Autonomous Navigation for Drone Swarms in GPS-Denied Environments Using Structured Learning

William Power(✉), Martin Pavlovski, Daniel Saranovic, Ivan Stojkovic, and Zoran Obradovic

Center for Data Analytics and Biomedical Informatics, Temple University, Philadelphia, PA, USA
{william.power,martin.pavlovski,daniel.saranovic, ivan.stojkovic,zoran.obradovic}@temple.edu

Abstract. Drone swarms are becoming a new tool for many tasks including surveillance, search, rescue, construction, and defense related activities. As their usage increases, so does the possibility of adversarial attacks on their contribution to these use cases. One possible avenue, whether deliberate or not, is to deny access to the position feedback offered by the Global Positioning System (GPS). Operating in these 'GPS denied' environments poses a new challenge; both in navigation, and in collision avoidance. This study proposes two novel concepts; a structural model of environmental deviance to aid in autonomous navigation, and a method to use the output of said model to implement a collision avoidance system. Both of these concepts are developed and tested in the framework of a simulated environment that mimics a GPS-denied scenario. Using data from hundreds of simulated swarm flights, this work shows structured learning can improve navigational accuracy without the need for externally provided position feedback.

Keywords: Machine learning · Adaptive control · Swarm intelligence · Decision-making

1 Introduction

Drone swarms are collections of remotely or autonomously controlled aerial vehicles which maintain some form of internal structure between the individuals. These swarms have utility in many domains and tasks; surveillance (coverage), search and rescue, construction, and defense related activities [9,16]. Swarms typically rely on some kind of position feedback to maintain the prescribed structure of the swarm. Referred to as localization, this task is typically achieved by

Published by Springer Nature Switzerland AG 2020
I. Maglogiannis et al. (Eds.): AIAI 2020, IFIP AICT 584, pp. 219–231, 2020.
https://doi.org/10.1007/978-3-030-49186-4_19

utilizing GPS data and other sensor modalities [10,20]. Without position feedback, maintaining a safe, collision-free flight within a crowded airspace becomes an increasingly difficult task [1,17].

In domains where adversaries or technical issues might disable or impede the efficacy of position feedback, drone swarms are at risk of losing the position information prerequisites that allow for safe, collision-free flight. In these GPS-denied environments, other methods must be used to provide position information to the drones, or to provide other navigation schemes that allow for safe flight. This paper seeks to investigate a method for coping with a GPS-denied environment without incurring these systemic costs of complex sensor fusion and navigational calculation. Given a set of drones with the most basic of intra-drone communication, and with a limited computational budget, are there still navigation methods that can cope with the loss of GPS provided position feedback? One of the simplest methods for dealing with the lack direct positional feedback operate is the technique of dead reckoning. In such an approach, the current estimate of a drones position is taken as the 'true' position, and new control signals are calculated based on the relative location of this position to a desired position.

Applying the method of dead reckoning to a drone swarm introduces a new source of error that might lead to collisions of drones within a swarm due to environmental deviations. Dead reckoning assumes the position estimate at the prior time-step is accurate, and only influenced by the action of the drone. It lacks a model of the influences of the environment. However, additional sensors, if available in the GPS-denied environment, provide intra-drone communication and distance keeping. Radio frequency sensors, or simple optical systems may provide such feedback [7,12]. While this is useful for collision avoidance, ultimately, they cannot also solve the navigation issue. This work assumes the existence of a simple inter-drone communication channel. This can be used to craft a nearest neighbor graph of the swarm. The history of drone locations over time, coupled with this graph, provides a spatio-temporal data-set.

This work hypothesizes that such a spatio-temporal data-set can be used to train a model which will improve upon the efficacy of dead reckoning in GPS-denied environments by predicting future environmental deviations based on changes in the structure of the swarm network. We hypothesize that such a network encodes some information about the environmental deviations that introduce the accumulated errors of dead reckoning. By using appropriate structural machine learning tools, this information could be leveraged to correct the estimated position of each drone within the swarm. In turn, this will reduce the accumulation of error, and lead to position estimates that are closer to their true value when the model is used versus pure dead reckoning.

The novelty of the proposed procedure is that it provides a framework to safely navigate a swarm of drones in a GPS-denied environment with a purposefully simplified set of sensor modalities and computational capabilities. This framework is shown to provide course maintenance and collision avoidance via structural learning of environmental deviance's in an online manner.

To support this, in this study we answer the following research questions: *1) Can a prediction-based approach properly estimate a drones trajectory after communication loss? 2) If so, do structure-based models improve this prediction capability? 3) Can a method for avoiding collisions between swarms be implemented, using the variance of the structured model as input?*

2 Prior Work

The past decade has seen substantial work on the issues of collision avoidance, localization, and mapping [6]. The prevalence of these swarms has incentivized the creation of high fidelity models of both individual drones [8] and of swarms of these individuals [2,18].

Due to their prevalence, the security and consistency of communication within a swarm becomes paramount [3]. The manner of communication that enables the localization of drones within the swarms becomes an attack surface. In systems where GPS is the communication channel for localization, entering a 'GPS-denied' environment poses a threat to the swarm. Work has been done to produce systems that leverage visual input to accommodate localization in such settings. On board visual sensors have been utilized to localize drones with respect to observations of visual indicators applied to each drone [14]. The bearing of the swarm were combined with observed landmarks to implement a sufficient simultaneous localization and mapping (SLAM) framework. It is also shown that other modalities can be utilized [15]. Radio frequency ranging has been investigate. This can provide a means of inter-drone localization within a swarm by observing the received signal strength of RF signals from neighboring drones [12].

When position feedback is lost, the technique of dead reckoning can be used. This manner of control simply assumes the current position estimate of the drone location is correct, and flies accordingly. Such a system is prone to quick accumulation of error when used in isolation. However, work has been done to improve upon these estimates. The combination of dead reckoning, and observations using a Kalman filter can still allow for improved trajectory tracking [21]. Dead reckoning can also be combined with other localization methods to improve its accuracy. Such a method is proposed in which the received signal strength of neighboring drone broadcasts is combined with a dead reckoning scheme to enable accurate localization [19].

3 Method

3.1 Problem Statement

Assume a swarm of D drones that is heading towards a certain destination in a stochastic environment. Given that global communication (e.g., GPS communication) is lost on the way towards the destination, the objective is to predict the positions of all drones at all subsequent time-steps.

Formally, a swarm $\mathcal{S}^{[T]} = \{\mathcal{D}_1^{[T]}, \ldots, \mathcal{D}_D^{[T]}\}$ is defined as a collection of D individual drones, where $\mathcal{D}_i^{[T]} = \{\mathbf{p}_i^1, \ldots, \mathbf{p}_i^T\}$ represents the trajectory of the i-th drone containing its positions $\mathbf{p}_i^t \in \mathbb{R}^K$ for each time-step $t = 1, \ldots, T$. Over the course of a swarm's flight during which *global communication is lost* at a certain time-step t_c, the *objective* is to learn a swarm-level function $F_S = (f_1, \ldots, f_D) : \mathbb{R}^{D \times \omega K} \to \mathbb{R}^{D \times K}$. Each $f_i : \mathbb{R}^{\omega K} \to \mathbb{R}^K$ is a drone-level function that, when at any time-step $t > t_c$, *predicts* the position of the i-th drone at the next time-step $\mathbf{p}_i^{t+1}$ based on a ω-sized *window* of past positions $\boldsymbol{\omega}_i^t = [\mathbf{p}_i^{t-\omega+1}, \ldots, \mathbf{p}_i^t]$.

The above statement assumes that none of the individual drones is aware of its true position upon communication loss since any position feedback will be unavailable (e.g., after a GPS-denied regime was initiated). Nevertheless, an intra-swarm communication method can be leveraged to allow the drones to share their true and predicted positions before and after communication loss, respectively. This assumption is supported by the fact that low cost and complexity methods are available for enabling such communication [7,12]. In addition, the task of distributing this information falls within a well researched field [4,5], which is out of the scope of this paper.

3.2 Structured Learning for Trajectory Prediction

Given a swarm $S^{[t_c]}$ that lost global communication at time-step t_c, we propose to use a Multi-Target Gaussian Conditional Random Field (MT-GCRF) model to predict the drones' positions at each subsequent time-step $t > t_c$ by leveraging the available positional data from all time-steps prior to t_c. To that end, an input matrix $\mathbf{W} = [\mathbf{W}^1, \ldots, \mathbf{W}^{t_c-\omega}]^\top$ is constructed for each previous time-step t. For a specific time-step t, $\mathbf{W}^t$ contains the ω-sized windows of all D drones before t, i.e. $\mathbf{W}^t = [\boldsymbol{\omega}_1^{t+\omega-1}, \ldots, \boldsymbol{\omega}_D^{t+\omega-1}]^\top$. Similarly, a target matrix $\mathbf{P} = [\mathbf{P}^1, \ldots, \mathbf{P}^{t_c-\omega}]^\top$ is obtained, where $\mathbf{P}^t = [\mathbf{p}_1^{t+\omega}, \ldots, \mathbf{p}_D^{t+\omega}]^\top$ contains the positions of all D drones at time-step t. Moreover, to capture the between-target similarities, a similarity matrix $\mathbf{S}$ was additionally constructed as $S_{ij}^t = 1 - \frac{d_{ij}^t}{\max_{i,j} d_{ij}^t}$, where d_{ij}^t is the Euclidean distance between $\mathbf{p}_i^t$ and $\mathbf{p}_j^t$. The resulting $\mathbf{S}^1, \ldots, \mathbf{S}^{t_c-\omega}$ are diagonally placed in a single supra matrix $\mathbf{S}$.

The input and target matrices $\mathbf{W}$ and $\mathbf{P}$, along with the similarity matrix $\mathbf{S}$, are then used to learn the parameters of an MT-GCRF model by maximizing the probability of the drones' target positions conditioned on their corresponding windows of previous points, $P(\mathbf{P}|\mathbf{W})$. Finally, the learned MT-GCRF model is used to estimate the positions for all drones at each time-step following t_c.

MT-GCRF Overview. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\top$ denote an input matrix of explanatory variables and a target vector of response variables, respectively, such that $\mathbf{y}_i \in \mathbb{R}^K$ is associated with $\mathbf{x}_i \in \mathbb{R}^M$, for each

$i = 1, \ldots, N$. A conventional Continuous Conditional Random Field [11] models the conditional probability over all target vectors $\mathbf{Y}$ given $\mathbf{X}$ as

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z} \exp\left\{ -\sum_{k=1}^{K} \left[\alpha \sum_{i=1}^{N} (y_{i,k} - f_k(\mathbf{x}_i))^2 + \beta \sum_{i \sim j} S_{ij} (y_{i,k} - y_{j,k})^2 \right] \right\}, \tag{1}$$

where $[\alpha, \beta]$ are the model parameters, f_k is a parameterized function that outputs an estimate of the k-th dimension of $\mathbf{y}_i$ given the i-th input vector, and Z is a normalization constant. The first term in the exponent is the association potential that models the pairwise relationship between the k-th target of $\mathbf{y}_i$ and its estimate. In the seconds term, S_{ij} describes the similarity between $\mathbf{y}_i$ and $\mathbf{y}_j$. The entire similarity matrix $\mathbf{S} = [S_{ij}]_{N \times N}$ can be thought of as an adjacency matrix of a weighted similarity graph. The structure of such a graph reflects the similarities among its nodes.

MT-GCRF Learning. To allow for efficient parameter learning, the conditional probability in Eq. 1 can be transformed to a Multivariate Gaussian form [13]:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{(2\pi)^{N/2} |\mathbf{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2} (\mathbf{Y} - \boldsymbol{\mu})^\top \mathbf{\Sigma} (\mathbf{Y} - \boldsymbol{\mu}) \right\}. \tag{2}$$

The above Gaussian-form model is referred to as a Multi-Target Conditional Random Field (MT-GCRF). The parameters of MT-GCRF are determined by maximizing the conditional likelihood of Eq. (2).

MT-GCRF Inference. Given an input matrix $\mathbf{X} \in \mathbb{R}^{J \times M}$, inference is carried out by calculating $\boldsymbol{\mu} = \alpha \mathbf{\Sigma} [\mathbf{b}_1, \ldots, \mathbf{b}_J]^\top$, where $\mathbf{b}_i = [f_1(\mathbf{x}_i), \ldots, f_K(\mathbf{x}_i)]^\top$.

3.3 Variance Estimation and Collision Avoidance

To address the third research question, a method for swarm-based collision avoidance was developed. “Swarm-based” refers to the fact that this is a procedure that allows a set of swarms to model each others environmental deviation. In the proposed approach each swarm will manage and train its own structure-based model. Then, recalling the assumption of an existing means of distributed communication, we extend the assumption to imply swarms will be close enough to share models.

Simply sharing the models does not provide a collision avoidance system. To leverage the models in creation of one, this work chooses to interpret the uncertainty of predicted future locations as output variance of the structured model, and in turn uses this to set the radius of sphere representing the ‘possible collision’ volume for the swarm. Since the swarms will have a copy of each others’ models, this radius can be deduced for each swarm. More specifically, we take the mean of the predicted positions of all drones in a swarm to be the swarm’s estimated ‘center’. Consequently, we define a swarm’s variance as the distance of the swarm’s center to the predicted position of the furthest drone, plus the maximum per-dimension variance of that drone.

During inference, each swarm uses the shared models to determine if their radius intersects that of another swarm since in such a situation a possible collision might occur soon. To avoid collision, the swarms update the target positions of their control systems such that their revised targets are at positions opposite one-another along the line connecting the centers of the swarms variance radius. The trajectories update distance along this line is set to keep the radii of the possibly colliding swarms from intersecting while simultaneously minimizing change from the intended mission plan. This is done by setting the distance from the center points along the connecting line for each swarm to be half the value of the overlap of the swarm radii. Once these new safe positions have been reached, the swarms reset their targets to the original location.

4 Experiments

We have conducted two sets of experiments. The first set is aimed to characterize the efficacy of the model, as measured by its ability to correct the errors induced by environmental deviations when communication is lost and the swarm utilizes dead reckoning without position feedback. In these experiments, the distance from a ground truth flight path (where no feedback is lost) is used to compare the errors from pure dead reckoning, dead reckoning corrected by an unstructured model, and dead reckoning corrected by the proposed structural MT-GCRF model. Parameters of the models were learned from simulated training data gathered while drones locations were available and the prediction were utilized to augment location estimates in a GPS-denied environment. In the second set of experiments, the qualitative efficacy of the proposed collision avoidance mechanism is tested.

4.1 Drone-Level Trajectory Prediction

To compare the efficacy of dead reckoning (DR) with no model, an unstructured regression model (UR), and a structured model (MT-GCRF in this case); four sets of data were collected for each simulated flight path. Each of these data sets is collected from a simulation that shares the same set of randomly generated gusts. This is accomplished by managing the random state for the simulations, so that they are all initialized with the same random seed.

The first simulation for each ‘run’ of the experiment is the baseline, or ground truth run. In this, no communication loss occurs, and the swarm proceeds in navigating with full position feedback for the entire flight path. This provides the basis of comparison for the three methods.

The second, third, and fourth simulations make use of dead reckoning, the unstructured model UR, and the structured model MT-GCRF, respectively. In the case of DR, once communication is lost the navigation of the drone swarm becomes based on an estimate of the current position. Without feedback, this estimate is updated by adding the current instantaneous velocity of each drone to its previous position estimate. In the third simulation, the drones’ positions after communication loss are predicted by UR, which are later used by MT-GCRF. In a sense, this is as if we set the β parameter of MT-GCRF to 0. In the fourth simulation, the full MT-GCRF model is used, which learns the relevance of the UR's outputs and the swarm's structure, represented by α and β, respectively.

In both sets of trajectory prediction experiments, the models were ran on simulated flight paths consisting of 150 time steps, with communication loss occurring at the 100-th timestep; meaning that 100 timestep were used for training, while the remaining 50 were used to evaluate the models. All prediction-based models used a ω-sized window, with ω being 10, to construct their training data. The UR model (and therefore the underlying model in MT-GCRF) was a Multi-Target Neural Network with a single hidden layer containing 30 hidden units, trained using the L-BFGS optimization method.

The root mean squared error (RMSE) between the true and predicted drones' positions, along a certain dimension, was used. For a swarm $\mathcal{S}$ of D drones, over a flight-path of T time-steps, the error along the k dimension is computed as $RMSE(\mathcal{S}, k) = \frac{1}{T-t_c}\sum_{t=t_c+1}^{T}\sqrt{\frac{1}{D}\sum_{i}^{D}(p_{i,k}^{t} - \hat{p}_{i,k}^{t})^2}$ where $\hat{p}_{i,k}^{t}$ is the predicted position of the i-th drone at timestep t after and $p_{i,k}^{t}$ is its true position at timestep t.

This set of experiments was split into two groups. One to investigate the lower-complexity swarms, and another to investigate higher-complexity swarm topologies. This was done to aid in highlighting the effect of structure in the efficacy of the prediction. More complex layouts provide more interesting structure, and in turn show a different response to the structured learning method.

Experiments for Lower-Complexity Swarms. In the lower-complexity swarms, the layouts was a four-drone planar swarm, and an eight-drone cubic swarm. In each case, the swarms start flights at one corner of a volume, and move towards the diagonal corner, with a set critical time in the center of the flight path. Each flight had at least one gust present in the training and inference stages. A set of 100 random simulations were conducted, collecting the previously described data.

Experiments for Higher-Complexity Swarms. The higher-complexity swarm experiments were conducted in the same manner as the lower-complexity swarm experiments, but using nine-drone planar, and 27-drone cubic layouts. This is of particular interest because these configurations contain 'internal' drones, whereas the low complexity configurations do not. This is hypothesized to increase the influence of the structure (the similarity matrix S) within the structural model, improving its efficacy over the unstructured model when compared to the lower complexity layouts.

4.2 Multi Swarm Collision Avoidance Experiments

To explore the utility of the proposed collision avoidance procedure, a simulated mission was constructed that places two swarms at opposite corners of a volume, with target flight paths that intersect. The critical time for GPS denial was set prior to the ostensible collision time of the swarms. Each swarm uses a separate model that will be shared with the other swarm. Each swarm's model was bagged five times using bootstrap aggregation to obtain its uncertainty (prediction variance). All other experimental settings remain the same as those described in Sect. 4.1. The resulting behaviour of the swarms is gathered from observation of animations of the flights, as well as from calculating the number (if any) of collisions.

4.3 Simulation Design

To test the hypothesis that a structural model can aid in navigation, a simulator must be able to model the behaviour of the drones in a deterministically reproducible way, under a reproducible set of environmental deviations. This must be done while tracking the spatio-temporal data set encoded in the nearest-neighbor graph of the drones. To accomplish this, a simulator was written in Python 3.6.9.

To simulate the use of GPS, and the subsequent loss of GPS feedback, the program tracks an actual and estimated position for each member of the swarm. When in a GPS-enabled regime, the actual position is used, while in a GPS-denied regime the estimated position is used.

The simulation introduces environmental deviation by use of a simple gusting wind model. A gust of wind is represented as a single vector of a random magnitude, pointing a randomly selected direction on the plane. For the duration of the gust (also selected from a random distribution), this vector is used to move the drone position within the simulated environment.

The wind deviation vector is not applied to each drones position identically. To introduce more realistic behaviour, and to partially encode some information about the deviation within the network structure, a model of 'drafting' was introduced. In this model, the relative location of each drone to others, along the axis in which the gust is pointing, will decrease the magnitude of the deviation applied to that drone. That is, drones 'behind' other drones along this gust vector will feel less deviation. To achieve this, a convex hull is calculated, and used to quickly determine the influence of drones on the hull onto those inside, with respect to the wind direction. This process is done recursively until the drafting effect of each drone on each other is calculated, yielding a scaled wind deviation for each drone.

5 Results

5.1 Drone-Level Trajectory Prediction

For the lower and higher complexity trajectory prediction tasks described in Sect. 4.1, the error is reported using the RMSE measure. The behaviour typical of all models in a single run is illustrated in Fig. 1. The unstructured model has a decreased stability compared to the structured model (2 of the 4 drones have erratic flight paths). When the structured model is used, the similarity matrix, and the learned association potentials, can effectively 'smooth out' the impact of the instability of the underlying regressors.

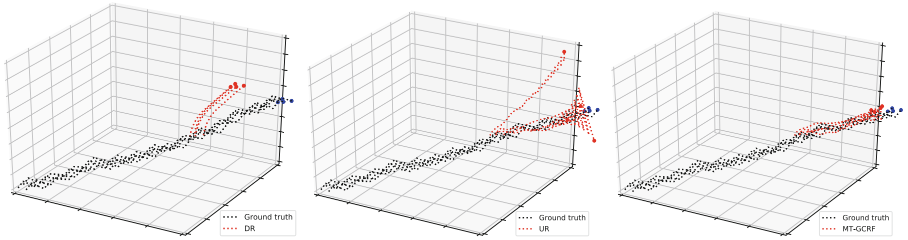

Fig. 1. Example of a simulated flight trajectories of a 4 drone planar swarm. Subfigures contain the ground truth trajectory and the predicted trajectories by DR (left), UR (middle), and MT-GCRF (right) upon communication loss.

Table 1. Trajectory prediction for lower-complexity swarms (4 and 8 drones) by DR, UR and MT-GCRF. RMSE and its 90% confidence interval in three dimensions (x, y, z) is reported for 100 random flights over 50 inference timesteps in a GPS-denied environment.

Model	RMSE		
	x	y	z
4 Drone Planar			
DR	2.4290 ± 0.0298	1.4102 ± 0.0141	0.3813 ± 0.0230
UR	0.5854 ± 0.0677	0.3095 ± 0.0333	0.3327 ± 0.0426
MT-GCRF	**0.4280** ± 0.0447	**0.2531** ± 0.0233	**0.2264** ± 0.0292
8 Drone Cubic			
DR	2.4204 ± 0.0265	1.4094 ± 0.0141	0.3401 ± 0.0224
UR	0.5994 ± 0.0713	0.3155 ± 0.0210	0.3582 ± 0.0264
MT-GCRF	**0.4228** ± 0.0374	**0.2395** ± 0.0130	**0.2875** ± 0.0198

Table 2. Trajectory prediction for higher-complexity swarms (9 and 27 drones) by DR, UR and MT-GCRF. RMSE and its 90% confidence interval in three dimensions (x, y, z) is reported for 100 random flights over 50 inference timesteps in a GPS-denied environment.

Model	RMSE		
	x	y	z
9 Drone Planar			
DR	2.3408 ± 0.0245	1.3720 ± 0.0145	0.4002 ± 0.0239
UR	0.7208 ± 0.1019	0.3826 ± 0.0480	0.4850 ± 0.0709
MT-GCRF	**0.5171** ± 0.0628	**0.3476** ± 0.0370	**0.2753** ± 0.0449
27 Drone Cubic			
DR	2.2934 ± 0.0224	1.3701 ± 0.0127	**0.3748** ± 0.0212
UR	0.6932 ± 0.0591	0.4127 ± 0.0385	0.4810 ± 0.0425
MT-GCRF	**0.5014** ± 0.0282	**0.3637** ± 0.0160	0.3965 ± 0.0238

Lower-Complexity Swarms. Results of repeating trajectory prediction experiments 100 times for the same flight plan of 4 and 8 drones and a variety of environmental conditions are summarized in Table 1. In this, we can see that both MT-GCRF and UR outperformed pure DR. Additionally, the structured model MT-GCRF outperformed the unstructured model UR.

Higher-Complexity Swarms. Trajectory prediction experiments were also repeated 100 times for more complex swarms of 9 and 27 drones. Results are summarized in Table 2. Similar to the low-complexity experiments, the structured model (MT-GCRF) outperforms the unstructured regressors (UR), which in turn outperforms dead reckoning (DR), in most instances.

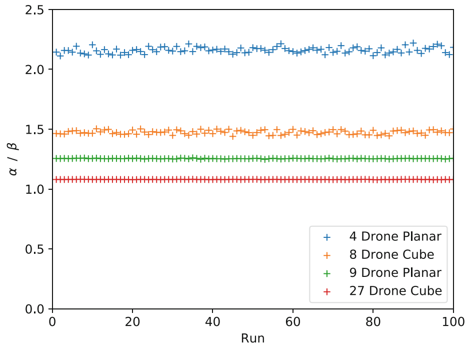

Fig. 2. The ratio of α to β learned for each swarm size and shape, across all runs. Recall that α and β represent the importances of the unstructured predictor and the swarm structure, respectively (refer to Eq. (1) for more details).

Parameter Analysis. The parameters learned by MT-GCRF, over all 100 runs, for both low and higher-complexity swarms are presented in Fig. 2. There are two main insights that the figure suggests. First, the α/β ratio decreases for more complex swarms, suggesting that the larger/complex a swarm is, the more relevant the structure is to the model (i.e. the structural term in (1) has a higher relative value). Moreover, the ratio of the structural terms exhibits a higher stability over multiple runs for larger/complex swarms (i.e. MT-GCRF is more stable w.r.t. its parameters).

5.2 Multi Swarm Collision Avoidance

In the trajectory prediction experiments it was observed that MT-GCRF was more accurate than its alternatives, therefore, MT-GCRF was selected as the prediction model for each of two swarms. Each swarm's MT-GCRF was ran on 150-timestep flight path, with t_c set at the 100-th timestep.

To investigate the efficacy of the avoidance procedure, 100 such simulations were performed and 0 collisions were observed. As a reference, the states of a two-swarm collision procedure, from a single simulation, before, during, and after collision avoidance was triggered, are illustrated in Fig. 3.

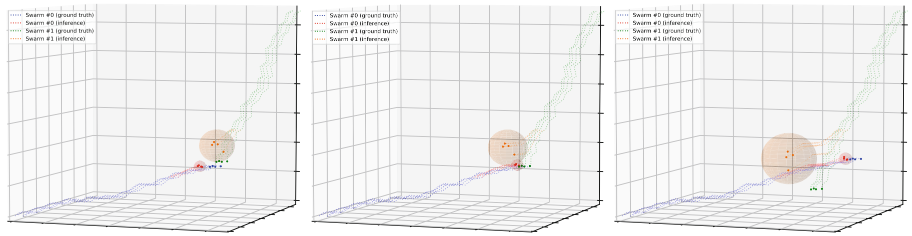

Fig. 3. Multi-swarm collision avoidance. Positions of each swarm's drones as well as the swarm's estimated variances are displayed for a time-step: Left: Prior to triggering the collision avoidance mechanism. Middle: During collision avoidance. Right: After the collision avoidance mechanism has terminated.

6 Conclusions

The results of our study provide evidence that the prediction-based approaches can sufficiently model deviations in drone swarms trajectories after communication loss without using additional sensors. In hundreds of conducted experiments both unstructured and structured predictive models showed significantly improved swarm-level RMSE over dead reckoning when compared to the ground truth. Spatio-temporal patterns of environmental deviations observed before GPS signal was lost were exploited the most effectively by the proposed MT-GCRF structured learning method for autonomous navigation in both the lower and higher complexity swarm configurations. Finally, the proposed structured regression based autonomous navigation method was able to be leveraged to form a collision avoidance framework that successfully managed and avoided a collision between two swarms in GPS-denied environment.

Acknowledgements. This research was supported in part by the NSF grants IIS-1842183, SES-1659998 and AFRL Contract No. FA8750-18-C-0041, subcontact 555027-78056.

References

1. Barbeau, M., Garcia-Alfaro, J., Kranakis, E.: Geocaching-inspired resilient path planning for drone swarms. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE INFOCOM 2019, Paris, France, pp. 620–625. IEEE, April 2019. https://doi.org/10.1109/INFCOMW.2019.8845318
2. Belkhouche, F., Vadhva, S., Vaziri, M.: Modeling and controlling 3D formations and flocking behavior of unmanned air vehicles. In: 2011 IEEE International Conference on Information Reuse & Integration, Las Vegas, NV, USA, pp. 449–454. IEEE, August 2011. https://doi.org/10.1109/IRI.2011.6009590
3. Brust, M.R., Danoy, G., Bouvry, P., Gashi, D., Pathak, H., Gonçalves, M.P.: Defending against intrusion of malicious UAVs with networked UAV defense swarms. In: 2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops), pp. 103–111, October 2017. https://doi.org/10.1109/LCN.Workshops.2017.71. http://arxiv.org/abs/1808.06900

4. Campion, M., Ranganathan, P., Faruque, S.: UAV swarm communication and control architectures: a review. J. Unmanned Veh. Syst. **7**(2), 93–106 (2018)
5. Chmaj, G., Selvaraj, H.: Distributed processing applications for UAV/drones: a survey. In: Selvaraj, H., Zydek, D., Chmaj, G. (eds.) Progress in Systems Engineering. AISC, vol. 366, pp. 449–454. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-08422-0_66
6. Chung, S.J., Paranjape, A.A., Dames, P., Shen, S., Kumar, V.: A survey on aerial swarm robotics. IEEE Trans. Robot. **34**(4), 837–855 (2018). https://doi.org/10.1109/TRO.2018.2857475. https://ieeexplore.ieee.org/document/8424838/
7. Faigl, J., Krajník, T., Chudoba, J., Přeučil, L., Saska, M.: Low-cost embedded system for relative localization in robotic swarms. In: 2013 IEEE International Conference on Robotics and Automation, pp. 993–998. IEEE (2013)
8. Fernando, H.C.T.E., De Silva, A.T.A., De Zoysa, M.D.C., Dilshan, K.A.D.C., Munasinghe, S.R.: Modelling, simulation and implementation of a quadrotor UAV. In: 2013 IEEE 8th International Conference on Industrial and Information Systems, Peradeniya, Sri Lanka, pp. 207–212. IEEE, December 2013. https://doi.org/10.1109/ICIInfS.2013.6731982. http://ieeexplore.ieee.org/document/6731982/
9. Hayat, S., Yanmaz, E., Muzaffar, R.: Survey on unmanned aerial vehicle networks for civil applications: a communications viewpoint. IEEE Commun. Surv. Tutor. **18**(4), 2624–2661 (2016). https://doi.org/10.1109/COMST.2016.2560343
10. Kokić, P., Tomaš, B.: Enhanced drone swarm localization using GPS and trilateration based on RF propagation model. In: Central European Conference on Information and Intelligent Systems, pp. 259–264. Faculty of Organization and Informatics Varazdin (2017)
11. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data (2001)
12. Lockspeiser, J.R., Don, M.L., Hamaoui, M.: Radio frequency ranging for swarm relative localization. Technical ARL-TR-8194, US Army Research Laboratory (2017)
13. Radosavljevic, V., Vucetic, S., Obradovic, Z.: Continuous conditional random fields for regression in remote sensing. In: ECAI, pp. 809–814 (2010)
14. Saska, M., et al.: System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization. Auton. Robots **41**(4), 919–944 (2016). https://doi.org/10.1007/s10514-016-9567-z
15. Singh, S., Sujit, P.: Landmarks based path planning for UAVs in GPS-denied areas. IFAC-PapersOnLine **49**(1), 396–400 (2016). https://doi.org/10.1016/j.ifacol.2016.03.086
16. Tahir, A., Böling, J., Haghbayan, M.H., Toivonen, H.T., Plosila, J.: Swarms of unmanned aerial vehicles – a survey. J. Ind. Inf. Integr. **16**, 100106 (2019). https://doi.org/10.1016/j.jii.2019.100106
17. Wang, C.-L., Wang, T.-M., Liang, J.-H., Zhang, Y.-C., Zhou, Y.: Bearing-only visual SLAM for small unmanned aerial vehicles in GPS-denied environments. Int. J. Autom. Comput. **10**(5), 387–396 (2013). https://doi.org/10.1007/s11633-013-0735-8
18. Watson, N., John, N., Crowther, W.: Simulation of unmanned air vehicle flocking. In: Proceedings of Theory and Practice of Computer Graphics, Birmingham, UK, pp. 130–137. IEEE Computer Society (2003). https://doi.org/10.1109/TPCG.2003.1206940. http://ieeexplore.ieee.org/document/1206940/
19. Wu, H., Qu, S., Xu, D., Chen, C.: Precise localization and formation control of swarm robots via wireless sensor networks. Math. Prob. Eng. **2014**, 1–12 (2014). https://doi.org/10.1155/2014/942306

20. Yanmaz, E., Yahyanejad, S., Rinner, B., Hellwagner, H., Bettstetter, C.: Drone networks: communications, coordination, and sensing. Ad Hoc Netw. **68**, 1–15 (2018). https://doi.org/10.1016/j.adhoc.2017.09.001
21. Zhou, Q.L., Zhang, Y., Qu, Y.H., Rabbath, C.A.: Dead reckoning and Kalman filter design for trajectory tracking of a quadrotor UAV. In: Proceedings of 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, QingDao, China, pp. 119–124. IEEE, July 2010. https://doi.org/10.1109/MESA.2010.5552088