Title	Exploring unknown plant configurations under a multiple model adaptive control framework
Authors	Jesús Ares-Milián, Marlon;Provan, Gregory;Sohège, Yves;Quinones-Grueiro, Marcos
Publication date	2023-11-22
Original Citation	Ares-Milián, M.J., Provan, G., Sohège, Y. and Quinones-Grueiro, M. (2023) 'Exploring unknown plant configurations under a multiple model adaptive control framework', IFAC-PapersOnLine, 56(2), pp. 2933–2938. Available at: https://doi.org/10.1016/j.ifacol.2023.10.1415.
Type of publication	Article (peer-reviewed);journal-article
Link to publisher's version	10.1016/j.ifacol.2023.10.1415
Rights	© 2023 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/) - https://creativecommons.org/licenses/by-nc-nd/4.0/
Download date	2026-09-22 18:34:47
Item downloaded from	https://hdl.handle.net/10468/15549

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 2933–2938

Exploring Unknown Plant Configurations under a Multiple Model Adaptive Control Framework

Marlon Jesús Ares-Milián * **Gregory Provan** * **Yves Sohège** *
Marcos Quinones-Grueiro **

* *School of Computer Science and IT, University College Cork, Cork, Ireland (e-mail:* 122100743@umail.ucc.ie, g.provan@cs.ucc.ie, ysohege@ucc.ie.*)*
** *Institute for Software Integrated Systems, Vanderbilt University, Tennessee, USA (e-mail: marcos.quinones.grueiro@vanderbilt.edu)*

Abstract: Multiple model adaptive control (MMAC) provides robust control guarantees for a range of plant configurations given a set of models that correspond to known operating conditions (modes). However, these guarantees may not hold in the face of unknown system configurations. This paper focuses on the first step of control under novel modes: the task of novel configuration detection. We analyze a change-point detection strategy for unknown system configurations using input/output data as input. We characterize the performance of the change-point detection strategy based on whether the performance loss is bounded or unbounded. We experimentally validate our algorithms using a quadcopter trajectory-tracking benchmark, comparing our approach to a Bayesian change-point detection strategy.

Keywords: multiple-model adaptive control, plant parameter changes, changepoint detection.

1. INTRODUCTION

Controlling dynamic plants with an extensive and uncertain plant parameter space, which requires multiple operating modes, is a well-studied research field. Among the proposed solutions, multiple-model adaptive control (MMAC) approaches have been widely accepted by the control community thanks to their stability properties and satisfactory performance (Kersting and Buss, 2018).

One limitation of MMAC approaches is that stability is guaranteed only for the designed (known) plant parameter region. Since in most cases, it is not feasible to design controllers for all possible operating conditions *a priori*, developing methods for adaptation to unseen operating modes becomes a crucial task to ensure system survivability in the real world. Enabling the adaption of the MMAC framework to deal with unseen operating conditions can be viewed as a two-step process: (1) identifying when the system is operating in an unseen mode, and (2) developing a controller to provide stability. In this work, we provide a solution to the first step, laying the ground for solving the second one.

We frame the problem of identifying when the system is experiencing an unseen operating mode as a change-point detection task (Ragot, 2022), and we leverage deep learning techniques for finding a solution. One key real-world issue of novel modes is whether such a mode M causes instability or not. We examine this issue by characterizing the stability of novel modes in terms of bounds on the loss experienced in M, and analyzing the performance of our algorithm relative to the loss bounds for M.

Our main contributions can be summarized as follows:

- We present a methodology to identify unseen plant configurations allowing us to detect when an MMAC system operates in a mode where stable control is not guaranteed.
- We empirically analyze the relationship between loss experienced in the unknown mode and change-point detection performance.
- We validate our analysis using a trajectory-following benchmark for an unmanned aerial vehicle demonstrating superior performance to Bayesian change-point detection algorithms.

2. RELATED WORK

Addressing the variability in the parameter space of a system through a set of controllers with a control selection law is the main goal of MMAC (Anderson et al., 2000). How to attain this goal is the main topic discussed in (Kersting and Buss, 2018), where the plant parameter space is divided into sub-regions by analyzing the v-gap distance between regions as well as the generalized stability margin achieved by the region's associated controller. Once coverage of the plant parameter region has been achieved and the set of controllers has been tuned, designing a control law has been solved using both switching (Cheong and Safonov, 2012) and blending (Sohège et al., 2021) strategies.

On the topic of novelty and out-of-distribution detection, a significant amount of work exists (Yang et al. (2021)). Xie et al. (2021) reviews the particular problem of change-point detection. Parametric change-point detection is the

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2023.10.1415

approach selected in this paper, which is a topic that has been recently addressed through neural network-based solutions (Oza and Patel, 2019). Yu and Zhao (2020), address the topic of fault detection and isolation by studying the reconstruction error from a multi-layer autoencoder and its distribution when reconstructing nominal samples. All of the previous works have addressed the detection of deviations from known system configurations under single controller frameworks. However, the study of novel modes in an MMAC context is still open for research.

3. OPERATION MANIFOLD FOR MULTIPLE-MODEL CONTROLLER

In this article, we consider a non-linear system/plant $\mathcal{S}$ whose dynamics can be partitioned into multiple operating conditions. We define the system in state-space form as:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{1}$$

$$\mathbf{y} = g(\mathbf{x}, \mathbf{u}), \tag{2}$$

where $\mathbf{x} \in \Re^{n_x}$ is a state vector, $\mathbf{u} \in \Re^{n_u}$ is an input/control vector and $\mathbf{y} \in \Re^{n_y}$ is an observation vector. Functions f and g capture the plant and observation dynamics respectively. The system parameter set $\boldsymbol{\theta}$ describes the parameters of interest for $\mathcal{S}$, and may include a subset of system inputs, outputs, and states, as well as external disturbances. We want to design $\mathcal{S}$ to exhibit stability within a pre-defined range of conditions, which we characterize by $\hat{\boldsymbol{\theta}}$.

Definition 1. (Operation manifold). $\hat{\boldsymbol{\theta}}$ is the operation manifold (or range) for $\mathcal{S}$, and is the variation range of $\boldsymbol{\theta}$. $\hat{\boldsymbol{\theta}}$ can be defined as

$$\hat{\boldsymbol{\theta}} = \{\hat{\theta}_i : \ \underline{\theta}_i \leq \hat{\theta}_i \leq \bar{\theta}_i\}$$

where $\underline{\theta}_i$ and $\bar{\theta}_i$ are the lower and upper bounds of variable $\hat{\theta}_i$, which are assumed to be known *a priori*.

Our focus in this article is to determine when a system $\mathcal{S}$ ceases to operate within this operation manifold. We consider the case of using MMAC for control. The multiple-model approach partitions this operation manifold into a collection of sub-manifolds and designs a controller for each sub-manifold to guarantee stability. It has been shown in (Narendra and Balakrishnan, 1997) that it does not matter how the sub-manifolds are defined or the controllers are assigned, as long as a stable controller is designed for each sub-manifold.

3.1 Multiple-Model Control

To design a multiple-model controller $\mathcal{C}$, we assume that the nonlinear system has a family of equilibrium points.

Definition 2. A point $\pi = (\mathbf{x_e}, \mathbf{u_e}, \mathbf{y_e})$ is called an equilibrium point or operating point when it satisfies the steady state equations as follows:

$$f(\mathbf{x_e}, \mathbf{u_e}) = 0, \qquad \mathbf{y_e} = g(\mathbf{x_e}, \mathbf{u_e}). \tag{3}$$

Definition 3. (Equilibrium manifold). $\boldsymbol{\theta}_e$ is called the equilibrium manifold for a nonlinear system $\mathcal{S}$, which can be constructed from the equilibrium points as follows:

$$\boldsymbol{\theta}_e = \{(\mathbf{x_e}, \mathbf{u_e}, \mathbf{y_e}) : f(\mathbf{x_e}, \mathbf{u_e}) = 0, \text{ and } \mathbf{y_e} = g(\mathbf{x_e}, \mathbf{u_e})\} \tag{4}$$

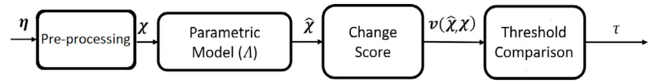

Fig. 1. Changepoint detection methodology.

We assume that $\boldsymbol{\theta}_e$ is constructed by defining N equilibrium points, and then generating a stabilizing controller for each. Many approaches have been defined for this construction, e.g., (Anderson et al., 2000). A standard approach consists of linearizing the nonlinear system around the i^{th} operating point $\pi_i = (\mathbf{x_e}, \mathbf{u_e}, \mathbf{y_e})_i$, for $i = 1, \cdots, N$, and then computing the set of models and controllers, e.g., see Anderson et al. (2000) and Kersting and Buss (2018).

3.2 Task Specification for Unknown Plant Configuration detection

We consider a family of trajectory following tasks, where a reference trajectory consists of a sequence of positions in a three-dimensional space, defined as: $\Sigma = \{\Sigma_1, \cdots, \Sigma_K\}, \Sigma_i \in \Re^3$. We assume that, given initial system state $\mathbf{x_0}$, we know the measurement trace, which consists of the input/output pair of $\mathcal{S}$ for time $t = 1, \cdots, n$: $\mathcal{O} = \{(\mathbf{y_1}, \mathbf{u_1}), \cdots, (\mathbf{y_n}, \mathbf{u_n})\}$. We define therefore a general task as follows:

Task 1. (Operation manifold containment). Given a nonlinear system $\mathcal{S}$, with initial state $\mathbf{x_0}$, multiple-model controller $\mathcal{C}$, and reference trajectory Σ, compute whether $\mathcal{S}$ exits the designed operation manifold based on measuring the trace $\mathcal{O}$.

4. CHANGEPOINT DETECTION METHODOLOGY

Figure 1 shows the proposed strategy to detect when the system exits the operation manifold for which the designed MMAC approach was designed. The main steps can be summarized as follows: 1) a feature vector $\boldsymbol{\chi}$ is constructed based on the system information $\boldsymbol{\eta} = \{\mathbf{y}, \mathbf{u}\}$ through signal processing techniques, 2) a parametric model characterizing the non-linear relationships of the system's variables generates a transformed feature vector $\hat{\boldsymbol{\chi}}$, 3) a change score is constructed through a change-point detection (CPD) method based on the generated feature vector and the original input $\upsilon(\hat{\boldsymbol{\chi}}, \boldsymbol{\chi})$, and 4) the change score is compared with a pre-defined threshold to generate a decision variable that determines transition time τ of the plant to an unknown plant configuration, i.e. outside the designed operation manifold.

4.1 Change score

The feature vector at each time-step $\boldsymbol{\chi}$ is a pre-processed combination of past inputs and outputs $(\mathbf{y_{t-1}}, \mathbf{u_{t-1}}, \mathbf{y_t})$ of the system S under multiple-model controller $\mathcal{C}$. We use an autoencoder, which is a popular parametric model given its non-linear modeling capabilities (Oza and Patel, 2019), to compute a reconstructed feature vector $\hat{\boldsymbol{\chi}} = \Lambda(\boldsymbol{\chi})$. The Autoencoder model is parameterized based on a dataset of feature samples generated under known system configurations (designed operation manifold). Given an Autoencoder model Λ, the change score is defined as a reconstruction error $\upsilon(\hat{\boldsymbol{\chi}}, \boldsymbol{\chi}) = MSE(\hat{\boldsymbol{\chi}}, \boldsymbol{\chi})$, where MSE represents the *mean-squared error* function.

4.2 Changepoint Detection

In order to detect when the system exits the designed operation manifold, a direct mapping is assumed between system configuration $\boldsymbol{\theta}$ and the input/output space of system $\mathcal{S}$. Variations in the system configuration are thus expected to impact the previously defined change score metric. Under this assumption, the following changepoint detection approach is defined.

Definition 4. (Changepoint Detection). Given a time-step sequence $t = 0, \cdots, T$, a changepoint $\tau \in [0, T]$ is defined as $\tau = \inf\{t > 0 : \upsilon(\Lambda(\boldsymbol{\chi}_t), \boldsymbol{\chi}_t) \geq \omega\}$.

where $\omega \in \Re$ is a threshold value.

4.3 Error threshold selection

In order to select the error threshold value ω, a dataset of feature samples, $X = \{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \dots\}$, generated under known system configurations, and control framework $\mathcal{C}$ is used. The change score metric is calculated for every sample in X, generating set $\Upsilon = \{\upsilon_1(.), \upsilon_2(.), \dots\}$. A threshold ω is then selected as the *Q-th* quantile of the probability distribution of the dataset Υ (Yu and Zhao, 2020). The quantile Q is a hyperparameter for the changepoint detection method.

5. QUADCOPTER BENCHMARK

An unmanned aerial vehicle (UAV) is selected as the application benchmark to test the proposed strategy. UAVs are complex non-linear systems that have been widely studied in the control literature (Miladi and Ladhari, 2020), and are subject to a wide range of operating modes and anomalous states. For these reasons, a single fixed controller is not able to stabilize all of the system configurations.

5.1 Multi-model control

In order to address the multiple operating modes for the quadcopter, a stochastic MMAC (SMMAC) architecture (Sohège et al., 2021) is implemented by tuning two sets of controllers: one for nominal conditions, and one for rotor loss of effectiveness conditions. This approach is an extension of MMAC, in that we introduce a randomized selection of blended control assignments. For more details on the overall control strategy, the reader is referred to our previous work (Sohège et al., 2021).

5.2 Trajectory tracking task and performance loss definition

A trajectory tracking task episode for the quadcopter consists in traversing the waypoint set Σ while minimizing the following loss function:

Definition 5. (Trajectory Tracking Loss). Given a reference trajectory Σ, the position of the vehicle along the executed trajectory $\rho \in \Re^3$, and a radius defining a safety flight zone envelope along the trajectory $\mathcal{R} \in \Re$, we define the trajectory tracking loss $\mathsf{L}_{0:T}$ as follows:

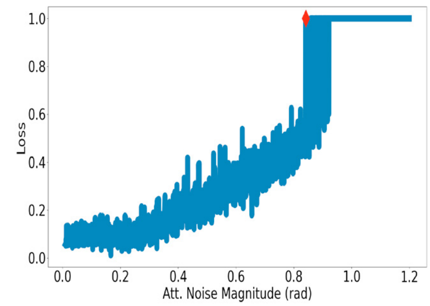

Fig. 2. Attitude noise performance analysis. The loss threshold is highlighted in red.

$$
\begin{aligned}
&\mathsf{L}_{0:T} = \mathsf{L}_{ct} + (1 - \mathsf{L}_{ct}) * \frac{\mathsf{L}_{ov}}{T};\ \mathsf{L}_{ov} = \sum_{t=0}^{T} \mathsf{L}_t, \\
&\mathsf{L}_{ct} = \begin{cases} 0, & \text{if trajectory completed} \\ 1, & \text{otherwise} \end{cases}, \\
&\mathsf{L}_t = \begin{cases} 1, & \text{if } |\Sigma_t - \rho_t| \geq \mathcal{R} \\ 0, & \text{otherwise} \end{cases}
\end{aligned} \tag{5}
$$

Episodes with a loss value of 1, are episodes for which mission completion (reaching the final goal) was not possible. Conversely, a loss value of 0 is associated with an episode that was completed without exiting the safety volume region at any time step. The trajectory tracking loss metric defined in this section relates to the control of the system and should not be confused with the anomaly detection task defined before in section 3.2.

5.3 Unknown configurations: Attitude Noise

Due to the complexity of the quadcopter model, we empirically explore unknown system configurations in order to produce stochastic loss guarantees for the implemented multi-model controller $\mathcal{C}$.

The case of attitude noise fault will be the main unknown configuration studied. Additive attitude noise ϑ_t is sampled at every timestep t from $\vartheta \sim U(-\hat{\vartheta}, \hat{\vartheta})$ for each of the attitude measurements (roll, pitch, and yaw); where $\hat{\vartheta}$ is the attitude noise magnitude. In order to establish empirical loss ($L_{0:T}$) guarantees, 500 linearly spaced attitude noise magnitudes (10 episodes per magnitude) are analyzed within the range $\hat{\vartheta} \in [0, 1.2]$ rad (Figure 2). An attitude noise magnitude threshold is selected at 0.84 rad. Smaller noise magnitudes allow trajectory completion ($max(L) < 1$) regardless of the number of time steps outside the safety envelope.

6. EXPERIMENTAL SETUP

The following section describes the implementation of the proposed changepoint detection strategy and its validation for the quadcopter benchmark.

6.1 Trajectory Randomization

We define 4 waypoints for every trajectory tracking episode following a closed convex quadrilateral shape in the XY plane at a fixed altitude ($z = 5m$). Only XY trajectories are studied in this paper in order to reduce computational complexity, however, the methodology proposed is seamlessly extendable to 3D trajectories. Distance (in meters) between every consecutive pair of waypoints is randomly sampled from $\mathcal{D} \sim U(5, 9)$. The angle between waypoints is also sampled randomly for each episode from $\mathcal{A} \sim U(80^o, 100^o)$. The orientation of the trajectory, i.e.: clockwise or counterclockwise, is also selected randomly for every episode.

6.2 Dataset generation

We define a feature vector, for each timestep, to combine the full measurements of the quadcopter state at two consecutive time steps, as well as the control action issued; and a maximum of 10000 time steps is set for each episode with simulation (and control) step of 0.02 s. Any episode that has not reached the goal by this time is considered a failed mission. Simulated episodes can be classified into three groups: episodes simulated fully under nominal conditions, episodes simulated with rotor loss of effectiveness faults, and episodes with attitude noise faults. We simulate abrupt permanent faults with a starting time sampled from a uniform distribution within the range $[10, 40]s$.

6.3 Parametric model tuning: Autoencoder

In our multilayer Autoencoder, the size of the latent dimension, the number of hidden layers, and the size of the hidden layers are all hyper-parameters that we selected through a standard grid search approach. We used the leaky ReLU activation function (He et al., 2015) for every layer of the model and Adam's optimization algorithm (Kingma and Ba, 2015) with the mean squared error as the loss function. The training and validation sets for Autoencoder model tuning were generated under known system configurations (nominal conditions and rotor loss of effectiveness conditions of up to 30%), with an MMAC framework as defined in previous sections. A total of 10000 samples were randomly collected for each dataset from 2000 trajectory tracking episodes.

Over 1 million samples from 4000 trajectory tracking episodes simulated under nominal and loss of effectiveness conditions were collected in order to select the error threshold ω for a specific value of the hyperparameter Q. The reconstruction error for every sample in the dataset was calculated and the threshold ω was selected as the *Q-th* quantile over the error distribution.

6.4 CPD post-processing

The CPD score presented is post-processed based on a time window $t - t_h, \ldots, t$ of data in order to make the final decision regarding changepoint detection. We define the post-processed CPD score as follows:

Definition 6. (CPD score II). Given an episode comprised of a timestep sequence $t = 0, \cdots, T$, and a time horizon t_h, a changepoint $\tau \in [0, T]$ is defined as a timestep such that $\upsilon(\Lambda(\boldsymbol{\chi}_t), \boldsymbol{\chi}_t) > \omega$ for $\frac{t_h}{2}$ or more timesteps within the time interval $\tau - t_h, \ldots, \tau$.

The time horizon t_h is a hyperparameter to be tuned. A high t_h value can increase detection delay but will increase the confidence in the detection decision.

6.5 Evaluation metrics

We define a ground truth changepoint $\tau^*[s]$ as the time instant when the system undergoes a change of operating mode. In order to assess the performance of the proposed approach, a changepoint τ is considered to be correctly detected if $\tau > \tau^*$. A detection alarm triggered before the ground truth changepoint ($\tau \leq \tau^*$) is considered a false positive. Similarly, if multiple changepoints are detected after the ground truth changepoint, only the first alarm is considered as correct, the rest are deemed false positives. The effectiveness of a CPD algorithm can therefore be expressed through a receiver operating characteristic (ROC) curve and its associated area under the curve (AUC) metric, which summarizes the relationship between the true positive rate (TPR) and the false positive rate (FAR):

$$TPR = \frac{N_{CR}}{N_{GT}}; \quad FAR = \frac{N_{AL} - N_{CR}}{N_{AL}} \tag{6}$$

where N_{CR} is the number of correctly detected ground truth changepoints, N_{GT} is the number of ground truth changepoints, and N_{AL} is the total count of triggered detection alarms. As a complementary performance measure for correctly detected changepoints, the average detection delay $\Delta\bar{\tau}$ is defined as:

$$\Delta\bar{\tau} = \frac{\sum_i^{N_{CR}} \tau - \tau_i^*}{N_{CR}} \tag{7}$$

7. RESULTS AND DISCUSSION

We performed two main groups of experiments to evaluate the proposed methodology. An initial group (Section 7.1) evaluates the performance of the CPD algorithm when dealing with unknown configurations for which no loss guarantees can be provided. A second group of experiments (Section 7.2) explores CPD performance for unknown system configurations with empirical loss guarantees. Finally, Section 7.3 compares the proposed CPD methodology with a well-known Bayesian CPD approach.

7.1 Configurations with no empirical loss guarantees.

We first evaluate the ability of the proposed CPD methodology to detect transitions from known to unknown system configurations with no loss guarantees, i.e., attitude noise conditions for which mission completion cannot be guaranteed.

We compiled a test dataset by combining simulated trajectory tracking episodes under the following conditions: 1000 nominal operation episodes, 1000 episodes with rotor loss of effectiveness faults (from 0.01% to 30% loss of effectiveness), and 2000 attitude noise episodes (from 0.84 rad to 1.2 rad).

We compared performance for different hyperparameter configurations of t_h and Q. The time horizon parameter was selected from the set: $t_h \in \{1, 10, 30, 40, 60\}$; and the threshold error quantile parameter was selected from the

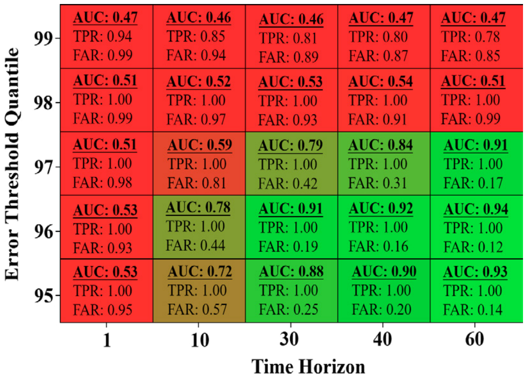

Fig. 3. AUC score comparison.

set $Q \in \{95, 96, 97, 98, 99\}$. This exploration resulted in a five-by-five grid of changepoint detection models and both, AUC score and average detection delay, were calculated for each of the 25 models.

Figure 3 shows the performance for different hyperparameter configurations. There is an improvement in performance as the time horizon increases. This is an expected result since an increase in the time horizon provides the CPD framework with more information to make a decision. The results presented in Figures 3 and 4 depict the trade-off between detection performance and detection delay.

An increase in the quantile parameter past the value of 96 results in an overall decrease in the AUC score. Increasing the error threshold, makes the attitude noise samples harder to identify, which reduces the number of false positives under known conditions. However, after the system encounters attitude noise conditions, a high error threshold causes the CPD algorithm to repeatedly (and erroneously) detect transitions back to the known operating modes, resulting in a high number of false positives (only the detected changepoint closest to the ground truth changepoint is considered a true positive, the following alarms are considered false positives).

All hyperparameter configurations with $t_h \geq 30$ and $Q \leq 97$ present good AUC scores. The hyperparameter selection must then depend on the acceptable detection delay for the system. For this reason, an analysis of average detection delay is presented in Figure 4, where an increase in the t_h parameter shows a higher detection delay. This is a natural consequence of delaying the detection decision until enough information has been gathered. Detection delay also increases significantly as the quantile hyperparameter increases due to the fact that the attitude noise samples become harder to detect as the reconstruction error threshold increases.

7.2 *Configurations with empirical loss guarantees.*

Here we explore unknown parameter configurations for which mission completion can be guaranteed. A dataset was generated comprising 2000 trajectory tracking episodes under attitude noise conditions from 0.01 to 0.84 rad. Due to the loss guarantees for these configurations under the available controller, overall CPD performance is not as relevant. For this reason, we are mostly interested in studying detection accuracy of transitions for different attitude noise values. Therefore, we evaluated TPR and average detection delay for all attitude noise magnitudes within the dataset, and compared the different quantile hyperparameter configurations, fixing t_h to 30.

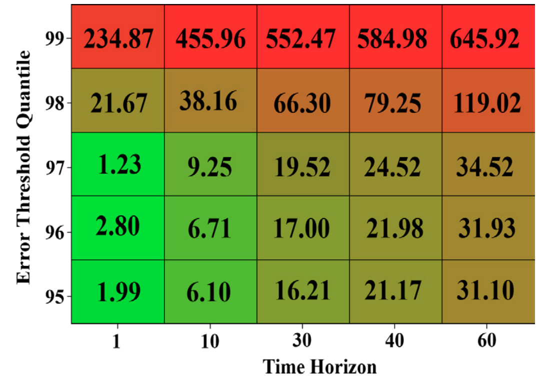

Fig. 4. Average detection delay (in number of timesteps) comparison.

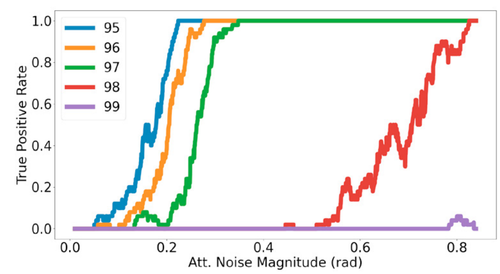

Fig. 5. True positive rate for stable attitude noise. Comparing quantile (Q) configurations.

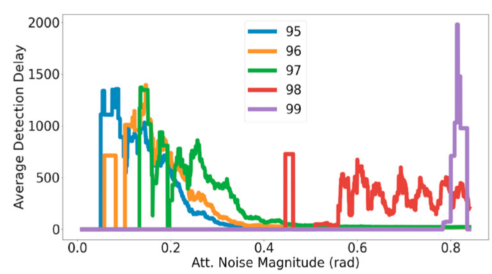

Fig. 6. Average detection delay for stable attitude noise. Comparing quantile (Q) configurations.

Figure 5 shows an increase in TPR as the attitude noise magnitude increases. These results mirror those in Figure 2, indicating that a shift in system configuration towards an unknown fault associated with a higher trajectory tracking loss is more likely to be detected.

Figure 6 presents the average detection delay for different attitude noise magnitudes. The behaviour for quantile values of 98 and 99 does not present a clear distribution pattern due to the low detection rates. The remaining quantile values present a clear pattern: detection delay is zero for low attitude noise magnitudes because the change in system dynamics is too small, causing null detection rates (Figure 5); as the noise magnitude increases, the changes in system dynamics start becoming detectable, however, they are not obvious enough to be detected quickly.

7.3 Comparison with Bayesian CPD strategy

The proposed CPD strategy was compared to a baseline Bayesian CPD strategy (Xuan and Murphy (2007)). The codebase used for comparison can be found at `https://github.com/hildensia/bayesian_changepoint_detection`. The feature vector $\boldsymbol{\chi}$ presented in Section 4.1 is used as input for both CPD strategies. Only transitions to unknown configurations with no loss guarantees were studied, using the datasets presented in Section 7.1.

	AUC	TPR	FAR	$\Delta\bar{\tau}$
Bayesian CPD	0.43	0.81	0.95	5.67
Parametric CPD ($t_h = 10$, $Q = 95$)	0.75	0.99	0.50	5.87
Parametric CPD ($t_h = 60$, $Q = 96$)	0.95	1.00	0.11	31.66

Table 1. Comparing CPD Strategies.

The Bayesian CPD strategy presents a moderately good TPR of 0.81 (Table 1). However, it also presents a massive FAR at 0.95, which is a prohibitively high value for a CPD algorithm.

Two hyperparameter configurations were chosen to evaluate the parametric CPD proposed in this work. The first configuration was evaluated for $Q = 95$ and $t_h = 10$. This configuration was selected since it presents a similar detection delay to the Bayesian CPD method. The second configuration is the configuration that produced the highest AUC score in Section 7.1

Both parametric solutions outperform the Bayesian approach significantly. This difference in performance may be based on multiple reasons. Firstly, the learning-based nature of the parametric model allows it to differentiate known from unknown configurations while identifying known configurations as similar. This is of particular importance due to the fact that transitions are being estimated by observing the input/output space, and, in the context of MMAC, known configurations span a wide range of input/output dynamics, among which, transitions can be mistakenly identified as change-points by the Bayesian CPD approach. Secondly, the high dimension of the feature vector ($\boldsymbol{\chi}$) is also a potential advantage for our approach, due to the efficient representation of the data forced by the Autoencoder architecture.

8. CONCLUSION

In this article, we present a parametric changepoint detection method using a deep Autoencoder model to detect when a system that is controlled through MMAC transitions outside the known plant parameter space. Using offline analysis, we empirically demonstrate that unknown system configurations with higher loss values for the control task present higher change-point detection rates as well as lower detection times. A quadcopter trajectory tracking task subject to varying faults and disturbances is used for experimental validation. Finally, the proposed methodology is shown to outperform a Bayesian change-point detection strategy. This is the first step towards generating online adaptive control methods for systems experiencing unknown operating conditions. We leave the problem of simultaneously detecting and providing loss guarantees for unknown configurations open for future works.

REFERENCES

Anderson, B.D., Brinsmead, T.S., De Bruyne, F., Hespanha, J., Liberzon, D., and Morse, A.S. (2000). Multiple model adaptive control. part 1: Finite controller coverings. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 10(11-12), 909–929.

Cheong, S.Y. and Safonov, M.G. (2012). Slow-fast controller decomposition bumpless transfer for adaptive switching control. *IEEE Transactions on Automatic Control*, 57(3), 721–726.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV2015)*, 1026–1034. IEEE.

Kersting, S. and Buss, M. (2018). How to Systematically Distribute Candidate Models and Robust Controllers in Multiple-Model Adaptive Control: A Coverage Control Approach. *IEEE Transactions on Automatic Control*, 63(4), 1075–1089.

Kingma, D.P. and Ba, J.L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Miladi, N. and Ladhari, T. (2020). EnMPC versus PID control strategies applied to a Quadcopter. In *New Trends in Robot Control. Studies in Systems, Decision and Control*, volume 270, chapter 10, 319–334.

Narendra, K. and Balakrishnan, J. (1997). Adaptive control using multiple models. *IEEE Transactions on Automatic Control*, 42(2), 171–187.

Oza, P. and Patel, V.M. (2019). C2AE: Class conditioned auto-encoder for open-set recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2302–2311.

Ragot, J. (2022). Detection of operating mode changes, without a priori model and in uncertain environments. *Transactions of the Institute of Measurement and Control*, 44(13), 2653–2671.

Sohège, Y., Quiñones-Grueiro, M., and Provan, G. (2021). A Novel Hybrid Approach for Fault-Tolerant Control of UAVs based on Robust Reinforcement Learning. *Proceedings - IEEE International Conference on Robotics and Automation*, 2021-May, 10719–10725.

Xie, L., Zou, S., Xie, Y., and Veeravalli, V.V. (2021). Sequential (Quickest) Change Detection: Classical Results and New Directions. *IEEE Journal on Selected Areas in Information Theory*, 2(2), 494–514.

Xuan, X. and Murphy, K. (2007). Modeling changing dependency structure in multivariate time series. *ACM International Conference Proceeding Series*, 227, 1055–1062.

Yang, J., Zhou, K., Li, Y., and Liu, Z. (2021). Generalized Out-of-Distribution Detection: A Survey. 1–22. URL `http://arxiv.org/abs/2110.11334`.

Yu, W. and Zhao, C. (2020). Robust monitoring and fault isolation of nonlinear industrial processes using denoising autoencoder and elastic net. *IEEE Transactions on Control Systems Technology*, 28(3), 1083–1091.

This publication has emanated from research supported in part by Grants from Science Foundation Ireland under Grant numbers 13/RC/2094 and 12/RC/228.