisher: Public Library of Science.

[34] Dakos, V., Nes, E., and Scheffer, M., "Flickering as an early warning signal," *Theoretical Ecology*, Vol. 6, 2013. https://doi.org/10.1007/s12080-013-0186-4.

[35] Carpenter, S. R., Ludwig, D., and Brock, W. A., "Management of eutrophication for lakes subject to potentially irreversible change," *Ecological applications*, Vol. 9, No. 3, 1999, pp. 751–771. Publisher: Wiley Online Library.

[36] Kalogirou, S. A., "Chapter eleven - Designing and Modeling Solar Energy Systems," *Solar Energy Engineering*, edited by S. A. Kalogirou, Academic Press, Boston, 2009, pp. 553–664. https://doi.org/https://doi.org/10.1016/B978-0-12-374501-9.00011-X, URL https://www.sciencedirect.com/science/article/pii/B978012374501900011X.

[37] Ross, T. J., *Fuzzy Logic with Engineering Applications*, 2nd ed., McGraw-Hill Inc.,US, New York, 1994.

[38] van Beers, J. J., and de Visser, C. C., "Peaking into the Black-box: Prediction Intervals Give Insight into Data-driven Quadrotor Model Reliability," *AIAA SCITECH 2023 Forum*, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2023. https://doi.org/10.2514/6.2023-0125, URL https://arc.aiaa.org/doi/10.2514/6.2023-0125.

[39] Kersbergen, S. J. K., "Quantifying Loss-Of-Control of Quadrotors," Master's thesis, Delft University of Technology, Delft, Oct. 2018.

[40] Sharma, S., Dijkstra, T., and Prasad, R. V., "Open Gimbal: A 3 Degrees of Freedom Open Source Sensing and Testing Platform for Nano- and Micro-UAVs," *IEEE Sensors Letters*, Vol. 7, No. 9, 2023, pp. 1–4. https://doi.org/10.1109/LSENS.2023.3307121, URL https://ieeexplore.ieee.org/document/10225720/.

[41] Badr, S., Mehrez, O., and Kabeel, A. E., "A design modification for a quadrotor UAV: modeling, control and implementation," *Advanced Robotics*, Vol. 33, No. 1, 2019, pp. 13–32. https://doi.org/10.1080/01691864.2018.1556116, URL https://doi.org/10.1080/01691864.2018.1556116, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01691864.2018.1556116.

[42] Dakos, V., Carpenter, S. R., Brock, W. A., Ellison, A. M., Guttal, V., Ives, A. R., Kéfi, S., Livina, V., Seekell, D. A., Nes, E. H. v., and Scheffer, M., "Methods for Detecting Early Warnings of Critical Transitions in Time Series Illustrated Using Simulated Ecological Data," *PLOS ONE*, Vol. 7, No. 7, 2012, p. e41010. https://doi.org/10.1371/journal.pone.0041010, URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0041010, publisher: Public Library of Science.

[43] Quax, R., Kandhai, D., and Sloot, P. M. A., "Information dissipation as an early-warning signal for the Lehman Brothers collapse in financial time series," *Scientific Reports*, Vol. 3, 2013, p. 1898. https://doi.org/10.1038/srep01898.

[44] Nannes, B., Quax, R., Ashikaga, H., Hocini, M., Dubois, R., Bernus, O., and Haïssaguerre, M., "Early signs of critical slowing down in heart surface electrograms of ventricular fibrillation victims," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 12140 LNCS, 2020, pp. 334–347. https://doi.org/10.1007/978-3-030-50423-6_25, iSBN: 9783030504229.

[45] Podolsky, D., and Turitsyn, K., "Critical slowing-down as indicator of approach to the loss of stability," *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, Venice, Italy, 2014, pp. 19–24. https://doi.org/10.1109/SmartGridComm.2014.7007616, URL http://ieeexplore.ieee.org/document/7007616/.

[46] Hermance, J. F., "Interannual and Interseasonal Variations in Monthly Rainfall," *Historical Variability of Rainfall in the African East Sahel of Sudan: Implications for Development*, edited by J. F. Hermance, Springer International Publishing, Cham, 2014, pp. 43–57. https://doi.org/10.1007/978-3-319-00575-1_3, URL https://doi.org/10.1007/978-3-319-00575-1_3.

[47] Brunke, L., "Forecasting Earnings Variance: Quantiles-Based Vs. Residuals-Based Approaches," , Mar. 2023. https://doi.org/10.2139/ssrn.4404839, URL https://papers.ssrn.com/abstract=4404839.

[48] Rich, R., and Tracy, J., "A Closer Look at the Behavior of Uncertainty and Disagreement: Micro Evidence from the Euro Area," *Journal of Money, Credit and Banking*, Vol. 53, No. 1, 2021, pp. 233–253. https://doi.org/10.1111/jmcb.12728, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/jmcb.12728, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jmcb.12728.

[49] Adil, O., Ali, A., and Sumait, B., "Comparison between the Effects of Different Types of Membership Functions on Fuzzy Logic Controller Performance," *International Journal of Emerging Engineering Research and Technology*, Vol. 3, 2015, pp. 76–83.

[50] Hantz, P., and Lázár, Z. I., "Precession Intuitively Explained," *Frontiers in Physics*, Vol. 7, 2019. https://doi.org/10.3389/fphy.2019.00005, URL https://www.frontiersin.org/articles/10.3389/fphy.2019.00005, publisher: Frontiers.

[51] Shao, P.-j., Dong, W.-h., Sun, X.-x., Ding, T.-j., and Zou, Q., "Dynamic surface control to correct for gyroscopic effect of propellers on quadrotor," *2015 IEEE International Conference on Information and Automation*, 2015, pp. 2971–2976. https://doi.org/10.1109/ICInfA.2015.7279797, URL https://ieeexplore.ieee.org/document/7279797.

Part II

Preliminary Analysis

*This part has been assessed for the course AE4020 Literature Study.

3

Literature Review

3.1. UAV Safety

Loss of Control (LoC) emerges as the predominant culprit behind quad-rotor accidents [4]. In 2000, the Commercial Aviation Safety Team founded the Joint Safety Analysis Team (JSAT), to investigate data on LOC events and recommend interventions that could lead to a 80% reduction in fatal accidents rate by 2007. JSAT defines LOC as "significant, unintended departure of the aircraft from controlled flight, the operational flight envelope, or usual flight attitudes, including ground events" [20]. Traditionally, the flight envelope is defined as the set of limits or boundaries that dictate where an aircraft (or drone) can safely operate [21]. A traditional flight envelope (depicted in Figure 3.1) typically consists of the maximum and minimum airspeed, as a function of different load factors. The envelope shows which combinations of these states allow a safe operation, and which will lead to structural damage, or even LoC.

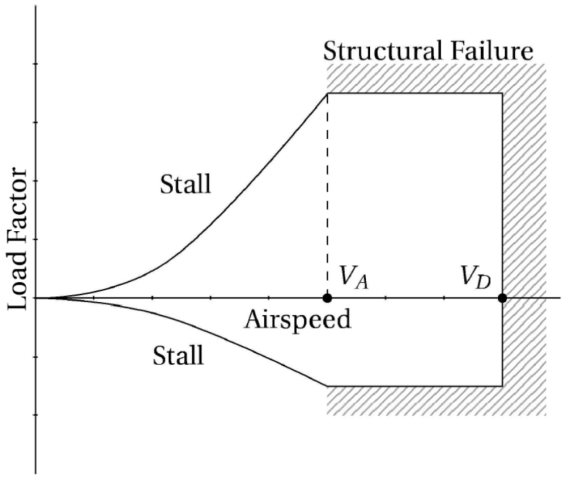

Figure 3.1: Traditional flight envelope [13]

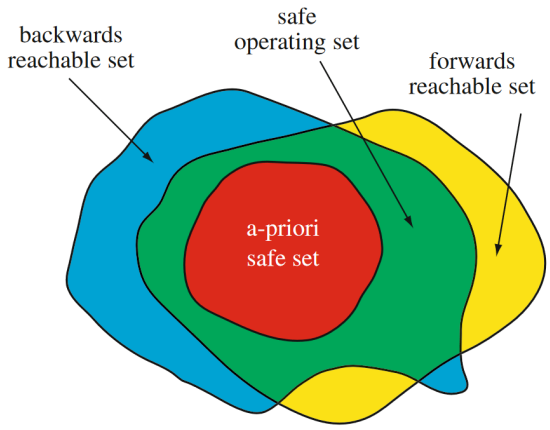

Figure 3.2: SFE as an intersection of forward and backward reachable set [7]

However, this definition is limited to a vehicle operating at nominal conditions, and primarily tailored to conventional aircraft. As such, an extended definition known as the Safe Flight Envelope (SFE) is defined as "The part of the state space for which safe operation of the aircraft and safety of its cargo can be guaranteed and externally posed constraints will not be violated" [22]. SFE applied to LoC prevention can be divided in two parts : first the SFE itself should be defined after which a controller that guarantees that the system remains in the SFE needs to be designed. Now the difficulty lies in the first part of the SFE, the determination of the SFE is not such a straightforward process. The reason being that the SFE is not only dependent on the dynamics of the aircraft or UAV, but also the conditions it's operating in. Traditionally, the dynamic flight envelope was researched through extensive wind tunnel testing and CFD analysis, in which the results were validated using flight tests [22], or through analytical methods using aircraft models, however this is computationally expensive. Another method was using bifurcation analysis to study the steady-state nonlinear dynamics of the vehicle, and to compute the entire set of steady state

solutions [23]. However this also requires the system to be in a quasi steady-state condition, which may not always be the case in operation, especially for drones.

More recently, a promising technique for SFE determination has been explored through the use of reachability analysis, a data-driven methods towards SFE determination[6]. The concept of the reachable set in aviation, applicable to both aircraft and UAVs, encompasses all attainable states within a defined time span, taking into account initial conditions. Subsequently, the Safe Flight Envelope (SFE), is established as the overlap between the forward and backward reachable sets (assuming time reversibility) as can be seen on Figure 3.2. Consider a system as Equation 3.1

$$\begin{aligned} \dot{\mathbf{x}}(t) =& f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) \\ & x(0) \in S_0 \text{ or } x(t_f) \in T_f t \in [0, t_f] \end{aligned} \tag{3.1}$$

where $0 \leq t_f < \infty$, $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{u} \in U \subset \mathbb{R}^m$ is the control input, $\mathbf{d} \in D \subset \mathbb{R}^q$ is the disturbance input, S_0 is a set of initial states, and $T_f = \{x \in \mathbb{R}^n | T(x) \leq 0\}$ is a set of target states. The dynamics defined by the equation above can be evolved backwards and forwards in time resulting in the two reachable sets respectively [22].

Definition 2.1 (Backwards Reachable Set) - *The backwards reachable set $T(\tau)$ at a time $\tau(0 < \tau \leq t_f)$, of the system* (Equation 3.1) *for the target set T_f, is the set of all states $x(\tau)$, such that there exists a control input $u(t) \in U(\tau \leq t \leq t_f)$, for all disturbance inputs $d(t) \in D(\tau \leq t \leq t_f)$, for which some $x(t_f) \in T_f$ are reachable from $x(\tau)$ along a trajectory satisfying* Equation 3.1

Definition 2.2 (Forwards Reachable Set) - *The forwards reachable set $S(\tau)$ at a time $\tau(0 < \tau \leq t_f)$, of the system* (Equation 3.1) *for the initial set S_0, is the set of all states $x(\tau)$, such that there exists a control input $u(t) \in U(\tau \leq t \leq t_f)$, for all disturbance inputs $d(t) \in D(\tau \leq t \leq t_f)$, for which $x(\tau)$ is reachable from some $x(0) \in S_0$ along a trajectory satisfying* Equation 3.1

The reachable set theory has been used for the determination of the SFE for different aircraft and UAVs. However there still exists the problem of computational cost, and curse of dimensionality which refers to the computational challenges and increased complexity that arises with increasing dimensions, which makes this even harder. In addition to the dynamic conditions the vehicle operates in, the vehicle dynamics itself could also change. Some notable examples are:

1. Rotor Damage - The most common form of hardware damage is damage sustained by the propellers or the primary actuation mechanism, i.e., the rotor assembly itself [24] [25]. Small fractures or chips on the propeller blades can cause an imbalance on the lift generated by one of the rotors, which can create instabilities. In more severe instances, the complete failure of an actuator may occur, resulting in the loss of control in one of the axis, or a complete loss of control.
2. Battery Health - The condition of the battery is crucial because it determines the flight time and overall performance of the drone. A degraded battery can lead to rotor saturation, and provide less power, which ultimately alters the flight envelope.
3. Other Factors - Payload Changes, Sensor Calibration Errors, Electrical Interference, Misalignment of IMU, Gimbal, and Camera.

Therefore, due to the dynamic operating conditions, and potential changes due to the dynamics of the drone itself, a model-based approach for LoC detection and prevention is challenging to construct. Hence, another way of preventing vehicles operating outside of their SFE needs to be explored. If the primary objective of the research is to be able to detect when a drone is operating near the edge of the SFE, there is no strict requirement for identifying the SFE itself. The emphasis is on obtaining an indicator of proximity to the SFE. It is important to highlight that there are numerous modes and varying environmental conditions that can lead the drone near the edges of the SFE, hence there is no singular, standardised method for detecting this proximity. However, there exists some traits that are typically observed when a drone operates on the edge of its SFE, including the emergence of approaching instabilities and a consequential reduction in the degree of remaining control authority. Therefore, it becomes logical to establish a definition of Loss of Control (LoC) based on the duration and intensity of instabilities experienced by the drone, as well as the drone's recoverability at any given moment. Consequently, the research's objective is to investigate the possibility of defining such indicators of LoC using a data-driven approach.

3.2. Data Driven Methods for Anomaly Detection & Time-series Prediction

The core objective of our research aims to grapple with the notion of LoC, striving to formulate a generalised definition for LoC. As previously mentioned in Section 3.1, LoC is traditionally defined by determining the SFE for which any operation outside the SFE bounds is considered a LoC event. However, as the ultimate goal of this research is not for the determination of the SFE, rather to obtain an indicator of proximity to the edge of the SFE. This problem can be simplified into detecting approaching instabilities as an indicator of proximity to the edge of the SFE.

An important consideration that underscores the essence of our research lies in the choice of our approach. We have opted to navigate away from model-based methodologies, an approach that requires precise modelling of our system. The rationale behind this choice is due to the fact that a data-driven method allows for reduced model assumptions, which entails better adaptability to real-world conditions, but also allows for continuous improvement, integration of multiple data sources, and better scalability. This necessitates a shift towards data-driven techniques that leverage empirical data and analysis to predict upcoming instabilities, a practical approach that aligns with the constraints of real-world applications. Thus, in this chapter presents three data-driven methods for predicting upcoming instabilities : Critical Slowing Down Theory, Complexity Measures, and Machine Learning Methods.

As we delve into the subsequent sections of this chapter, we'll explore these data-driven methodologies. We aim to highlight their roles in addressing LoC prediction in the realm of drone technology, with one methodology grounded in theory, while the other relies solely on data-driven autonomy.

3.2.1. Critical Slowing Down (CSD) Theory

The behavior of numerous natural and human-made systems is influenced by multiple parameters that evolve over time. Their responses to external perturbations may vary: they may exhibit a smooth, gradual change in the systems state, or they may show little reaction within certain ranges of conditions, responding more as they approach a critical threshold. This situation, where a small perturbation can lead to an abrupt shift in the system's state when it reaches a critical threshold is commonly referred to as *tipping* or *critical transition*. Tipping events can be observed in real-world systems like climate systems [26], ecological systems [27], financial systems [28], and biological systems [29].

Predicting these critical transitions present significant challenges, as the system's state often exhibits minimal change prior to reaching the tipping point. Furthermore, models of complex systems typically lack the precision required for reliable predictions of where critical thresholds may occur. Interestingly however, research indicates that various systems typically exhibit common symptoms as they approach a critical point [30].

In particular, several research papers on ecological systems [31][32][33] have emphasized the occurrence of *Critical Slowing Down* (CSD) in empirical/time-series measurements as they approach a tipping point. CSD theory is based on the generic phenomenon that, in the vicinity of various tipping points, the rate at which a system recovers from small perturbations becomes very low [31]. It is important to note that not all transitions can show signs of CSD. This can be visualized in Figure 3.3, where the transitions are grouped into Rapid Regime Shift (abrupt changes in the state and conditions of system behavior), Bifurcations (qualitative changes in system behavior due to the passing of a threshold in underlying parameters or conditions), and CSD (observed behavior of slow system response to perturbation).

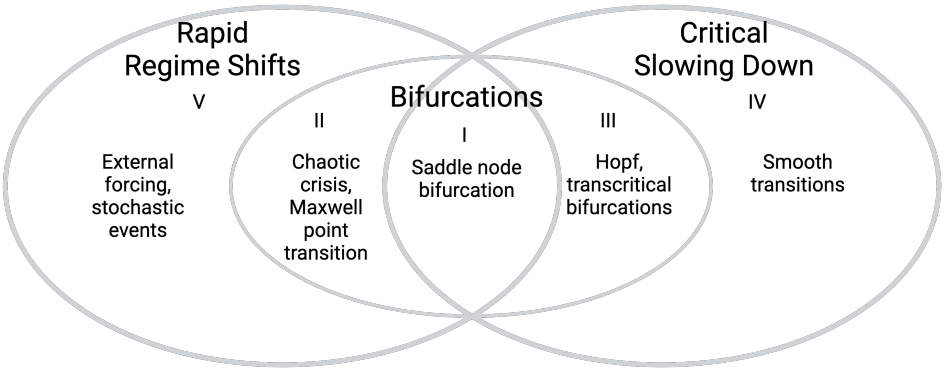

Figure 3.3: Intersecting domains of Rapid Regime Shifts, Bifurcations, and Critical Slowing Down. Domain I represents catastrophic bifurcations preceded by CSD, Domain II describes catastrophic bifurcations not preceded by CSD, Domain III represents non-catastrophic bifurcations preceded by CSD, Domain IV represents CSD in the absence of bifurcations or regime shifts, and finally Domain V represents catastrophic regime shifts without bifurcations or CSD. Smaller labels under each domain illustrate specific phenomena occurring within these categories, adapted from [34].

Figure 3.3 illustrates that not all regime shifts include bifurcations, and bifurcations can exist without exhibiting signs of CSD, and some regime shifts can also exhibit signs of CSD. Consequently, CSD should not be interpreted as a universal predictor of tipping points nor specific to approaching tipping point. Rather, CSD should be regarded as a *broad spectrum* indicator of potential fundamental changes in the current state [31]. It is worth noting that much of the prior work on CSD has been centered around its application in open-loop systems. To the best our knowledge, the successful application of CSD theory in closed-loop systems remains unexplored, opening new avenues for investigation.

To accurately apply CSD analysis, it is crucial to pre-process the data by detrending the signals. This process involves removing long-term trends or shifts from the time series data, thereby isolating the residual oscillations. Detrending is essential because it minimizes the influence of non-stationary trends on the analysis, allowing for a clearer detection of the slowing down phenomena that precede critical transitions.

Although CSD occurs for a class of different bifurcations (Figure 3.3), the following explanation will focus on the saddle-node (fold) bifurcation. This type is particularly intuitive to understand due to the *folded* nature of the system and the presence of distinct basins of attraction. To understand why CSD occurs at a fold bifurcation—where the system recovery behavior following a perturbation diminishes and approaches zero—consider the dynamics of a simple system:

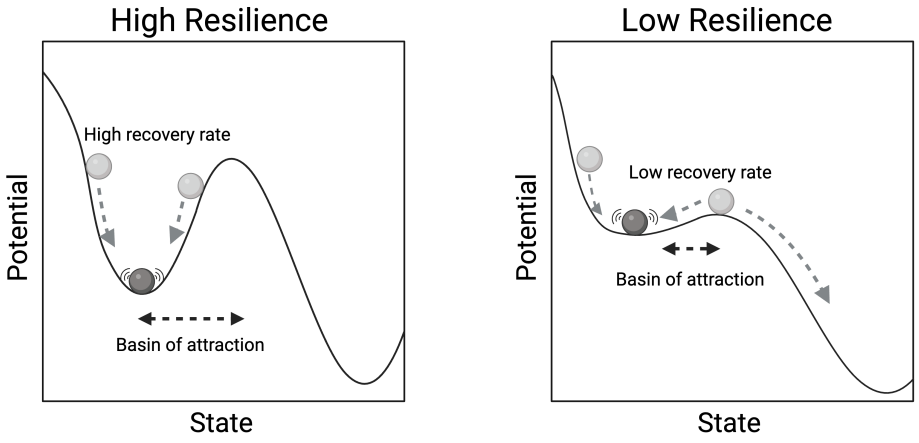

Figure 3.4: Reduced system resilience due to Critical Slowing Down (CSD) in a saddle-node bifurcation. The rate of recovery from minor perturbations is lower when the basin of attraction is smaller, and thus lower resilience (right side) compared to when it is larger, and hence higher resilience (left side).

$$\frac{dx}{dt} = r + x^2 \tag{3.2}$$

In this model, r represents the bifurcation parameter. When r is negative, there are two equilibrium points : $\bar{x}_{stable} = \sqrt{-r}$ and $\bar{x}_{unstable} = \sqrt{-r}$, with one being stable, and the other unstable which can be visualized in left figure of Figure 3.4, where the left basin represents the stable equilibrium and right basin represents the unstable equilibrium. As r approaches zero from the negative side, these two fixed points move closer to each other. At $r = 0$, they collide and annihilate, causing a catastrophic bifurcation causing sudden and irreversible change to a different state. This can be visualized on the right side of Figure 3.4, as r approaches 0, the system loses its resilience, meaning the basin of attraction is much smaller, and the slope of the basin is also smaller, effectively slowing its rate of recovery.

Now lets examine the dynamics near the tipping point, to analyze the stability of the equilibrium points, and how perturbations decay, linearizing the system round a fixed point x_0 by introducing a small perturbation ϵ such that $x = x_0 + \epsilon$:

$$\frac{d\epsilon}{dt} = \frac{d}{dt}(x - x_0) = \frac{dx}{dt} = r + (x_0 + \epsilon)^2 \tag{3.3}$$

Linearizing this equation using a first-order Taylor expansion yields :

$$\frac{d\epsilon}{dt} \approx r + x_0^2 + 2x_0\epsilon \tag{3.4}$$

Since x_0 is a fixed point, $r + x_0^2 = 0$, and the equation simplifes to:

$$\frac{d\epsilon}{dt} \approx 2x_0\epsilon \tag{3.5}$$

Here, the coefficient $2x_0$ is essentially the eigenvalue λ for the linearized system around the fixed point x_0 Now, as r approaches zero from the negative side, we can substitute x_{stable} into our expression for the eigenvalue:

$$\lambda = 2x_{stable} = 2(-\sqrt{-r}) \Rightarrow \lim_{r \to 0} \lambda = 2(-0) = 0 \tag{3.6}$$

Thus, as the system approaches a bifurcation, the parameter r approaches zero, and the eigenvalue associated with the stable fixed point approaches zero. Since the eigenvalue can be interpreted as the rate at which small perturbations to the system's state will grow or decay, this proves that as the system approaches a bifurcation, it's rate of recovery will diminish.

The primary implication of CSD is that after a small experimental perturbation, the decreased recovery rate can indicate the system's proximity to a bifurcation point [35]. While it may be impractical to monitor recovery rates of experimental perturbations systematically, given, that most natural systems regularly face natural perturbation, this can serve as an effective early warning signal for impending critical transitions. Notable indicators include an increase in lag-1 auto-correlation and variance in system fluctuations. This can be explained intuitively, since CSD reduces the system's rate of change, making its current state more like its past, effectively increasing its *memory* as measured by lag-1 auto-correlation [36] [37]. Additionally, as a tipping approaches, the variance in the system fluctuations also increased due to the decreased rate of recover, the disturbances' effects do not dissipate, leading to an increase in the variance of the state variable CSD [38].

This phenomenon can also be shown mathematically using a simple first-order auto-regressive model, where repeated disturbances every Δt leads to a progressively slower return to equilibrium, characterized by a recovery rate λ:

$$x_{n+1} - \bar{x} = e^{\lambda \Delta t}(x_n - \bar{x}) + \sigma\epsilon_n \tag{3.7}$$

$$y_{n+1} = e^{\lambda \Delta t} y_n + \sigma\epsilon_n \tag{3.8}$$

y_n represents the deviation of the state variable x from its equilibrium, and ϵ_n is a zero-mean noise with a standard deviation of σ. Assuming that the recovery rate λ and the time interval Δt are independent of y_n, this setup can be effectively described as a first-order autoregressive (AR(1)) process:

$$y_{n+1} = \alpha y_n + \sigma \epsilon_n \tag{3.9}$$

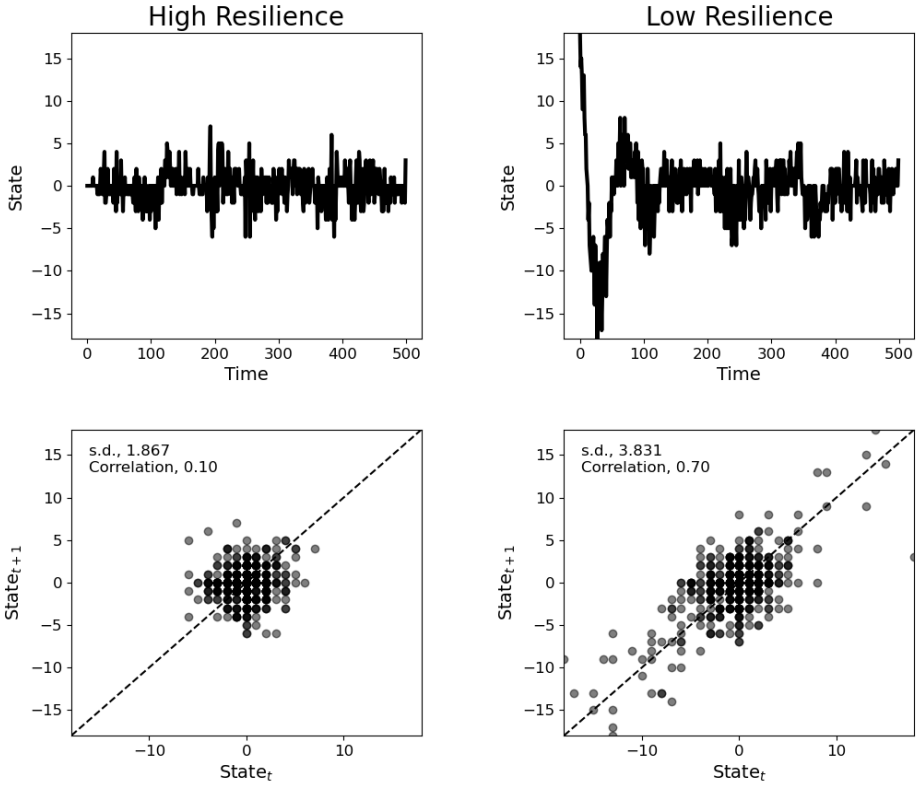

Figure 3.5: The effect CSD measured in stochastically induced fluctuations in the state of the system as increased variance and "memory" as shown by lag-1 auto-correlation.

In an AR(1) process, the AR1 coefficient, denoted as $\alpha \equiv e^{\lambda \Delta t}$, quantifies how much each value in the series depend on its immediate predecessor. For white noise, this coefficient is zero, whereas for red (auto-correlated) noise, the coefficient is close to one [39]. The expectation of an AR(1) process $y_{n+1} = c + \alpha y_n + \sigma \epsilon_n$ is [40]:

$$E(y_{n+1}) = c + \alpha E(y_n) + E(\sigma \epsilon_n) \Rightarrow \mu = c + \alpha \mu + 0 \Rightarrow \mu = \frac{c}{1-\alpha} \tag{3.10}$$

can be understood as a linear combination of the previous value, y_n, adjusted by the coefficients c and α, with an added term ϵ_n for noise scaled by σ [40]. For $c = 0$, the mean equals zero and the variance is found to be :

$$Var(y_{n+1}) = E(y_n^2) - \mu^2 = \frac{\sigma^2}{1-\alpha} \tag{3.11}$$

Close to the critical point, the return recovery rate to equilibrium decreases, implying that λ approaches zero and the auto-correlation α tends to one. Thus the variance tends to infinity.

While CSD has traditionally been applied to ecological systems, van der Pluijm [13] attempted to explore the application of CSD in predicting critical transitions in quadrotors. He investigated whether EWS could be detected in the time-series data of a damaged quadrotor with a single rotor failure (SRF) using CSD indicators. The objective was to gather data on the steady-state and transient behavior of a damaged quadrotor experiencing various wind speeds, and to examine if there were EWS that could be detected prior to the quadrotor crashing as the wind speed was increased.

The analysis revealed that w_2, and q were the most relevant variables, as angles were found to be less likely to contain EWS, compared to angular rates as stochastic effects are less pronounced in angles. The

study's conclusions yielded that using CSD theory alone to identify EWS in quad-rotor systems was not feasible due to the unrealistic constraints it placed on wind speed behavior. As noted by the author, CSD requires a slow and monotonic transitions, whereas damaged quadrotors usually are subject to sudden changes, resembling rapid regime shifts. Observations from Figure 3.3 indicate that some rapid regime shifts show no signs of slowing down, impacting the effectiveness of CSD based on the rate of wind speed changes. However, by expanding the scope beyond traditional CSD indicators and incorporating other domain specific indicators such as actuator saturation metric, more reliable EWS could be generated. Additionally, a sensitivity analysis demonstrated that the most critical parameter is the size of the rolling window used, with larger windows yielding fewer false positives; however, for robotic applications, smaller windows are preferred.

This insight is consistent with a study by Wen *et al.* [41], which emphasizes the sensitivity of EWS detection to several factors, including the intrinsic early warning period (intrinsic time-scale of LoC event, and window size), data frequency, and the choice of test statistic.

Another insightful example comes from a study by Dakos *et al.* [42], which delved into the potential EWS in a highly stochastic system using a model that describes the shift of a lake from an oligotrophic to a eutrophic state [43]. Initially, it was assumed that in such noisy conditions, CSD might not be relevant, and traditional EWS would not be expected to be effective. However, the research uncovered an unexpected connection. Even though this system did not align with the typical characteristics of a slow-time-varying system, but rather being characterized by strong perturbations that can lead to rapid regime shifts, or "flickering". The study revealed that the behavior of the system often exhibited rising variance, AR1, and skew. These patterns closely resembled the effects of CSD, challenging traditional assumptions and highlighting the adaptability of CSD Theory in unforeseen, highly stochastic environments. This unexpected finding underscores the versatility of CSD theory in identifying critical transitions, even in systems with dynamic, non-monotonic perturbations.

3.2.2. Complexity Measures

Measuring complexity within time-series data is critical in understanding, predicting and controlling the behavior of a dynamic system. It offers means to quantify the level or disorder, emergence of patterns, and inherent instability that may occur as a system evolves in time. Various complexity measures have been developed to compare time-series, to distinguish their regular, chaotic, and random behavior. The main types of complexity parameters are entropies, fractal dimensions, and Lyapunov Exponents [44]. However, for the scope of this research, Lyapunov Exponents will be omitted as they are generally considered a model-based method.

Entropy-based Parameters

Entropy is a powerful tool for measuring the degree of disorder or irregularities in a system [45]. If the degree of randomness is low, the system tends to become more organized, whereas high disorder in the data will give higher entropy values. Various entropy measures are available for time-series prediction, including, but not limited to, Shannon Entropy and Permutation Entropy.

Consider a time series $A = \{a(i) : i = 1, 2, ..., N\}$, the Shannon Entropy (SE) is defined as:

$$SE = \sum_{i=1}^{n} p(a_i) \log_2 \frac{1}{p(a_i)} = -\sum_{i=1}^{n} p(a_i) \log_2 p(a_i) \tag{3.12}$$

where $p(a_i)$ is the probability of a specific outcome or value a_i occurring. The SE of variable A is a measure of the expected randomness obtained through the measurement of the values in variable A. A higher entropy is obtained by more uncertainty in the time series [46]. Applied to time series, SE assesses predictability, measuring the information needed to anticipate the next value in the sequence. SE assumes that the data points in the time series are independent, identically distributed, and although it can be applied to non-stationary time series, it may not capture changes in the underlying probability distribution. Therefore, assumptions of stationarity can make it more appropriate [44].

Permutation entropy introduced by Bandt and Pompe [44], which combines the ideas from SE and ordinal pattern analysis, through the estimation of related frequencies of ordinal patterns obtained from the time series data [47]. There are two parameters which must be defined to calculate the PE : the embedding

dimension (m) which determines how many data points are considered at each time step, and the time delay (τ) indicating the time gap between these data points. The following is a description of how to calculate PE as provided by Bandt and Pompe:

For a time series, the delay embedding vector is contructed at each time point i. It is a m dimensional vector that includes data points from the time series at specific time indices, determined by the embedding dimension and time delay, it can be defiend as:

$$A_i^m = [a(i), a(i+\tau), ..., a(i+(m-2)\tau), a(i+(m-1)\tau)] \tag{3.13}$$

The vector A_i^m represents a segment of the time series at time i. Now each delay embedding vector A_i^m can have a permutation π. A permutation is an arrangement of intergers $\pi = (r_0, r_1, ..., r_{m-1})$ that reflects the order of data points in A_i^m. They key condition is that the data points in A_i^m must be ordered from the smallest to largest according to the permutation:

$$\begin{aligned} &a(t+r_0\tau) \leq a(t+r_1\tau) \leq \cdots \leq a(t+r_{m-1}\tau) \\ &\text{where} : 0 \leq r_i \leq (m-1) \text{ and } r_i \neq r_j \end{aligned} \tag{3.14}$$

For each permutation π you can calculate its relative frequency, denoted by $p(\pi)$. This frequency represents how often that specific ordinal pattern appears within the time series. The numerator counts how many times A_i^m has a specific permutation of π in the time series, while the denominator normalises the count.

$$p(\pi) = \frac{\#\{t | t \leq N-(m-1)\tau, A_i^m \text{ has type } \pi\}}{N-m+1} \tag{3.15}$$

Now, the PE for a given embedding dimension m is calculated by summing over the relative frequencies of all possible permutations π and applying a logarithimic function. It quantifies the complexity or entropy of ordinal patterns within the time series:

$$PE = -\sum p(\pi) \log_2 p(\pi) \tag{3.16}$$

Now, the maximum possible value for PE for a specific embedding dimension is $\log(m!)$, where all possible permutations have equal probabilities. To make the measure independent of m the Normalised Permutation Entropy (NPE) is defined by dividing the PE by $\ln(m!)$. NPE provides a standardised measure that facilitates comparison across different time series with varying embedding dimensions.

Note that the equations for SE (Equation 3.12) and PE (Equation 3.16) look identical. However the difference between the two lies in how $p(a_i)$ or $p(\pi)$ is calculated. For SE, $p(a_i)$ represents the probability of a specific value occurring, based on the probability distribution of data values obtained from the time series itself. In contrast, for PE, $p(\pi)$ represents the frequency of a specific ordinal pattern appearing within the time series.

Fractal Dimensions

Fractals are mathematical and geometrical objects that exhibit self-similarity at different scales. This means that as you zoom in on a fractal object, you see similar patterns or structures at different levels of magnification [48]. In the context of time series data, the idea is to explore whether the data exhibits self-similarity, or fractal-like behaviour, at different time scales. This can be achieved by considering the time series at different levels of temporal resolution or by using various techniques.

Now the Hurst Exponent (H) is a measure related to the fractal dimension (D) of the time series as $H = 2 - D$ [49]. The Hurst exponent quantifies the degree of self-similarity or long-range dependence in a time series, and the scaling of the standard deviation is used to estimate the Hurst exponent.

Calculating the standard deviation of a time series within different time intervals or windows of varying lengths, essentially measures how the variability or fluctuations in the data change as the time scale changes.

Now if the said time-series is self-similar, it will exhibit a scaling behaviour, meaning that the standard deviation should change systematically with the length of the time interval over which it is calculated. The standard deviation typically scales with time scale T as a power-law relationship :

$$\sigma(T) = K \cdot T^H \tag{3.17}$$

Where K is a constant, and H is the Hurst Exponent, the exponent in the power-law relationship that describes how the standard deviation scales with the time scale. The calculation of H can be performed following the Multifractal Detrended Fluctuation Analysis (MFDFA) [50] [51].

H represents a measure of how persistent or correlated a signal is. If an increase in the signal's value is more likely to be followed by another increase, we call the signal persistent. In such cases, H exceeds 0.5. Conversely, an anti-persistent or negatively correlated signal, where an increase in value is mostly followed by a decrease or vice versa, corresponds to H values less than 0.5. Uncorrelated white noise, where there's no systematic correlation, results in an H of 0.5.

Fractal analysis has a wide range of practical applications across various fields, including life sciences, engineering, econophysics, and geophysics [52][53][54][55][56][57].

3.2.3. Machine Learning Methods

Over the past two decades, the field machine learning has witnessed substantial growth in various domains, evolving from an experimental concept in research labs to a versatile technology widely adopted for commercial purposes. In the realm of artificial intelligence (AI), machine learning (ML) has become the preferred approach for creating functional software solutions across various domains, including computer vision, speech recognition, natural language processing, robot control, and numerous other applications. In addition to machine learning, the past decade has also witnessed a substantial growth in the ability of networked and mobile computing systems to gather and transport vast amounts of data, a phenomenon often referred to as "Big Data". This surge in extensive data across all areas of human endeavour has triggered a wave of new demands on the underlying machine-learning algorithms [58]. Machine Learning are quintessentially data-driven, residing within the domain of black-box algorithms. Here, the system's strength lies in its ability to autonomously decipher patterns and relationships within the data, without explicit dependence on established theories or principles. In this method, we feed data to the machine learning algorithm, and it learns from the data itself, progressively improving its predictive capabilities.

The most popular type of machine learning methods are supervised learning methods [59]. In supervised learning, prediction systems typically create their outputs through a learned mapping $f(x)$. This mapping is responsible for generating an output y for each input x, or alternatively, a probability distribution over y based on x. Various types of mappings, such as decision trees, decision forests, logistic regression, support vector machines, neural networks, kernel machines, and Bayesian classifiers, have been explored and employed [59].

Within the context of unmanned aerial systems, machine learning methods have opened up new possibilities in predicting critical transitions and Loss of Control (LoC) events in quadrotors. There are several machine learning methodologies that are well suited for time-series predictions, each offering a unique perspective on LoC prediction:

- **Recurrent Models :** Echo State, Long Short-Term Memory, Gated Recurrent Units
- **Sequence Modelling Models :** Transformer and Attention-Based Models
- **Probabilistic and Uncertainty-Aware Models :** Bayesian Neural Networks
- **Non-Traditional Approaches :** Fuzzy Logic Inference System

Recurrent Models

A recurrent neural network is a type of a neural network specifically designed for handling time series or sequential data. Unlike a standard neural network, RNNs can retain information over time, influencing the current and future states using past input data. This is achieved through a feedback loop, where the RNN incorporates the output from the previous input into its calculations before passing through the activation function. This feedback loop can be thought of as a series of interconnected single-input-single-output (SISO) neural networks, which can be visualised in Figure 3.6.

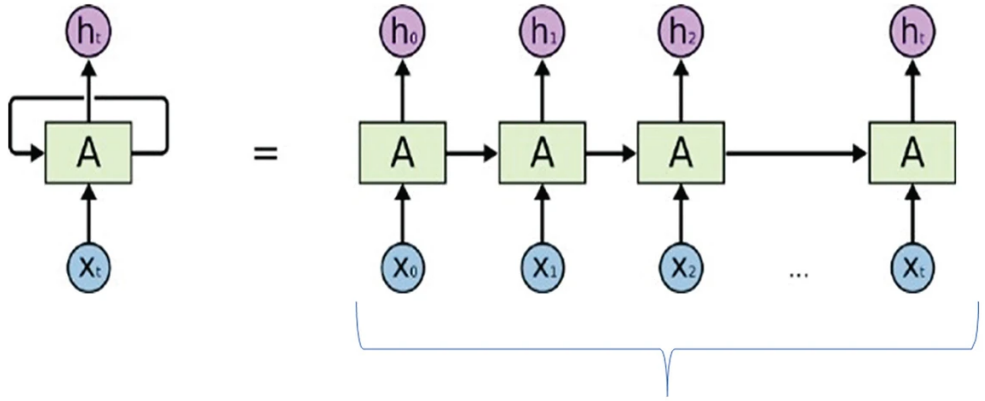

Figure 3.6: Unrolled Recurrent Neural Network [60]

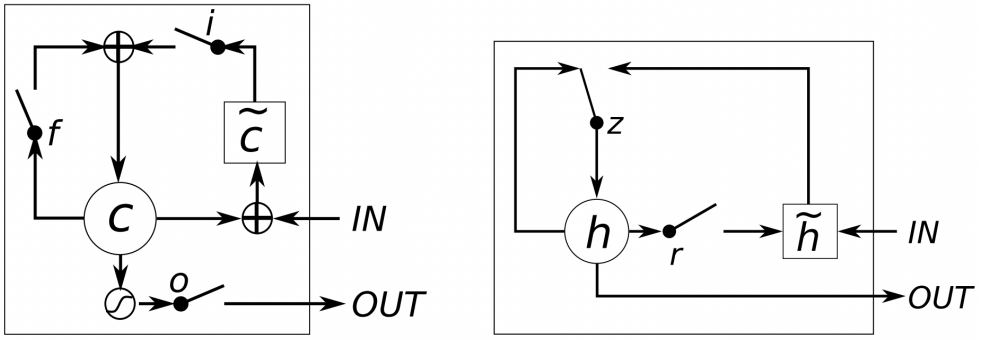

Figure 3.7: Illustration of LSTM(Left) and GRU(Right). Left : i, f and o are the input, forget, and output gates, respectively. c and $\tilde{c}$ denote the memory cell and the new memory cell content. Right : r and z are the reset and update gates, and h and $\tilde{h}$ are the activation and the candidate activation [62].

One of the essential feature of an RNN is that they share weights and biases across all sequential inputs, ensuring that the number of parameters remains constant, regardless of the length of the input sequence. Although this brings some benefits such as computational efficiency, it also comes at a cost of the vanishing/exploding gradient problem. This problem arises when a weight that multiplies the output fed back into the network is larger than 1. As the sequence length increases, the gradient becomes disproportionately large, resulting in slow convergence towards the minimum of the loss function [60]. To overcome these challenges, alternative RNN architectures have been developed, such as Long Short-Term Memory (LSTM), Gated Recurrent Unit, and Echo State Networks.

LSTM is an extension to the standard RNN which allows the RNNs to remember their input over a longer period of time. It achieves this by incorporating memory cells that function in a similar manner to a computer's memory. The LSTM cell is capable of reading, writing and erasing information through its three gates: input, forget, and output gate [61]. GRUs were proposed by Cho *et al.* [63] in order to allow each recurrent unit to flexibly capture dependencies at various time scales. Both LSTM and GRU units share a key feature : the additive update from one time step to another, which the traditional RNNs lack. In RNNs, the activation is always replaced with a new value based on the current input and previous hidden states. However, LSTM and GRU retain the existing content and add the new content on top of it, preserving valuable features over long sequences, and creating shortcut paths that prevent the vanishing/exploding gradient problem. However, there are two main differences. The LSTM unit controls the exposure of the memory content through the output gate, whereas the GRU exposes its full content without control. Secondly, the location of the input gate differs between the two. LSTM computes new memory content independently from the amount of information carried from the previous step, whereas the GRU controls information flow from the previous activation but does not independently regulate the amount of candidate activation added. This essentially means that LSTM has two separate mechanisms (input and output gates) that controls how much old and new information is included in the memory cell,

whereas GRUs only have a single mechanism that simultaneously manages the flow of information from the previous state and the addition of new information [62].

Finally, Echo State Networks, another subset of RNNs that isn't trained using the traditional gradient-descent based methods, instead uses the reservoir computing framework. Reservoir computing is when the input signal is connected to a fixed (non-trainable) and random dynamical system (the reservoir/neural network), thus creating a higher dimension representation (embedding). This embedding is then connected to the desired output via trainable units [64]. One recommendation is that the number of nodes in the reservoir N should be in the range $\frac{T}{10} \leq N \leq \frac{T}{2}$, where T is the number of sample data [65]. The optimal value of N depends on the periodicity of the training data and the complexity of the learning task. A reservoir size that is too small may result in model inaccuracy, whereas a reservoir size that is too large may lead to slow training and data over-fitting [66]. Due to the fixed random weights, and the reservoir computing aspects of the ESN, this helps prevent the vanishing/exploding gradient problem [67].

Attention Based Methods

The Transformer is a machine learning architecture introduced in the paper "Attention is All You Need" by Vaswani *et al.* [68] in 2017. It has since become a fundamental building block for a wide range of natural language processing (NLP) tasks and beyond. The key innovation of the Transformer architecture is the concept of "self-attention."

Attention is a mechanism which allows a model to weigh the importance of different segments of an input data when making predictions. Self-attention allows the model to consider the entire input sequence simultaneously, instead of dealing with sequential data 'sequentially', assigning different attention weights to various segments of the sequence. This allows for the model to focus on the more relevant context elements, leading to a more informed and context-aware predictions.

The main advantages of a Transformer model is its parallelization, dealing with long-term dependencies, and bi-directionality. Traditional models that deal with sequential data such as RNNs process data sequentially, which can slow. Transformers can parallelize the computation due to the self-attention. Transformers are also able to capture long-term dependencies, and also analyse data bidirectionally meaning they can consider the elements both before and after a given position. These traits allows the transformer to excel in handling sequential data and capturing complex, long-term trends by efficiently modelling relationships between elements in a sequence, making it a powerful tool for wide range of ML applications [68].

Probabilistic Models

Bayesian Neural Networks (BNNs) are a variant of the traditional neural networks that incorporate the principles of stochastic modelling. They differ from a traditional neural network in how they handle uncertainty and make predictions. In a Bayesian neural network, model parameters such as weights and biases are treated as a probability distribution, which allows them to have a distribution of possible parameter values, typically represented as a posterior distribution. Once the posterior distribution is estimated, BNNs can sample from it to obtain multiple sets of parameter values, each representing a different hypothesis about the models structure, and these samples capture the uncertainty in the parameter values, reflecting the models uncertainty about its own architecture. In addition, when making predictions with BNNs, instead of a single point estimate, the BNN outputs a distribution of possible outcomes. This distribution is obtained by running inference using each set of sampled parameters. The spread of variance of these predictions represents the model's uncertainty. For regression, this can be a predictive distribution, while for classification, it can be a probability distribution over classes [69].

Fuzzy Logic

Fuzzy Logic Inference System (FLIS) is the process of formulating the mapping from given input(s) to output(s) using fuzzy logic. Fuzzy Logic is based on the idea that the truth can be expressed as a continuous *fuzzy* variable , not a binary output [70]. This approach may be particularly useful in scenarios like predicting the onset of Loss of Control (LoC), where a quadrotor's state might not strictly be binary; it could be experiencing varying degrees of LoC.

FLIS is instrumental across various domains, including automatic control, data classification, decision-making, expert systems, robotics, and more, due to its capacity to handle nuances in data and decision-making processes [71]. The system comprises five primary components:

- **Rule Base** - A collection of fuzzy rules that use logical operators such as IF, OR, and NOT.
- **Database** - Defines membership functions for input and output variables.
- **Fuzzification Interface** - Converts crisp inputs into fuzzy inputs through degrees of match with linguistic variables (membership functions).
- **Decision-Making Unit** - Executes inference on the fuzzy rules.
- **Defuzzification Interface** - Transforms fuzzy inference results back into crisp outputs.

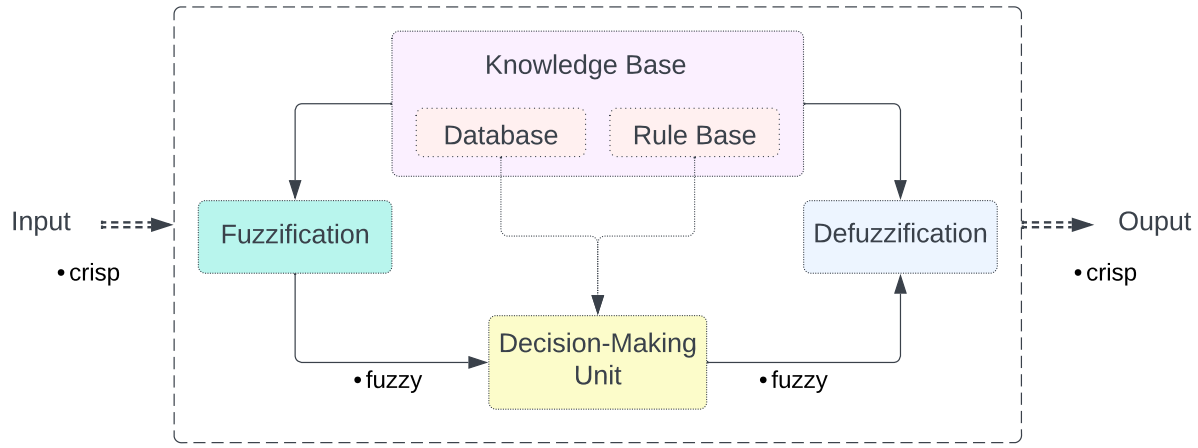

Figure 3.8: Structure of a Fuzzy Logic Inference System (FLIS), with the five main components : Rule Base, Database, Fuzzification Interface, Decision-Making Unit, and Defuzzification Interface

The process begins with fuzzification, transforming specific inputs into degrees of match with membership functions, based on predefined intervals. This transformation results in fuzzified variables, representing the antecedents in the fuzzy logic system. These fuzzified variables are then processed by the decision-making unit. This unit employs logical operators ('AND' for the minimum, 'OR' for the maximum, and 'NOT' for the complement of the membership value) to associate the antecedents' membership functions with a consequent membership function in the output. The choice and application of these logical operators critically influence the degree of match between the antecedents and the consequent, effectively determining the output's membership value. Upon applying all the rules, where each rule connects antecedents to a consequent, the decision-making unit aggregates these consequent values to produce a single, crisp output through defuzzification. This final step typically involves calculating the centroid of the aggregated fuzzy set, culminating in a crisp output value reflective of the fuzzy logic inference.

Fuzzy logic inference systems has various advantages, such as handling uncertainty, inherent interpretability, and most importantly domain expertise. If domain expert knowledge regarding quadrotor LoC are available, this can be leveraged to define appropriate membership functions and fuzzy rules that can reflect the intricacies of the system, such as combining CSD, and actuator phase delay parameters.

Part III

Closure

4

Conclusion

4.1. Answer to Research Questions

The research questions posed in Chapter 1 are repeated below for convenience.

Main Research Question

What early warning signals (EWS) act as precursors for the onset of loss-of-control (LoC) in quadrotors approaching a critical transition, if any?

To answer the main research question, it was found that CSD theory applied to onboard measurements provided from the Inertial Measurement Unit (IMU) from the quadrotor can act as precursors for the onset of loss-of-control. More specifically, calculating the lag-1-autocorrelation coefficients (AR1) and the Inter-quartile Range (IQR) on the detrended signals of gyro rates: p, q, r, and measured rotor speeds (debug): $\omega_0, \omega_1, \omega_2, \omega_3$ using a rolling window analysis. In addition to CSD, the actuator phase to act as a suitable precursor to the onset of LoC, which was the time-delay variable, which was a measure of the actuator phase delay. However this variable seemed to only have an effect towards larger quadrotors with larger actuators. This will be further discussed in answering RQ2.

Sub-Research Questions

RQ1 : How can LoC in quadrotors be accurately detected and defined using a data-driven method, considering data sources, sensor inputs, and data analysis techniques?

- What data sources and sensors can effectively capture relevant information about an upcoming LoC event?
- What data pre-processing techniques are required to extract informative features for LoC detection?
- Can sliding window analysis and the data pre-processing methods applied to the sensor outputs be performed in real-time and onboard the drone for timely LoC detection?

In addressing the first sub-research question, we discovered that the most effective data comes from gyro rates, which are measured by the gyroscopic sensor. In addition, the measured rotor speeds recorded by the Electronic Speed Controllers (ESC) through the use of Bi-directional DShot600, were also analyzed using CSD, however although they can detect a LoC event accurately, they were not suitable as an EWS indicator, as they did not appear prior to the LoC, and their increase in AR1 was too abrupt.

The process of data-processing involved several key steps. Initially, it was necessary to convert the data from the onboard black-box logs into an accessible CSV format. Following this, the most crucial step in data preparation was the detrending of the signals. Our findings indicate that for CSD analysis to effectively reveal meaningful insights, its essential to remove long-term trends from the Data. This step isolates the signal's residual fluctuations, in order to measure the rate of recovery to minor perturbations. Without this, any remaining trends or periodic patterns in the time-series could lead to false indications.To obtain meaningful results from CSD analysis on quadrotors, it is imperative that the long term trends in

the data are removed (detrending) to isolate the fluctuations in the residuals. If this is not done properly, the remaining correlations (trends or periodicity in the time series) can lead to false indications. Detrending is done by finding a function to approximate the signal's long-term trends. Once this can be captured, these long-term trends may be removed from the original signal. The result is the detrended signal, which only contains the residual fluctuations to external disturbances. During the preliminary phase of the research, four detrending techniques were evaluated: Moving Average, Differencing, Kernel Regression, High Pass Filter. The investigation revealed that the gyro rate data yielded the most effective EWS through CSD when detrended using Kernel Regression. Conversely, the measured rotor speed is best detrended using a High-Pass Filter.

Currently, real-time and onboard calculation of LoC detection is not feasible. However, potential solutions exist to make this achievable. The primary challenge lies in the detrending process, particularly in the HPF implementation. The HPF is designed using the `filtfilt` function from SciPy's Signal library in Python. This function utilizes filter coefficients obtained from the Butterworth filter and employs a forward-backward filtering technique. This technique filters the data both forward and backward, ensuring zero phase distortion. However, the backward filtering makes this method non-causal, limiting its application to offline use once all data has been collected. An alternative approach is to use the `lfilter` function from SciPy's Signal library, which performs single forward pass filtering. While this maintains causality, enabling online application, it introduces phase distortion. Therefore future work should explore the feasibility of detecting EWS using `lfilter` for the IQR values.

Sub-Research Questions

RQ2 : How can different EWS indicators, when used collectively, contribute to more accurate detection and prediction of the onset of LoC in quadrotors, while effectively mitigating erroneous alarms (Type I errors) and missed warnings (Type II errors)?

- What role do Critical Slowing Down (CSD) indicators play in predicting the onset of a critical transition in quadrotors?
- How do actuator saturation levels impact the onset of a critical transition in quadrotors?
- In what ways can the integration of multiple parameters improve the robustness of LoC prediction, ensuring a lower rate of erroneous alarms (Type I errors) and missed warnings (Type II errors)?
- How can the model's predictive performance be evaluated, and what are the most relevant evaluation metrics?

In addressing the second sub-research question, CSD has shown its pivotal role in detection of critical transitions in quadrotors. It operates on the principle that as a system nears a critical point, its recovery from disturbances slows down. This slowing down is a result of the dominant eigenvalue approaching zero, which means the system takes longer to return to equilibrium after a perturbation. This characteristic of CSD provides an effective early warning signal, with specific indicators such as increased lag-1 auto-correlation and a rise in variance of system fluctuations signaling an imminent critical transition. Essentially, CSD increases the 'memory' of the system, making its current state more similar to its past states, which can be observed through patterns in lag-1 auto-correlation and variance.

The primary causes of LoC in quadrotors is actuator saturation [72], which causes non-linear actuator responses, phase lags, and instability. When saturated, control authority deteriorates, and without control authority, the ability to counteract external disturbances is lost. Recognizing this, the detection of such lag is considered for use as an additional indicator of LoC. From the results it was found that if a phase lag between the commanded and actual rotor speed can be detected, the time delay variable τ acts as an effective EWS for LoC maneuvers. In addition, the integrated version of time delay τ, the delay duration t_τ acts better as an indicator of LoC, as nominal flights that contains fast throttle changes also do show instaneous occurrences of τ, that are not related to approaching LoC. However, the results show that no (measurable) phase lag in smaller drones (DataCan, GimbalDrone), pointing to a correlation between actuator size and τ detection. This seems plausible, since larger actuators, with their mass distributed away from the rotation axis, have a higher moment of inertia and thus exhibit a slower response times. In contrast, smaller actuators exhibit lower inertia and quicker response times. Considering this, increasing the sampling rate for smaller drones could offer a finer resolution for detecting shorter lags, given their rapid response capabilities. Ideally, CSD would serve as an EWS indicator and LoC detector, with $t_\tau$acting

as a redundancy layer to minimize false positives, and increase detection rate. Future research should delve deeper into the time delay variable, exploring how variations in sampling rate or different controller designs might influence lag detection. Additionally, testing Roll LoC maneuvers on larger drones with gimbal setups could substantiate the assumption that τ, detection correlates with actuator size and not LoC mode, ensuring that the results are applicable across various drone configurations.

To integrate multiple indicators, a Fuzzy Logic Inference System (FLIS), combining CSD indicators, τ, and stick inputs. This integration improved LoC detection rates and significantly reduced the occurrence of false positives (Type I errors). Specifically, for Yaw and Roll induced LoC scenarios, the detection rates were impressively high (96% and 98%, respectively). However, relying solely on the CSD parameter AR1 led to false positives due to its sensitivity to the detrending method used. By incorporating a stick-neutrality condition, numerous variables, and indicators, the CSD outputs could be inferred in an effective way, the final false positive rate at 8% and 9% for yaw and roll induced LoC, respectively.

Evaluating the model's predictive performance, the most relevant metrics were the Detection Rate, False Positive Rate, and Time to Detection. These metrics provide a comprehensive understanding of the model's effectiveness in predicting and detecting LoC across different quadrotors and different LoC modes.

Sub-Research Questions

RQ3: To what extent can these LoC events be detected through parameters such as CSD and actuator phase lag?

- How do aggressive yaw, and roll maneuvers influence these parameters and contribute to LoC risk?
- Do different maneuvers exhibit consistent global trends in CSD, and actuator saturation that can be used for early LoC detection and prediction?
- How are these indicators affected by nominal flight?

In addressing the third sub-research question, this research investigated the nature of Roll induced LoC and Yaw induced LoC.

For yaw-induced LoC, perturbations are introduced in roll and pitch oscillations stemming from the "anti-gravity" feature of BetaFlight (the flight controller), which induces oscillations due to a misalignment between the c.g and the IMU. The high yaw rates results in translational accelerometer measurements due to the c.g and IMU misalignment. Consequently, the flight controller misinterprets these readings as disturbances in roll and pitch angles, endeavoring to counteract them, which induces oscillations around the roll and pitch axes. While manageable at low rotational rates or when actuators are not saturated, these oscillations intensify and become unstable under saturation, ultimately leading to controller-induced LoC. Conversely, roll-induced LoC introduces perturbations in yaw through the coupling of roll and yaw axis due to gyroscopic precession. This phenomenon, observed in rotating bodies, redirects angular momentum around the z-axis, causing yawing motion. As yawing intensifies, it also triggers pitch oscillations, culminating in LoC when BetaFlight deactivates the I term to avoid integrator windup, halting the roll motion. Although roll-induced LoC also ultimately occurs due to the controller killing the I term, the reasons differ: for Yaw-LoC, the anti-gravity feature induces error accumulation due to corrective measures, whereas for Roll-LoC, the error accumulates due to the coupling of roll and yaw. Integrating findings from both types of LoC maneuvers underscores a key takeaway: despite differing dynamics, CSD analysis consistently identifies trends applicable across various LoC scenarios. This highlights the efficacy and broad applicability of CSD as a method for detecting potential LoC events, irrespective of their specific causes or maneuvers involved.

As CSD focuses on identifying decreased resilience within the signal's residuals, the detrending process should primarily reveal the system's response to external disturbances. However, the detrending methods would occasionally retain gyro oscillations from intended maneuvers. To address this, a stick-neutrality condition was imposed, allowing Gyro AR1 inference only when no stick inputs occurred in the past 450 time steps. This duration of 450 time-steps is due to the nature of a rolling window analysis for generating AR1 values. As the AR1 is calculated using a window of the previous 450 time-steps, the impact of a stick input may persist within the AR1 calculation window. While Gyro AR1 sufficed for detecting LoC in uncoupled flights, coupled maneuvers like those in Yaw-LoC SysID flights necessitated additional measures

due to the stick neutrality condition. Therefore, CSD-τ rules were used as an auxiliary measure to enhance LoC detection rates and address the limitations of the stick neutrality condition when using Gyro-AR1 rules. Since the GYRO-AR1 rules are contingent on the absence of stick inputs, the CSD-τ rules provide a supplementary indication of stability when GYRO-AR1 rules are inapplicable, such as during maneuvers that involve stick inputs in all three axes. These rules utilize Gyro IQR, measured rotor speed's AR1, and the time delay duration variable t_τ to assess the quadrotor's risk of LoC.

5

Recommendations

This chapter provides a brief overview of the primary recommendations for the future continuation of this research project.

Recommendation 1 - Detrending using stick-neutrality condition
The primary limitation of CSD lies in its sensitivity to the detrending method used. Ideally, detrending should isolate oscillations stemming solely from the quadrotor's response to external perturbations. However the detrending method cannot selectively filter out intended oscillations, occasionally retaining some oscillations caused by intended maneuvers.This leads to false positive rates, notably pronounced in Yaw LoC scenarios, which required corrective maneuvers prior to the LoC maneuver due to free flight conditions. To mitigate this, a stick-neutrality condition was introduced, allowing Gyro AR1 inference only in the absence of stick inputs over the past 450 time steps. While this reduced false positives, it remains limited, especially in flights with coupled maneuvers across all axes, rendering Gyro-AR1 outputs obsolete. Therefore future work should explore methods to integrate the stick condition into detrending, dynamically adjusting the bandwidth/cutoff frequency based on stick inputs. This adaptive approach could enhance filtering during stick inputs, and reduce it during their absence. Additionally, FLIS could also assign weights to the Gyro AR1 outputs, adjusting their significance based on the magnitude of stick input for a given axis.

Recommendation 2 - Furhter investigation of the Time Delay Indicator
The main limitation of the Time Delay indicator was its generalizability. From the analysis, it was revealed that no (measurable) phase lag was detected in smaller drones (DataCan, GimbalDrone), pointing to a correlation between actuator size and τ detection. This seems plausible, since larger actuators, with their mass distributed away from the rotation axis, have a higher moment of inertia and thus exhibit a slower response times. In contrast, smaller actuators exhibit lower inertia and quicker response times. Therefore, future work should examine how variations in sampling rate or different controller designs might influence lag detection. Additionally, testing Roll LoC maneuvers on larger drones with gimbal setups could substantiate the assumption that τ, detection correlates with actuator size and not LoC mode, ensuring that the results are applicable across various drone configurations.

Recommendation 3 - Causal Filtering for HPF
Currently, real-time and onboard calculation of LoC detection is not feasible. However, potential solutions exist to make this achievable. The primary challenge lies in the detrending process, particularly in the HPF implementation. The HPF is designed using the `filtfilt` function from SciPy's Signal library in Python. This function utilizes filter coefficients obtained from the Butterworth filter and employs a forward-backward filtering technique. This technique filters the data both forward and backward, ensuring zero phase distortion. However, the backward filtering makes this method non-causal, limiting its application to offline use once all data has been collected. An alternative approach is to use the `lfilter` function from SciPy's Signal library, which performs single forward pass filtering. While this maintains causality, enabling online application, it introduces phase distortion. Therefore future work should explore the feasibility of detecting EWS using `lfilter` for the IQR values.

References

[1] Robert Mahony et al. "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor". en. In: *IEEE Robotics & Automation Magazine* 19.3 (Sept. 2012), pp. 20–32. DOI: 10.1109/MRA.2012.2206474. URL: https://ieeexplore.ieee.org/document/6289431/ (visited on 10/25/2023).

[2] *Commercial Drone Market Size, Share & Trends Analysis Report By Product, By Application, By End-use, By Propulsion Type, By Range, By Operating Mode, By Endurance, By Region, And Segment Forecasts, 2023 - 2030*. Technical Report 978-1-68038-482-6. Grand View Research, Inc.

[3] Matias Grez. *Drone crashes onto piste, misses champion skier by inches*. en. Dec. 2015. URL: https://www.cnn.com/2015/12/23/sport/marcel-hirscher-drone-crash/index.html (visited on 09/18/2023).

[4] Graham Wild et al. "Exploring Civil Drone Accidents and Incidents to Help Prevent Potential Air Disasters". en. In: *Aerospace* 3.3 (July 2016), p. 22. DOI: 10.3390/aerospace3030022. URL: http://www.mdpi.com/2226-4310/3/3/22 (visited on 09/18/2023).

[5] Ye Zhang et al. "Database Building and Interpolation for a Safe Flight Envelope Prediction System". en. In: *2018 AIAA Information Systems-AIAA Infotech @ Aerospace*. Kissimmee, Florida: American Institute of Aeronautics and Astronautics, Jan. 2018. DOI: 10.2514/6.2018-1635. URL: https://arc.aiaa.org/doi/10.2514/6.2018-1635 (visited on 10/16/2023).

[6] Ye Zhang et al. "Database-Driven Safe Flight-Envelope Protection for Impaired Aircraft". In: *Journal of Aerospace Information Systems* 18.1 (Jan. 2021). Publisher: American Institute of Aeronautics and Astronautics, pp. 14–25. DOI: 10.2514/1.I010846. URL: https://arc.aiaa.org/doi/10.2514/1.I010846 (visited on 09/19/2023).

[7] E. R. van Oort et al. "Maneuver Envelope Determination through Reachability Analysis". In: *Advances in Aerospace Guidance, Navigation and Control*. Ed. by Florian Holzapfel et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 91–102.

[8] Thomas Lombaerts et al. "Safe Maneuvering Envelope Estimation based on a Physical Approach". en. In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. Boston, MA: American Institute of Aeronautics and Astronautics, Aug. 2013. DOI: 10.2514/6.2013-4618. URL: https://arc.aiaa.org/doi/10.2514/6.2013-4618 (visited on 10/16/2023).

[9] June Chongvisal et al. "Loss-of-Control Prediction and Prevention for NASA's Transport Class Model". In: *AIAA Guidance, Navigation, and Control Conference*. AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, Jan. 2014. DOI: 10.2514/6.2014-0784. URL: https://arc.aiaa.org/doi/10.2514/6.2014-0784 (visited on 09/19/2023).

[10] Zhidong Lu et al. "Flight Envelope Prediction via Optimal Control-Based Reachability Analysis". In: *Journal of Guidance, Control, and Dynamics* 45.1 (2022). Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.G006219, pp. 185–195. DOI: 10.2514/1.G006219. URL: https://doi.org/10.2514/1.G006219 (visited on 04/24/2024).

[11] Sihao Sun et al. "Quadrotor Safe Flight Envelope Prediction in the High-Speed Regime: A Monte-Carlo Approach". en. In: *AIAA Scitech 2019 Forum*. San Diego, California: American Institute of Aeronautics and Astronautics, Jan. 2019. DOI: 10.2514/6.2019-0948. URL: https://arc.aiaa.org/doi/10.2514/6.2019-0948 (visited on 09/19/2023).

[12] Lauren Kaffa. "Safe Flight Envelope Prediction and Experimental Validation of Multirotor UAV: An Analysis on the Effects of Longitudinal Center of Gravity Position and Actuator Dynamics on Quadcopter Flight Envelopes". en. MA thesis. Delft: Delft University of Technology, July 2023. URL: https://repository.tudelft.nl/islandora/object/uuid%3A5d3f5432-3801-4b5b-9dcf-6db1ce2cb150 (visited on 11/04/2023).

[13] Anthony van der Pluijm. "Early Warning Signals for Loss of Control Prediction of a Damaged Quadcopter". en. MA thesis. Delft: Delft University of Technology, Jan. 2020.

[14] T Hoppenbrouwer. "Loss of Control Prediction for Quadrotors with Single Rotor Failure". en. MA thesis. Delft: Delft University of Technology, June 2023.

[15] Sihao Sun et al. "High-Speed Flight of Quadrotor Despite Loss of Single Rotor". In: *IEEE Robotics and Automation Letters* 3.4 (Oct. 2018). Conference Name: IEEE Robotics and Automation Letters, pp. 3201–3207. DOI: 10.1109/LRA.2018.2851028. URL: https://ieeexplore.ieee.org/document/8398406 (visited on 09/28/2023).

[16] Sihao Sun et al. "Incremental Nonlinear Fault-Tolerant Control of a Quadrotor With Complete Loss of Two Opposing Rotors". In: *IEEE Transactions on Robotics* 37.1 (Feb. 2021). Conference Name: IEEE Transactions on Robotics, pp. 116–130. DOI: 10.1109/TRO.2020.3010626. URL: https://ieeexplore.ieee.org/document/9160894 (visited on 09/28/2023).

[17] Hussein Hamadi. "Fault-tolerant control of a multirotor unmanned aerial vehicle under hardware and software failures". en. PhD Thesis. Université de Technologie de Compiègne, July 2020.

[18] Peng Lu et al. "Active fault-tolerant control for quadrotors subjected to a complete rotor failure". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 4698–4703. DOI: 10.1109/IROS.2015.7354046.

[19] Anique V.N. Altena et al. "Loss-of-Control Prediction of a Quadcopter Using Recurrent Neural Networks". In: *Journal of Aerospace Information Systems* 20.10 (Oct. 2023). Publisher: American Institute of Aeronautics and Astronautics, pp. 648–659. DOI: 10.2514/1.I011231. URL: https://arc.aiaa.org/doi/10.2514/1.I011231 (visited on 03/21/2024).

[20] Paul Russell et al. *JSAT Loss of Control - CAST Approved Final Report Results and Analysis*. Technical Report. Federal Aviation Administration : Commerical Aviation Safety Team, Dec. 2000.

[21] Guy Gratton. "The Flight Envelope". In: *Initial Airworthiness: Determining the Acceptability of New Airborne Systems*. Dec. 2015, pp. 69–94. DOI: 10.1007/978-3-319-11409-5_4.

[22] E. R. Van Oort. "Adaptive Backstepping Control and Safety Analysis for Modern Fighter Aircraft". PhD thesis. Delft: Delft University of Technology, May 2011.

[23] Sanjiv Sharma et al. "Numerical continuation and bifurcation analysis in aircraft design: an industrial perspective". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 373.2051 (Sept. 2015). Publisher: Royal Society, p. 20140406. DOI: 10.1098/rsta.2014.0406. URL: https://royalsocietypublishing.org/doi/10.1098/rsta.2014.0406 (visited on 09/19/2023).

[24] Karim Ahmadi et al. "Active fault-tolerant control of quadrotor UAVs with nonlinear observer-based sliding mode control validated through hardware in the loop experiments". en. In: *Control Engineering Practice* 137 (Aug. 2023), p. 105557. DOI: 10.1016/j.conengprac.2023.105557. URL: https://linkinghub.elsevier.com/retrieve/pii/S0967066123001260 (visited on 11/06/2023).

[25] Sihao Sun et al. "Autonomous Quadrotor Flight Despite Rotor Failure With Onboard Vision Sensors: Frames vs. Events". en. In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 580–587. DOI: 10.1109/LRA.2020.3048875. URL: https://ieeexplore.ieee.org/document/9312462/ (visited on 11/06/2023).

[26] Timothy M. Lenton et al. "Tipping elements in the Earth's climate system". In: *Proceedings of the National Academy of Sciences* 105.6 (Feb. 2008). Publisher: Proceedings of the National Academy of Sciences, pp. 1786–1793. DOI: 10.1073/pnas.0705414105. URL: https://www.pnas.org/doi/full/10.1073/pnas.0705414105 (visited on 10/02/2023).

[27] Marten Scheffer et al. "Catastrophic shifts in ecosystems". en. In: *Nature* 413.6856 (Oct. 2001). Number: 6856 Publisher: Nature Publishing Group, pp. 591–596. DOI: 10.1038/35098000. URL: https://www.nature.com/articles/35098000 (visited on 10/02/2023).

[28] Didier Sornette et al. “Large financial crashes”. In: *Physica A: Statistical Mechanics and its Applications* 245.3 (Nov. 1997), pp. 411–422. DOI: 10.1016/S0378-4371(97)00318-X. URL: https://www.sciencedirect.com/science/article/pii/S037843719700318X (visited on 10/02/2023).

[29] Jose G. Venegas et al. “Self-organized patchiness in asthma as a prelude to catastrophic shifts”. en. In: *Nature* 434.7034 (Apr. 2005). Number: 7034 Publisher: Nature Publishing Group, pp. 777–782. DOI: 10.1038/nature03490. URL: https://www.nature.com/articles/nature03490 (visited on 10/02/2023).

[30] Manfred Schroeder. *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*. English. Mineola, N.Y: Dover Publications, Aug. 2009.

[31] Marten Scheffer et al. “Anticipating Critical Transitions”. en. In: *Science* 338.6105 (Oct. 2012), pp. 344–348. DOI: 10.1126/science.1225244. URL: https://www.science.org/doi/10.1126/science.1225244 (visited on 06/06/2023).

[32] Chunhua Zeng et al. “Noise and large time delay: Accelerated catastrophic regime shifts in ecosystems”. In: *Ecological Modelling* 233 (May 2012), pp. 52–58. DOI: 10.1016/j.ecolmodel.2012.03.025. URL: https://www.sciencedirect.com/science/article/pii/S030438001200141X (visited on 10/02/2023).

[33] Vasilis Dakos et al. “Spatial correlation as leading indicator of catastrophic shifts”. en. In: *Theoretical Ecology* 3.3 (Aug. 2010), pp. 163–174. DOI: 10.1007/s12080-009-0060-6. URL: https://doi.org/10.1007/s12080-009-0060-6 (visited on 10/02/2023).

[34] Carl Boettiger et al. “Early warning signals: the charted and uncharted territories”. en. In: *Theoretical Ecology* 6.3 (Aug. 2013), pp. 255–264. DOI: 10.1007/s12080-013-0192-6. URL: http://link.springer.com/10.1007/s12080-013-0192-6 (visited on 10/03/2023).

[35] Egbert H Van Nes et al. “Slow recovery from perturbations as a generic indicator of a nearby catastrophic shift”. In: *The American Naturalist* 169.6 (2007). Publisher: The University of Chicago Press, pp. 738–747.

[36] Valerie N Livina et al. “A modified method for detecting incipient bifurcations in a dynamical system”. In: *Geophysical research letters* 34.3 (2007). Publisher: Wiley Online Library.

[37] Vasilis Dakos et al. “Slowing down as an early warning signal for abrupt climate change”. In: *Proceedings of the National Academy of Sciences* 105.38 (Sept. 2008). Publisher: Proceedings of the National Academy of Sciences, pp. 14308–14312. DOI: 10.1073/pnas.0802430105. URL: https://www.pnas.org/doi/full/10.1073/pnas.0802430105 (visited on 10/02/2023).

[38] Stephen R Carpenter et al. “Rising variance: a leading indicator of ecological transition”. In: *Ecology letters* 9.3 (2006). Publisher: Wiley Online Library, pp. 311–318.

[39] Cees Diks et al. “Critical slowing down as an early warning signal for financial crises?” en. In: *Empirical Economics* 57.4 (Oct. 2019), pp. 1201–1228. DOI: 10.1007/s00181-018-1527-3. URL: https://doi.org/10.1007/s00181-018-1527-3 (visited on 08/30/2023).

[40] Anthony R Ives. “Measuring resilience in stochastic systems”. In: *Ecological Monographs* 65.2 (1995). Publisher: Wiley Online Library, pp. 217–233.

[41] Haoyu Wen et al. “How one might miss early warning signals of critical transitions in time series data: A systematic study of two major currency pairs”. en. In: *PLOS ONE* 13.3 (Mar. 2018). Publisher: Public Library of Science, e0191439. DOI: 10.1371/journal.pone.0191439. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0191439 (visited on 11/09/2023).

[42] Vasilis Dakos et al. “Flickering as an early warning signal”. In: *Theoretical Ecology* 6 (Aug. 2013). DOI: 10.1007/s12080-013-0186-4.

[43] Stephen R Carpenter et al. “Management of eutrophication for lakes subject to potentially irreversible change”. In: *Ecological applications* 9.3 (1999). Publisher: Wiley Online Library, pp. 751–771.

[44] Christoph Bandt et al. “Permutation Entropy: A Natural Complexity Measure for Time Series”. en. In: *Physical Review Letters* 88.17 (Apr. 2002), p. 174102. DOI: 10.1103/PhysRevLett.88.174102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.88.174102 (visited on 06/08/2023).

[45] Alfréd Rényi. “On measures of entropy and information”. In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. Vol. 4. University of California Press, 1961, pp. 547–562.

[46] Marta Borowska. “Entropy-Based Algorithms in the Analysis of Biomedical Signals”. In: *Studies in Logic, Grammar and Rhetoric* 43.1 (2016), pp. 21–32. DOI: doi:10.1515/slgr-2015-0039. URL: https://doi.org/10.1515/slgr-2015-0039.

[47] Francesco Carlo Morabito et al. “Multivariate Multi-Scale Permutation Entropy for Complexity Analysis of Alzheimer’s Disease EEG”. en. In: *Entropy* 14.7 (July 2012). Number: 7 Publisher: Molecular Diversity Preservation International, pp. 1186–1202. DOI: 10.3390/e14071186. URL: https://www.mdpi.com/1099-4300/14/7/1186 (visited on 11/07/2023).

[48] Benoit B. Mandelbrot et al. “The Fractal Geometry of Nature”. In: *American Journal of Physics* 51.3 (Mar. 1983), pp. 286–287. DOI: 10.1119/1.13295. URL: https://doi.org/10.1119/1.13295 (visited on 10/04/2023).

[49] H. E. Hurst. “Long-Term Storage Capacity of Reservoirs”. EN. In: *Transactions of the American Society of Civil Engineers* 116.1 (Jan. 1951). Publisher: American Society of Civil Engineers, pp. 770–799. DOI: 10.1061/TACEAT.0006518. URL: https://ascelibrary.org/doi/10.1061/TACEAT.0006518 (visited on 10/04/2023).

[50] Jan W. Kantelhardt et al. “Multifractal detrended fluctuation analysis of nonstationary time series”. In: *Physica A: Statistical Mechanics and its Applications* 316.1 (Dec. 2002), pp. 87–114. DOI: 10.1016/S0378-4371(02)01383-3. URL: https://www.sciencedirect.com/science/article/pii/S0378437102013833 (visited on 10/04/2023).

[51] Induja Pavithran et al. “Effect of rate of change of parameter on early warning signals for critical transitions”. en. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.1 (Jan. 2021), p. 013116. DOI: 10.1063/5.0025533. URL: https://pubs.aip.org/aip/cha/article/1059628 (visited on 06/05/2023).

[52] Vineeth Nair et al. “Multifractality in combustion noise: predicting an impending combustion instability”. en. In: *Journal of Fluid Mechanics* 747 (May 2014). Publisher: Cambridge University Press, pp. 635–655. DOI: 10.1017/jfm.2014.171. URL: https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/multifractality-in-combustion-noise-predicting-an-impending-combustion-instability/B5D2BC7ADB07899855AD264C717B5993 (visited on 10/04/2023).

[53] Kun Hu et al. “Endogenous circadian rhythm in an index of cardiac vulnerability independent of changes in behavior”. en. In: *Proceedings of the National Academy of Sciences of the United States of America* 101.52 (Dec. 2004). Publisher: National Academy of Sciences, p. 18223. DOI: 10.1073/pnas.0408243101. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC539796/ (visited on 10/04/2023).

[54] Dariusz Grech et al. “The local Hurst exponent of the financial time series in the vicinity of crashes on the Polish stock exchange market”. In: *Physica A: Statistical Mechanics and its Applications* 387 (July 2008), pp. 4299–4308. DOI: 10.1016/j.physa.2008.02.007.

[55] Jose Alvarez-Ramirez et al. “Time-varying Hurst exponent for US stock markets”. In: *Physica A: Statistical Mechanics and its Applications* 387 (Oct. 2008), pp. 6159–6169. DOI: 10.1016/j.physa.2008.06.056.

[56] Krzysztof Domino. “The use of the Hurst exponent to predict changes in trends on the Warsaw Stock Exchange”. In: *Physica A-statistical Mechanics and Its Applications - PHYSICA A* 390 (Jan. 2011), pp. 98–109. DOI: 10.1016/j.physa.2010.04.015.

[57] Vishnu R. Unni et al. “Multifractal characteristics of combustor dynamics close to lean blowout”. en. In: *Journal of Fluid Mechanics* 784 (Dec. 2015). Publisher: Cambridge University Press, pp. 30–50.

DOI: 10.1017/jfm.2015.567. URL: https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/multifractal-characteristics-of-combustor-dynamics-close-to-lean-blowout/F6B79AAC6E4D4FFE88B6CFC23A53A739 (visited on 10/04/2023).

[58] M. I. Jordan et al. "Machine learning: Trends, perspectives, and prospects". In: *Science* 349.6245 (July 2015). Publisher: American Association for the Advancement of Science, pp. 255–260. DOI: 10.1126/science.aaa8415. URL: https://www.science.org/doi/full/10.1126/science.aaa8415 (visited on 10/06/2023).

[59] Trevor Hastie et al. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY: Springer, 2001. DOI: 10.1007/978-0-387-21606-5. URL: http://link.springer.com/10.1007/978-0-387-21606-5 (visited on 10/06/2023).

[60] G. R. Kanagachidambaresan et al. "Recurrent Neural Network". en. In: *Programming with TensorFlow: Solution for Edge Computing Applications*. Ed. by Kolla Bhanu Prakash et al. EAI/Springer Innovations in Communication and Computing. Cham: Springer International Publishing, 2021, pp. 53–61. DOI: 10.1007/978-3-030-57077-4_7. URL: https://doi.org/10.1007/978-3-030-57077-4_7 (visited on 10/31/2023).

[61] Anirudh Koul et al. *Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow*. English. 1st edition. Beijing Boston Farnham Sebastopol Tokyo: O'Reilly Media, Nov. 2019.

[62] Junyoung Chung et al. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. arXiv:1412.3555 [cs]. Dec. 2014. URL: http://arxiv.org/abs/1412.3555 (visited on 10/31/2023).

[63] Kyunghyun Cho et al. *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. arXiv:1409.1259 [cs, stat]. Oct. 2014. DOI: 10.48550/arXiv.1409.1259. URL: http://arxiv.org/abs/1409.1259 (visited on 10/31/2023).

[64] Astrid Maritza González-Zapata et al. "Optimizing Echo State Networks for Enhancing Large Prediction Horizons of Chaotic Time Series". In: *Mathematics* 10.20 (2022). DOI: 10.3390/math10203886. URL: https://www.mdpi.com/2227-7390/10/20/3886.

[65] Herbert Jaeger. "Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach". In: (2002). Publisher: GMD-Forschungszentrum Informationstechnik Bonn.

[66] Abubakar Bala et al. "Applications of metaheuristics in reservoir computing techniques: a review". In: *IEEE Access* 6 (2018). Publisher: IEEE, pp. 58012–58029.

[67] Chenxi Sun et al. *A Review of Designs and Applications of Echo State Networks*. arXiv:2012.02974 [cs]. Dec. 2020. DOI: 10.48550/arXiv.2012.02974. URL: http://arxiv.org/abs/2012.02974 (visited on 10/31/2023).

[68] Ashish Vaswani et al. *Attention Is All You Need*. arXiv:1706.03762 [cs]. Aug. 2023. DOI: 10.48550/arXiv.1706.03762. URL: http://arxiv.org/abs/1706.03762 (visited on 11/01/2023).

[69] Vikram Mullachery et al. *Bayesian Neural Networks*. arXiv:1801.07710 [cs, stat]. Jan. 2018. URL: http://arxiv.org/abs/1801.07710 (visited on 11/01/2023).

[70] Soteris A. Kalogirou. "Chapter eleven - Designing and Modeling Solar Energy Systems". In: *Solar Energy Engineering*. Ed. by Soteris A. Kalogirou. Boston: Academic Press, 2009, pp. 553–664. DOI: https://doi.org/10.1016/B978-0-12-374501-9.00011-X. URL: https://www.sciencedirect.com/science/article/pii/B978012374501900011X.

[71] Timothy J. Ross. *Fuzzy Logic with Engineering Applications*. English. 2nd ed. New York: McGraw-Hill Inc.,US, Sept. 1994.

[72] S J K Kersbergen. "Quantifying Loss-Of-Control of Quadrotors". en. MA thesis. Delft: Delft University of Technology, Oct. 2018.