

Learning Switching Dynamical Models for Abnormality Detection in Autonomous Quad-copters

Mohammad Rahmani

mohammad.rahmani@aau.at

Networked and Embedded Systems
Klagenfurt University
Austria

Berhard Rinner

bernhard.rinner@aau.at

Networked and Embedded Systems
Klagenfurt University
Austria

October 27, 2021

Abstract

In this paper, we extend a Switching Dynamical Linear System (SDLS) based on a multi level Dynamic Bayesian Network (DBN) from 4-dimensional state spaces to state spaces of arbitrary dimensions. The optimized number of discrete variables of the SDLS which form the discrete level of the DBN is determined by Growing Neural Gas (GNG). Additionally, a dynamic method is used to determine the neighborhoods of these discrete variables where quasi-static motions are valid. We examine its efficiency with three abnormality indicators for 6-dimensional training data derived from a simulated quad-copter by using three deviated scenario from a reference scenario according to which the DBN is trained.

1 Introduction

A self-aware (SA) robot can become the subject of its own attention such that it can process its own sensory data to understand experiences through which it goes [1]. In recent years, to improve autonomy and scalability of artificial systems, SA have been implemented in a vast variety of areas from self-aware cameras [2] to robotics [3]. Inference and anomaly detection are two important capabilities of any self-aware robot [4]. Inference provides the robots with the ability to predict their future states [5] while abnormality detection makes them capable of noticing the difference between their observations and predictions [6]. Hence, robots will be capable of detecting new experiences to improve their inference models [7].

This paper proposes a modeling, inference and abnormality detection for an autonomous robot following the ideas of self-awareness. Figure 1 presents an overview of this approach to estimate the current state of a robot and derive abnormality indicators. During offline learning, the robot follows a reference trajectory multiple times and captures positional data from its GPS sensor. Using Growing Neural Gas (GNG), the derived data is partitioned into clusters from which a set of regions with quasi-static motion is derived. These clusters are used to learn the probabilistic parameters of a Dynamic Bayesian

Network (DBN) [8]. During online prediction and abnormality detection, this DBN is used together with the current sensor measurements to compute abnormality signals which indicate whether the currently observed behavior is matching the predicted behavior from the DBN.

The main contributions of this paper is extending a DBN-based model of a single robot behavior and assessing its inference and abnormality detection performance. It is different than the previous work [9] in the following way: First, regions of quasi-static motions are represented by neighborhoods that their extensions are determined dynamically according to measurements in their vicinity. Furthermore, Mahalanobis distance is used to cluster measurements and compute the likelihood of the variables at different levels of the proposed DBN. Additionally, Growing Neural Gas (GNG) is used to dynamically determine the optimal number of clusters needed at discreet level. Finally, it has extended the model to any dimension and tested it by 6-dimensional data derived from a simulated quad-copter.

The reminder of this paper is organized as follows: Section 2 discusses related work. Section 3 describes the offline phase of our approach, and Section 4 summarizes the online phase. Section 5 presents the conducted simulation study, and Section 6 concludes the paper with a summary and outlook about future

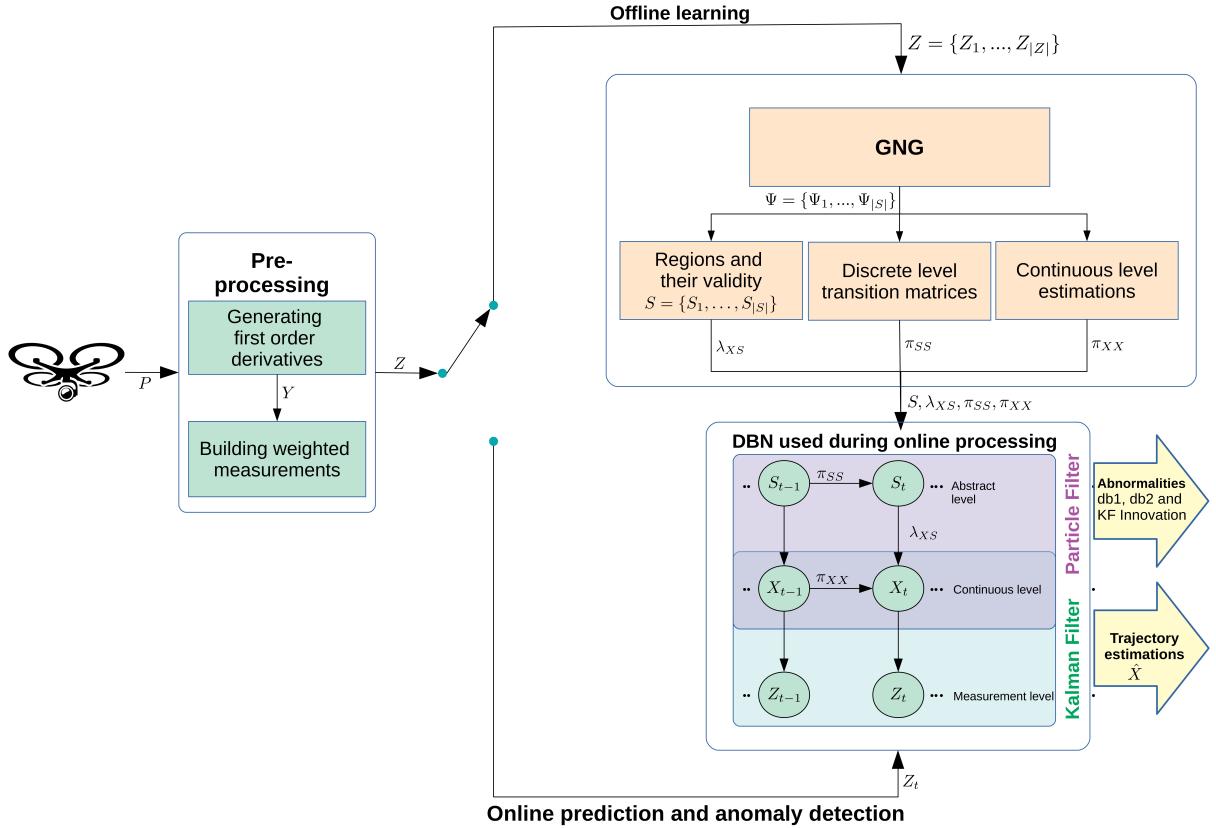


Figure 1: The proposed solution consist of two phases. The offline learning phase uses a sequence of measurements from a reference scenario to learn the necessary probabilistic parameters of a DBN. The online phase exploits the learned probabilistic DBN models to estimate the state and generate abnormality signals. In the DBN, the horizontal arrows present temporal causalities while the vertical arrows present causalities between the continuous and discrete levels

work.

2 Related works

Three major, bio-inspired, self-awareness theories, Damasio [10, 11], Haykin [12] and Friston [13] consider hierarchical modeling and temporal reasoning as important properties that every biological, self-aware agent should incorporate. DBNs are hierarchical data driven models which can be used to infer the agent's future states (temporal reasoning) and detect abnormalities. They use Bayesian filtering methods [14] at different levels of their hierarchy to predict, estimate the current state and detect abnormalities according to the most recent sensory measurements. They build temporal relations between the states according to the data derived from an already known experience and consequently, during online testing, they update the posterior probability of robot states based on this measurement.

Switching Dynamical Linear System (SDLS)[15, 16, 17] are special DBNs that enable to represent a dynamic non-linear model as a sequence of linear models. They use switching, discrete random variables (super states) associated with these linear dynamic models at higher levels of the DBN to recursively estimate the joint posterior of state and the switching variables. Markov Jump Particle Filter (MJPF) [9] is one such filtering system which uses 2-Time Slice DBN [18] SDLS for inference and abnormality detection at lower continuous and higher discrete level of its hierarchy. At continuous level, it uses a bank of Kalman Filters [19] which are associated with linear dynamic models derived from super states as Kalman Filter is only applicable in linear dynamic systems. These super states corresponds to particles of a Particle Filter [20] at discrete level from which the aforementioned Kalman Filters get the control input of the dynamic system using which they predict the next state.

MJPF has been used in a variety of dynamic systems. For example [21, 22] has used it for abnormality detection and implementing collective awareness in connected autonomous vehicles. [23] has applied it for anomaly detection in video data while [24] has used the same idea with different abnormality indicators for jammer detection in cognitive radios.

3 Offline learning

State space definition. Since the goal in this paper is to estimate the state using regions of state space in which quasi-static velocities are valid, we

consider the state space as

$$X_t = [x_1, \dots, x_n, \dot{x}_1, \dots, \dot{x}_n]^T \quad (1)$$

where x_1, \dots, x_n represent 0th-order derivatives (position) and $\dot{x}_1, \dots, \dot{x}_n$ represent corresponding first-order derivatives (velocities) at time step t . To accomplish this goal, let $p_t = [p_1, \dots, p_n]$ be the noisy measurement for the 0th-order derivatives of a state of a dynamical system. We define

$$Z = \{D[p_t, \frac{p_t - p_{t-1}}{\Delta t}]^T \mid t \in \{1, \dots, |Z|\}\} \quad (2)$$

as a sequence weighted measurements where Δt is the sampling time and

$$D = \begin{pmatrix} \alpha I_n & 0_n \\ 0_n & \beta I_n \end{pmatrix} \quad (3)$$

such that $\beta > \alpha$. This multiplies the 0th derivative part (position) of the each member of Z by α and the 1st derivative part (velocities) by β . Such multiplication causes the clustering algorithm that we apply later on Z to prefer changes in velocities to cluster Z rather than positions.

Learning discrete variable. To cluster the weighted measurements Z , we use Growing Neural Gas (GNG) [25] which is an incremental clustering algorithm that can be used to cluster vectors without pre-defining the number of clusters in advance. Additionally, parameters of this algorithm such as local and global and scaling errors are just needed to be set once before running it. GNG detects a set of nodes, $\Psi = \{\Psi_1, \dots, \Psi_{|S|}\}$, in state space from which we measure the Mahalanobis distance for each weighted measurement $d(Z_i, \Psi_j)$. represents Mahalanobis distance between the weighted measurement Z_i and the node Ψ_j , then Z_i belongs to the cluster with the minimum distance $d(Z_i, \Psi_j)$ of all nodes. As such we cluster Z into $|S|$ clusters.

$$S = \{S_1, \dots, S_{|S|}\} \quad (4)$$

Each of these clusters can be presented by the mean and covariance matrix of the measurements they include. Each mean and covariance matrix pair (μ_i, Σ_i) is referred as a super state. The super state S_j is said to be activated if the measurement Z_i is a member of the neighborhood

$$C(\mu_j, 3\sqrt{tr(\Sigma_j)}) \quad (5)$$

such that μ_j is its center and $3\sqrt{tr(\Sigma_j)}$ is its radius where $tr(\cdot)$ is the trace operator of Σ_j . We refer to these neighborhoods as known regions. The radius of these neighborhoods are dynamic since they

vary depending on clusters' members. Regions outside these neighborhoods are referred as unknown regions. Defining known regions using dynamic radius in comparison to static regions in [9] limits better noisier measurements in the probabilistic parameters calculations we discuss later.

Learning super states transition matrices. Each two consecutive members of Z activate two consecutive super states of S . Counting consecutive activated super states for all members of Z gives a probability transition matrix such as:

$$\pi_{SS} = \{p(S_{t-1}^i | S_t^j, s) \mid i, j \in \{1, \dots, |S|\}\} \quad (6)$$

to predict the probability of staying in a known region or moving to another known region conditioned on knowing the previous super state after s time steps.

Process model at continuous level Each μ_i for each super state, similar to each member of Z in Equation 2 is formed of position components and corresponding velocities. The velocity part of each μ_i can be used as the control input in a process model in order to predict the next continuous state based on the current state. Mathematically, we can define the process model in continuous state space as follows

$$X_t = FX_{t-1} + BU_{S_{t-1}} + \nu_{t-1} \quad (7)$$

where

$$F = \begin{pmatrix} I_n & 0_n \\ 0_n & 0_n \end{pmatrix} \quad (8)$$

is the transition matrix. $B = [I_n \Delta t \ I_n]^T$ is the control matrix where Δt is sampling time, $U_{S_{t-1}} = [I_n \ 0_n] \mu_{S_{t-1}}$. The process noise ν_{t-1} is drawn from a multi variate normal distribution such that $\nu_{t-1} \sim \mathcal{N}(0, \Sigma_{S_t})$. Whenever the previous state doesn't match any known region, $U_{S_{t-1}}$ is set to $0_{n,1}$. Similar to Equation 6, we can define a set of probabilistic models such that

$$\pi_{XX} = \{p(X_t | X_{t-1}, S_{t-1}^i)\} \mid i \in \{1, \dots, |S|\}\} \quad (9)$$

The likelihood of continuous states belong to a super state Furthermore, the likelihood of continuous states belonging to a super states can be defined as

$$\lambda_{XS} = \{p(X_t | S_t)\} \mid i \in \{1, \dots, |S|\}\} \quad (10)$$

where

$$p(X_t | S_t) = \frac{d(X_t, \mu_t)}{\sum_{i=1}^{|S|} d(X_t, \mu_i)}. \quad (11)$$

4 Online processing

Having π_{SS} , π_{XX} and λ_{XS} computed using the methods described in previous section, with given set of measurements, we can extend MJPF to any dimension with the definition of known regions. This model makes inferences using the learned DBN (cp. Figure 1). It uses a Particle Filter which is coupled with a bank of Kalman Filters at the continuous and discrete levels jointly for estimation. At the continuous level, states are inferred from measurements and π_{XX} predictions are computed by the Kalman Filters corresponded to each region. At the discrete level, π_{SS} is used by the Particle Filter for future estimations. These two levels are related to each other by using the means of the super states to which particles correspond as the control inputs in each Kalman Filter.

Particle Filter The Particle Filter uses a set of weighted vectors in the state space to estimate the posterior of an unknown probability distribution. Intuitively, these weights present the possibility of having solutions close to these corresponding states according to the most recent measurement and a given importance function, q , from which we sample instead of the unknown probability distribution. Since a sequential importance resampling (SIR) is used for the Particle Filter in this work, the importance function is taken as $q = p(S_t | S_{t-1})$. To estimate the posterior of X_t based on the most recent measurement Z_t , let $V = \{v_t^{(i)} = (X_t^{(i)}, w_t^{(i)}) \mid i \in \{1, \dots, |V|\}\}$ be the set of particles at time step t where $X_t^{(i)}$ is the particle's associated state at the continuous state space, and $w_t^{(i)}$ is its corresponding weight. The associated state of each particle $v_t^{(i)}$ is either within a known region or in an unknown region. The symbol we use hereafter for such region is $S_t^{(i)}$.

Particle movement between regions According to most recent measurement, we consider three different transition probabilities for a particle between regions $v_t^{(i)} = (X_t^{(i)}, w_t^{(i)})$. When a particle has moved from one known region to another known region, the probability is q multiplied by the probability that it stays in the same region after a given amount of time. When it has moved from a known region to an unknown region, then it is $1 - p(S_t | S_{t-1})$ where S_t is the region with its mean closest to the particle's state. Finally, when a particle in an unknown region moves to a known region with μ_j and Σ_j , then the probability is taken as the maximum of $\{0, 1 - \frac{d(X_t^{*}, \mu_j)}{\mu_j + 3\sqrt{\text{tr}(\Sigma_j)}}\}$.

Computing the weight of the particles To compute the weight of the particles when new measurements become available Z_t , we need to compute three probabilities. The first is

$$p(Z_t|X_t^{(j)}) = \frac{d(Z_t, X_t^{(j)})}{\sum_{i=1}^{|V|} d(Z_t, X_t^{(i)})} \quad (12)$$

The second is

$$p(X_t^{(j)}|S_t^{(j)}) = \frac{d(X_t^{(j)}, \mu_t^{(j)})}{\sum_{i=1}^{|V|} d(X_t^{(i)}, \mu_t^{(j)})} \quad (13)$$

where $\mu_t^{(i)}$ is the mean corresponded to super state $S_t^{(i)}$. The last probability is $p(X_t^{(i)}|X_{t-1}^{(i)}(S_{t-1}^{(i)}))$ in which $X_{t-1}^{(i)}(S_{t-1}^{(i)})$ means the probability depends on the super state $S_{t-1}^{(i)}$. This probability is computed by using the the Kalman Filter corresponding to super state $(S_{t-1}^{(i)})$ with control input $U_{S_{t-1}}$ in Equation 7. As such, the weights of particles are approximated as

$$W_t = W_{t-1} \sum_{i=1}^{|V|} p(Z_t|X_t^{(i)}) p(X_t^{(i)}|S_t^{(i)}) p(X_t^{(i)}|X_{t-1}^{(i)}(S_{t-1}^{(i)})) \quad (14)$$

where W_{t-1} is

$$W_{t-1} = \exp^{-(db1+db2)} \quad (15)$$

such that

$$db1 = -\ln \sum_{i=1}^{|V|} \sqrt{p(X_t^{(i)}|S_t^{(i)}) p(X_t^{(i)}|X_{t-1}^{(i)}(S_{t-1}^{(i)}))} \quad (16)$$

and

$$db2 = -\ln \sum_{i=1}^{|V|} \sqrt{p(Z_t|X_t^{(i)}) p(X_t^{(i)}|X_{t-1}^{(i)}(S_{t-1}^{(i)}))} \quad (17)$$

$db1$ and $db2$ are the Bhattacharyya distances [26] between $p(X_t^{(i)}|X_{t-1}^{(i)}(S_{t-1}^{(i)}))$, the particle's state prediction, and $p(X_t^{(i)}|S_t^{(i)})$ which is the probability of particle being inside a predicted region and $p(Z_t|X_t^{(i)})$ which measures how much the particle is likely according to measurement Z_t .

Abnormality indicators As it implies from the definition of $db1$ and $db2$, they are good indicators to detect deviations from a learned model. $db1$ presents the distance between the predicted super state and the probability of being in that super state. Intuitively, it indicates if particles are coherent with the predicted super state of the learned model. Large values of this indicator show that the current experience

is outside the known regions. In other words larger values denote measurements outside the domain of known regions. Simply put, it presents the distance between the predicted super state and the probability of being in that super state. This indicator considers only the known region in which the agent is and ignores its velocity vector. On the contrary, $db2$ can be attributed to the similarity between the state prediction and the continuous state evidence related to the new measurement in each super state. That is, it denotes the difference between the expected prediction and the likelihood behavior. Finally, since the Kalman Innovation is defined as

$$g = d(Z_t, H\hat{X}_{t|t-1}) \quad (18)$$

where H is the measurement matrix and $\hat{X}_{t|t-1}$ is the prediction of state at t given the state at $t-1$, it can be used as an another abnormality indicator. Similar to $db2$, it presents the distance between the predicted state and current measurement. As such in any experiment, we expect these two values to approximate each other.

5 Experiments

In this section, the robot's DBN model learns the parameters mentioned in Section 3 from a sequence of measurements generated from a GPS sensor of a simulated quad-copter (see Figure 3) while following a reference scenario multiple times (see Figure 2). In the online processing, the quad-copter follows deviated scenarios from the aforementioned scenario and generates the abnormality signals described in Section 4 to assess quad-copters ability in detecting deviations from learned scenarios. To generate the aforementioned measurement sequence, we deploy the CTU-MRS simulation framework [27]. This framework uses Robot Operating System (ROS) [28] to control quad-copters and Gazebo [29] to graphically realize the simulations. In each of the following experiments, a set of points of the reference scenario (input trajectory) is fed to a ROS service of a quad-copter realized in this framework. The quad-copter then follows these points using an autopilot based on Model Predictive Control (MPC) [30]. As such, it is possible to record the values generated by a ROS topic that outputs GPS sensor data (observed trajectory). These positions are 3-dimensional vectors such as $P_t = [x_1, x_2, x_3]^T$ at time step t . The GPS sensor noise is assumed to be zero mean Gaussian. The covariance matrix of sensor's noise is considered to be equal in all regions to $0.127 \times I_6$. We take $\alpha = 1$ and

$\beta = 20$ in Equation 3 to build members of Z similar to definition of Equation 2.

Scenario 1, reference scenario, Threshold values In this scenario, a quad-copter is requested to follow a square trajectory as illustrated in Figure 2 10 times with constant velocity magnitude. The autopilot completes each cycle in 4000 time steps and generates a total of 40000 position data points which are used to learn the DBN parameters as discussed in Section 3. After pre-processing, the resulting sequence of weighted measurements are fed to a GNG for clustering. In this experiment the following GNG parameter setting is used: maximum number of nodes: 100; maximum age for edges: 100; maximum number of iterations: 3000; local error rate: 0.5; global error rate: 0.0005; and scaling amount: 0.5. These values are chosen according to [31, 32] which have conducted studies on the optimal settings for GNG.

Figure 4 presents the clusters of the measurements from this experiment such that each color presents a different cluster. Feeding the learned DBN with the same sequence of measurements, we can compute abnormality values for all three indicators using which it is possible to define threshold values for each of them by taking their mean plus their variance as threshold. In this experiment these values are 0.25 for $db1$ and 0.32 for $db2$ and the Kalman Innovation.

Scenario 2, obstacle avoidance by horizontal turning In this scenario, the quad-copter turns left in the middle of the first side of the square as illustrated in Figure 5 to avoid an imaginary obstacle using its autopilot’s MPC controller. Considering this deviation from the learned model in scenario 1, abnormality values are expected to exceed the threshold values approximately from time steps 333 to 1333 which is the time interval that the deviation from the learned scenario (scenario 1) happens. Figure 6 illustrates quad-copter’s GPS sensors recordings while performing scenario 2.

Figure 7 shows the abnormality values derived from the learned DBN model and the corresponding GPS data. As it is expected, since both $db2$ and Kalman Innovation measure the distance between the new measurement and the continuous state prediction inside each region, then we expect their values to be close to each other. Since the velocity vector in the middle of this zone is similar to reference scenario, both these indicators can predict the next states in this zone and hence the abnormality values drop below the threshold. Intuitively, $db1$ should indicate if particles are coherent with the predicted region.

Since turning left falls outside the known regions of learned experience, $db1$ generates abnormality values larger than threshold. This value is the largest in the middle of this zone since the largest distance between the known regions from the reference scenario and unknown regions in this scenario occur in this section.

Scenario 3, Turn left, fly up In this scenario, the quad-copter performs two different maneuvers to avoid two obstacles on the opposite sides of the square in the reference scenario. On the side to which the quad-copter reaches first, the quad-copter turns left around the obstacle and on the opposite side, the quad-copter increases height to avoid it. As such, abnormalities are expected to happen between 333 to 1333 and 3000 and 4000 time steps accordingly. Figure 8 illustrates how quad-copters autopilot is requested to behave and Figure 9 illustrates its corresponded GPS recorded values.

Figure 10 illustrates the abnormality values resulted by performing this scenario. Similar to presence of a single deviation (scenario 2) from scenario 1, in this case, presence of two deviations are detected by all three indicators. During flying up maneuver, the autopilot request for rotors’ power increase in advance to reach the highest point at requested time step. As such, abnormality values higher than threshold appear earlier than time step 3000. At time step 3666, the auto pilot, tries to lose height while still keeping some horizontal velocity which results in observing high abnormalities even after time step 4000.

Scenario 4, U-turn In this scenario, the quad-copter is requested to return back to opposite direction when it faces two obstacles at the end of two opposite sides of the square in scenario 1. The requested maneuver is illustrated in Figure 11, and Figure 12 illustrates quad-copter’s recorded GPS data. Making quad-copter’s autopilot to follow such trajectory points, results in expecting abnormal signals higher than threshold values from time steps 1000 to 2000 and 4666 to 5666. As Figure 13 shows, $db2$ which presents the difference between the likelihood behavior and expected prediction and Kalman Innovation raise above the threshold from the beginning of the first interval and remains high until the end of the second interval. This is the result of the opposite velocity in comparison to what has been learned in scenario 1. On the contrary, $db1$, raises only during the two expected time intervals. The reason that $db1$ doesn’t raise above the threshold in regions with opposite velocity in comparison to the learned model

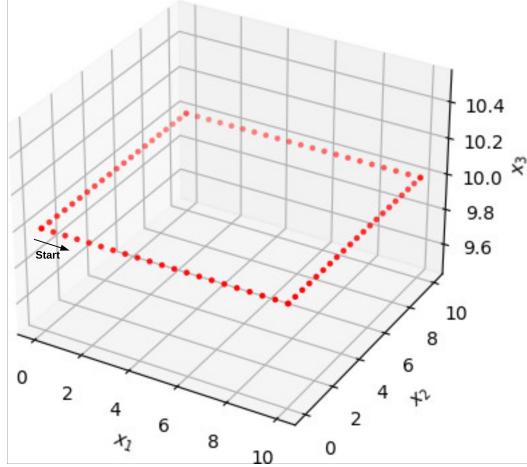


Figure 2: Points for quad-copter's MPC controller to follow scenario 1 trajectory

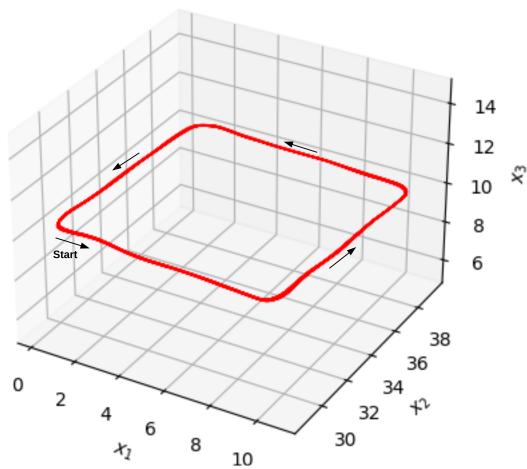


Figure 3: Measured GPS sensor data (observed trajectory) from a quad-copter following the input trajectory with an MPC-based autopilot.

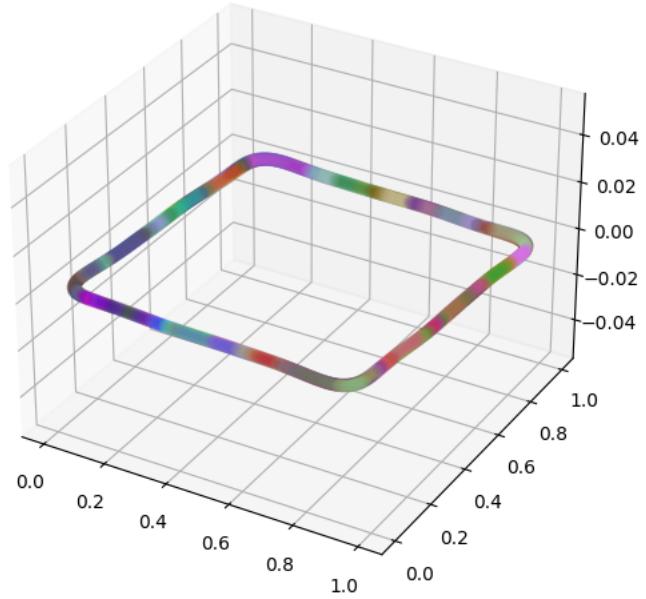


Figure 4: GNG clusters from the data set illustrated in Figure 3

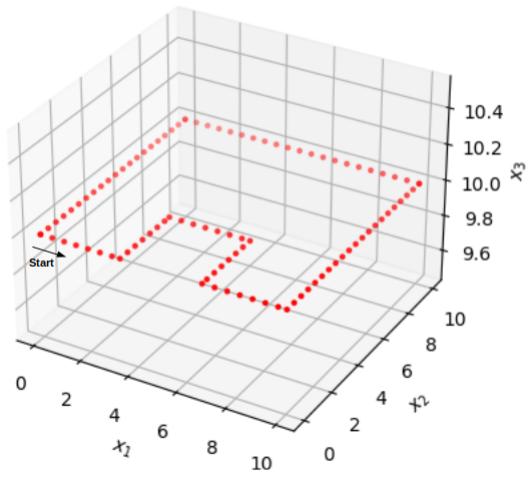


Figure 5: Points for quad-copter's MPC controller to follow scenario 2 trajectory

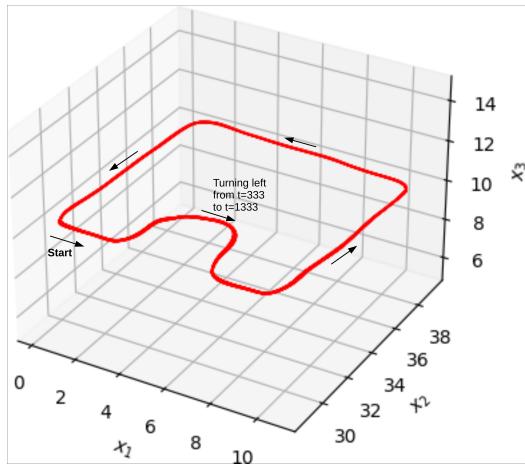


Figure 6: Measured GPS sensor data (observed trajectory) from a quad-copter following the input trajectory from scenario 2 with an MPC-based autopilot.

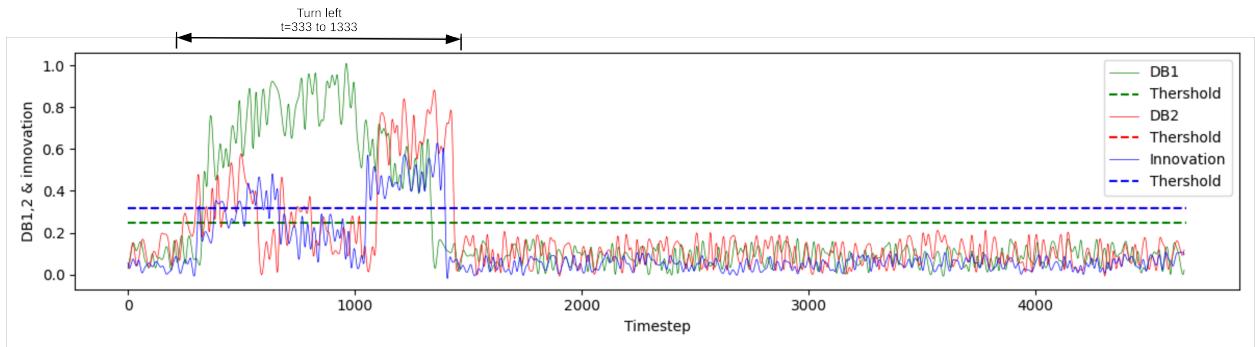


Figure 7: Abnormality values generated from DB1, DB2 and Kalman Innovation while realizing scenario 2 trajectory

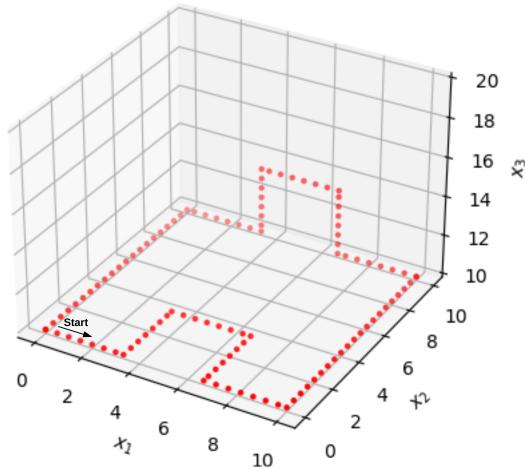


Figure 8: Points for quad-copter's MPC controller to follow scenario 3 trajectory

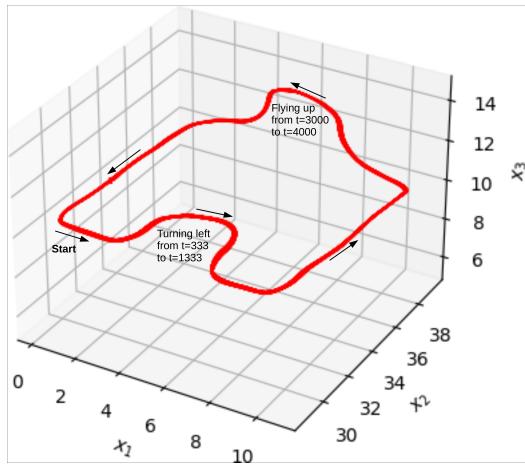


Figure 9: Measured GPS sensor data (observed trajectory) from a quad-copter following the input trajectory from scenario 3 with an MPC-based autopilot.

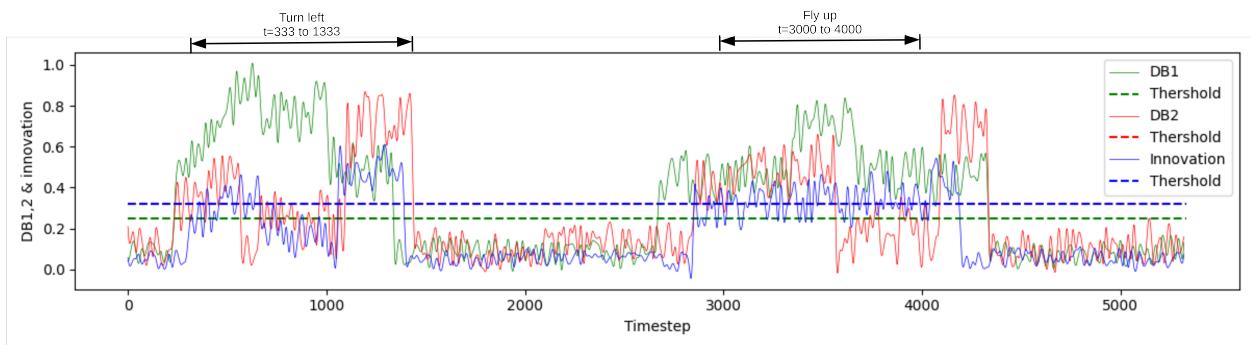


Figure 10: Abnormality values generated from DB1, DB2 and Kalman Innovation while performing scenario 3

in scenario 1 is that the distance between the predicted continuous state and the probability by which the quad-copter falls in a super state is low since it is a region from which the quad-copter has crossed during the reference scenario.

6 Conclusion and future work

In this paper we improved Markov Jump Particle Filter for state estimation and abnormality detection from 4 dimensional state spaces to state spaces of any dimension and we tested it by a moving quad-copter. GNG is used to determine the number of regions in which quasi-static velocities are valid. We proposed these regions as neighborhood with varying radius to limit contribution of nosier measurements in training the MJPF. Furthermore, instead of Euclidean distance, Mahalanobis was used to cluster the measurements and compute the likelihoods. As a future work, we will consider extending the aforementioned improvements to coupling two MJPFs learned from interaction of two quad-copters as two Coupled Hidden Markov Models [33] similar to what is proposed in [34].

References

- [1] A. Morin, “Levels of consciousness and self-awareness: A comparison and integration of various neurocognitive views,” *Consciousness and Cognition*, vol. 15, pp. 358–371, 2006.
- [2] B. Rinner, L. Esterle, J. Simonjan, G. Nebehay, R. Pflugfelder, G. Fernandez, and P. R. Lewis, “Self-Aware and Self-Expressive camera networks,” *Computer*, vol. 48, no. 7, pp. 21–28, 2015.
- [3] A. Winfield, *Robots with Internal Models: A Route to Self-Aware and Hence Safer Robots*, pp. 232–257. 10 2014.
- [4] C. S. Regazzoni, L. Marcenaro, D. Campo, and B. Rinner, “Multisensorial generative and descriptive self-awareness models for autonomous systems,” vol. 108, pp. 987–1010, 2020.
- [5] A. K. Seth, “Interceptive inference, emotion, and the embodied self,” *Trends in cognitive sciences*, p. 565—573, 2013.
- [6] M. Apps and M. Tsakiris, “The free-energy self: A predictive coding account of self-recognition,” *Neuroscience & Biobehavioral Reviews*, vol. 41, pp. 85–97, 2014.
- [7] K. J. Friston, “The free-energy principle: a unified brain theory?,” *Nature Reviews Neuroscience*, vol. 11, pp. 127–138, 2010.
- [8] K. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, 01 2002.
- [9] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni, “Learning switching models for abnormality detection for autonomous driving,” 2018.
- [10] A. Damasio, *Looking for Spinoza : joy, sorrow, and the feeling brain*. Orlando, Fla: Harcourt, 2003.
- [11] A. Damasio, *The feeling of what happens : body and emotion in the making of consciousness*. New York: Harcourt Brace, 1999.
- [12] S. Haykin, “Cognitive dynamic systems: Perception-action cycle, radar and radio,” 2012.
- [13] K. Friston, B. Sengupta, and G. Auletta, “Cognitive dynamics: From attractors to active inference,” *Proceedings of the IEEE*, vol. 102, no. 4, pp. 427–445, 2014.
- [14] Z. Chen, “Bayesian filtering: From kalman filters to particle filters, and beyond,” *Statistics*, vol. 182, 01 2003.
- [15] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, “Nonparametric bayesian learning of switching linear dynamical systems,” in *Advances in Neural Information Processing Systems* (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), vol. 21, Curran Associates, Inc., 2009.
- [16] A. Doucet, N. Gordon, and V. Krishnamurthy, “Particle filters for state estimation of jump markov linear systems,” 2001.
- [17] K. Murphy and S. Russell, *Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks*, pp. 499–515. Springer New York, 2001.
- [18] D. Koller and U. Lerner, *Sampling in Factored Dynamic Systems*, pp. 445–464. Springer New York, 2001.
- [19] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [20] N. Gordon, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, pp. 107–113(6), 1993.

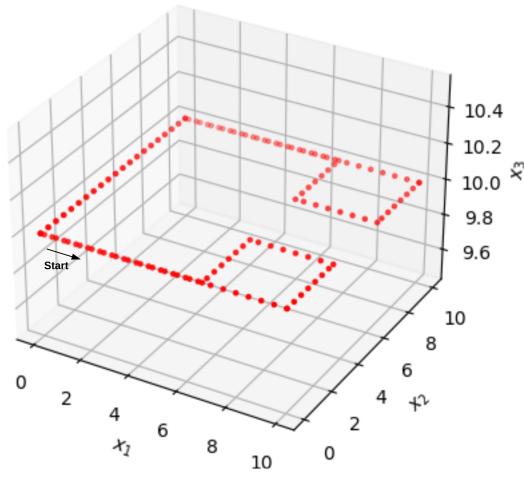


Figure 11: Points for quad-copter's MPC controller to follow scenario 4 trajectory

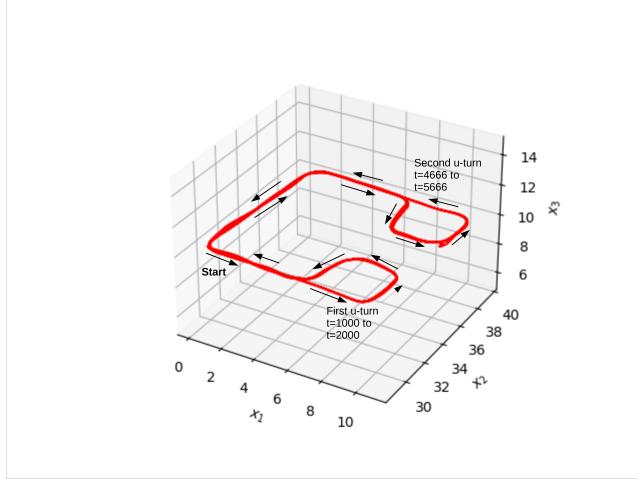


Figure 12: Measured GPS sensor data (observed trajectory) from a quad-copter following the input trajectory from scenario 4 with an MPC-based autopilot.

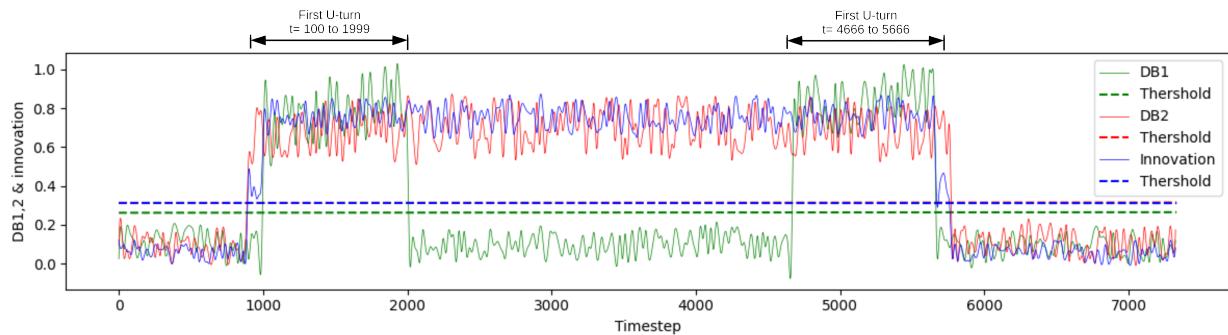


Figure 13: Abnormality values generated from DB1, DB2 and Kalman Innovation while performing scenario 4

- [21] D. Kanapram, F. Patrone, P. Marín-Plaza, M. Marchese, E. L. Bodanese, L. Marcenaro, D. M. Gómez, and C. S. Regazzoni, “Collective awareness for abnormality detection in connected autonomous vehicles,” 2020.
- [22] D. Thekke Kanapram, M. Marchese, E. Bodanese, D. Martín Gómez, L. Marcenaro, and C. Regazzoni, “Dynamic bayesian collective awareness models for a network of ego-things,” *IEEE Internet of Things Journal*, vol. 8, pp. 3224–3241, 02 2021.
- [23] G. Slavic, D. Campo, M. Baydoun, P. Marín, D. Martín Gómez, L. Marcenaro, and C. Regazzoni, “Anomaly detection in video data based on probabilistic latent space models,” 03 2020.
- [24] A. Krayani, M. Baydoun, L. Marcenaro, A. S. Alam, and C. Regazzoni, “Self-learning bayesian generative models for jammer detection in cognitive-uav-radios,” in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–7, 2020.
- [25] B. Fritzke, “A growing neural gas network learns topologies,” *Neural Information Processing Systems*, vol. 7, 03 1995.
- [26] A. Bhattacharyya, “On a measure of divergence between two statistical populations defined by probability distributions,” *Bull. Calcutta Math. Soc*, vol. 35, p. 99–109, 1943.
- [27] T. Báča, M. Petrlík, M. Vrba, V. Spurný, R. Pěnička, D. Hert, and M. Saska, “The mrs uav system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles,” 08 2020.
- [28] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, “Ros: an open-source robot operating system,” vol. 3, 01 2009.
- [29] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2149–2154 vol.3, 2004.
- [30] C. E. Garcia, D. M. Prett, and M. Morari, “Model predictive control: Theory and practice - a survey,” *Autom.*, vol. 25, pp. 335–348, 1989.
- [31] J. Holmström, “Examensarbete dv 3 2002-0830 growing neural gas experiments with gng , gng with utility and supervised gng,” 2002.
- [32] H. Iqbal, D. Campo, M. Baydoun, L. Marcenaro, D. Gomez, and C. Regazzoni, “Clustering optimization for abnormality detection in semi-autonomous systems,” 10 2019.
- [33] I. Rezek, P. Sykacek, and S. Roberts, “Learning interaction dynamics with couple hidden markov models,” *Science, Measurement and Technology, IEE Proceedings -*, vol. 147, pp. 345 – 350, 12 2000.
- [34] M. Baydoun, D. Campo, D. Kanapram, L. Marcenaro, and C. S. Regazzoni, “Prediction of multi-target dynamics using discrete descriptors: an interactive approach,” 2019.