

PCA-based Maximum Correntropy Kalman Filter Application for Agricultural Unmanned Aerial Vehicle

Fethi Candan^{1,2,*} and Johnny Li^{1,*}

Abstract—This paper addresses challenges in agricultural unmanned aerial vehicle (A-UAV) positioning, emphasizing the significance of accurate position estimation for applications like coverage path planning under depended noises. The study introduces a solution involving a PCA-based maximum correntropy Kalman filter (PCA-MCKF) to mitigate issues such as low-altitude flight control, inaccurate position estimation due to coloured noise, and non-Gaussian distribution, including wind effects. Comparative analysis with traditional methods, such as Kalman filter (KF), PCA-KF, and PCA-MCKF, is conducted using four rotor-wing UAVs with linear and nonlinear dynamical models. The paper employs interval type-2 Fuzzy PID as an intelligent controller method and constant acceleration and constant velocity manoeuvre models for estimation. Root mean square error is used as the accuracy metric, and real-time simulations in Webots demonstrate the superiority of the proposed PCA-MCKF in enhancing agricultural UAV applications.

I. INTRODUCTION

Smart agriculture is a potential solution to the issues that negatively impact farming and make some cropland useless. Unmanned Aerial Vehicles (UAVs) are beneficial for smart agriculture. UAVs are low-cost, modifiable equipment solutions and manoeuvre capability, and they can control teleoperators or autonomous based. They have many advantages; their battery capacities and weights may be a problem. Considering the performances of the UAVs, the Time of Flight (ToF) for UAVs is below 40 minutes. For example, if a farming area is more than 100 acres, the UAV may fly multiple times, or a UAV swarm may be another solution.

Generally, UAVs have been used for inspection, crop detection, treating crop health, and spreading pesticides. Even with human-based control or autonomous control, getting high accuracy/precision in agriculture has remained a problem [2], [13]. The accuracy/precision is directly related to position and localization; on the other hand, Global Positioning System (GNSS/GPS) and Inertial Navigation System (INS) play essential roles at this stage. The example of target tracking with radar illustrates how even if measurement noise is independent of an object's motion,

the sampling process introduces additional noise, creating dependence [11]. The discussion emphasises the impact of down-sampling dynamics in filtering problems and points to its relevance in various practical applications, as discussed in the referenced literature.

In literature, in ref [12], it has been presented a novel quaternion-based attitude estimation algorithm aiming to enhance accuracy, numerical stability, and computational efficiency. In the research, there are some contributions and one of them is directly related to extending state-dependent noise properties based on Choukroun et al.'s work and deriving comprehensive expressions for covariance matrices inspired by a generalized approach outlined in [7]. In ref [15], it has been introduced a significant contribution: a generalized extended Kalman filter designed for estimating the state of discrete-time dynamical processes affected by state-dependent measurement noise. The proposed algorithm facilitates the consistent treatment of control problems, particularly those involving sensors with state-dependent measurement noise, like bearing-only sensors [6], [15], [16]. The effectiveness of the filter is demonstrated through numerical simulations in the context of a control and estimation problem featuring a mobile bearing-only sensor.

This paper aims to enhance the accuracy of the Kalman estimator in the presence of both coloured and white measurement noise using Principal Component Analysis (PCA) [1]. PCA, a technique in multivariate statistical methods [14], is known for data whitening, approximation, and compression, finding applications in various fields like Wireless Sensor Network (WSN) data analysis, process control, fault detection, and disturbance diagnosis [14]. The key advantage of PCA lies in its capacity to transform correlated variables into a new set of uncorrelated orthogonal variables. The proposed approach introduces an online PCA-based Kalman filter estimator, where sensor measurements with coloured noise serve as inputs to a PCA block. The PCA outputs are then utilized to propagate states and the error covariance matrix in the Kalman estimator, aiming to mitigate the impact of correlated variables and enhance estimation accuracy.

The paper is structured as follows: In Section II, we explained the problem definition. The subsequent section delves into the Kalman filter and maximum correntropy Kalman filter with principal component analysis. Section IV provides the simulation results in terms of numerical analysis and Webots simulation environment. Finally, the concluding remarks and avenues for future research are presented in the last section.

*This project has been supported by the National Science Foundation, USA. 2312081 ERI: Drones-Enabled Active Sensing System (DEASS) for Detection of Subsurface Defects in Civil Infrastructure through Synchronized Drone Swarming and Thermal Imaging.

¹ Fethi Candan, Author is with the Department of Soil & Water Systems and Mechanical Engineering, University of Idaho, Moscow, ID83844, United States fcandan@uidaho.edu

² Fethi Candan, Author is with the Department of Aerospace Engineering, Faculty of Aeronautics and Astronautics, Erciyes University, 38030, Kayseri, Turkey fethicandan@erciyes.edu.tr

¹ Johnny (Liu) Li, Author is with the Department of Soil & Water Systems and Mechanical Engineering, University of Idaho, Moscow, ID83844, United States liujunl@uidaho.edu

II. PROBLEM DEFINITION

There are several advantages to using UAVs in agriculture areas [13], but they have some limitations regarding battery consumption and autonomous decisions. When connections with multiple sensors, the ToF of the UAV will be decreased dramatically because of energy consumption; also, when the position sensors are out of communication, the UAVs must guarantee stability and follow the reference trajectory in an emergency. At this point, positioning of the UAVs and communication between UAVs and teleoperators or UAVs and UAVs are significant. Using more accurate and expensive sensors, noise distribution may be calculated.

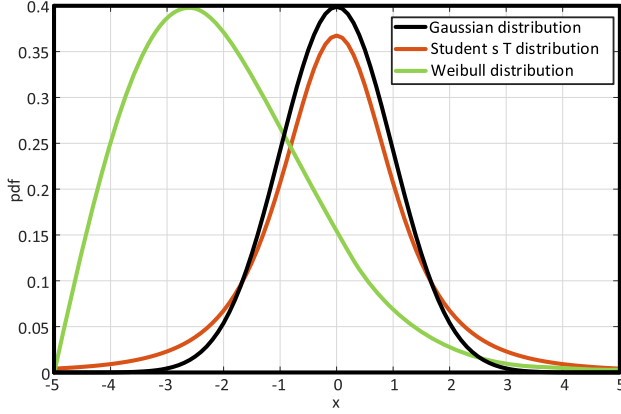


Fig. 1. Gaussian and non-Gaussian Distributions [18]

Considering these expensive and high-accuracy sensors, their datasheets give calculation values for estimation. These values can be defined as internal noises. However, every sensor has a different distribution, and the external noises directly affect the sensors negatively. If given an example for external noises, climate change, wind effects and other weather conditions are defined as external noises, and these calculations are highly challenging topics. That means the sensor noises could be Gaussian-based or non-Gaussian-based distributions. In addition, covariance and standard deviation calculations of the distribution are not the same with Gaussian models. Therefore, it is a challenging problem to predict/estimate values.

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}_0\mathbf{u}(t) + \mathbf{B}_1\mathbf{d}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \omega(t) \end{aligned} \quad (1)$$

$$\omega_t = \Gamma_{t-1}\omega_{t-1} + \xi_{t-1} \quad (2)$$

Generally, Weibull and Student's T distribution have been used for camera-based positioning data estimation, spectral camera, and wind modelling. Therefore, they will be essential to design and implement a model-free sensor estimation algorithm. This proposal focuses on external non-Gaussian and Gaussian noises.

III. METHODOLOGY

In this section, we present the main structure of our methods. The Coverage Path Planning (CPP) reference points

are generated via GPS coordinates, and subsequently, the interval type-2 Fuzzy PID (IT2-FPID) is implemented into the UAV as a position controller [5]. Sensor values are then collected from GPS data, which is able to include Gaussian and non-Gaussian noises [19]. To achieve high accuracy in position data, we employ the Maximum Correntropy Kalman Filter (MCKF) [4]. However, varying levels of noise and disturbance significantly impact UAV performance, despite the MCKF's favourable results under non-Gaussian distributions. To address this, we propose merging adaptive state estimation and multi-kernel methods with MCKF while employing PCA-based regression to classify noises affecting the UAV [14], [20]. This integration results in the PCA-based MCKF (PCA-MCKF), showcasing the innovative "Adaptive State Estimation with Multi-Kernel Approach" feature. In the PCA aspect, the number of samples and the distinction between dependent and independent noise play a crucial role in the estimation process.

A. Kalman Filter

Conventional Kalman Filter (KF) has been widely used for state estimation [4], [6]. In this project, the conventional KF and PCA-KF have been implemented. In Algorithm 1, KF formula has been written. States, state error covariance, model process covariance, and measurement noise covariance have been defined as $\mathbf{x}$, $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{R}$, respectively. In the algorithm, $(\cdot)_k$ is represented as an iteration value.

Algorithm 1 The implementation pseudocode for one time-step of the KF

Inputs: $\bar{\mathbf{x}}_{k-1}$, $\mathbf{P}_{k-1}$, $\mathbf{Q}_{k-1}$, $\mathbf{R}_k$

Prediction: (Time Update)

- 1) $\hat{\mathbf{x}}_k = \mathbf{F}\bar{\mathbf{x}}_{k-1}$
- 2) $\mathbf{P}_k = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}_{k-1}$

Estimation: (Measurement update)

- 1) $\mathbf{K}_k = \mathbf{P}_k\mathbf{H}^T(\mathbf{H}\mathbf{P}_k\mathbf{H}^T + \mathbf{R})^{-1}$
- 2) $\bar{\mathbf{x}}_k = \hat{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_k)$
- 3) $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k)\mathbf{P}_k$

Outputs: $\bar{\mathbf{x}}_k$, $\mathbf{P}_k$.

B. Maximum Correntropy Kalman Filter

Maximum correntropy Kalman filter (MCKF) gives better-estimated results under non-Gaussian or mixed noises. In Algorithm 2, the initial values are the same with KF and also degrees of freedom for distributions (ν), correntropy small positive threshold (ϵ) number and kernel bandwidth (σ) have been added. In [6], The MCKF formula and its application have been detailed and it is represented in Algorithm. 2.

After the explanation of the algorithms, the principal component analysis has been explained in the section.

C. Principal Component Analysis Block Diagram

- 1:** Involves generating a zero-mean dataset from $\bar{\mathbf{Y}}_{1 \times N}$. To ensure the proper functioning of Principal Component Analysis (PCA), it is essential to subtract the mean from each dimension of the data.

Algorithm 2 The implementation pseudocode for one time-step of the MCKF

Inputs: $\bar{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}, \mathbf{Q}_{k-1}, \mathbf{R}_k, v, \varepsilon, \sigma$

Prediction: (Time Update)

- 1) $\hat{\mathbf{x}}_k = \mathbf{F}\bar{\mathbf{x}}_{k-1}$
- 2) $\mathbf{P}_k = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}_{k-1}$

Estimation: (Measurement update)

- 1) $\tilde{e}_k = d_k - \mathbf{w}_k$
- 2) $\mathbf{B}_p = \text{Chol}(\mathbf{P}_{k-1}), \mathbf{B}_r = \text{Chol}(\mathbf{R})$
- 3) $\mathbf{M}_{x,k} = \text{diag}(G_\sigma(\mathbf{e}_{1,k}), \dots, G_\sigma(\mathbf{e}_{n,k}))$
 $\mathbf{M}_{y,k} = \text{diag}(G_\sigma(\mathbf{e}_{n+1,k}), \dots, G_\sigma(\mathbf{e}_{n+m,k}))$
- 4) $\mathbf{P}_{k-1} = \mathbf{B}_{p,k-1}(\mathbf{M}_{x,k})^{-1}\mathbf{B}_{p,k-1}^T$,
 $\mathbf{R}_k = \mathbf{B}_{r,k}(\mathbf{M}_{y,k})^{-1}\mathbf{B}_{r,k}^T$
- 5) $\mathbf{K}_k = \mathbf{P}_{k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k-1}\mathbf{H}_k^T + \mathbf{R})^{-1}$
- 6) $\bar{\mathbf{x}}_k = \hat{\mathbf{x}}_k + \mathbf{K}_k(\tilde{e}_k - \mathbf{H}_k\hat{\mathbf{x}}_k)$
- 7) $\frac{\|\bar{\mathbf{x}}_k - \hat{\mathbf{x}}_k\|}{\|\bar{\mathbf{x}}_k\|} \leq \varepsilon$,
- 8) $\mathbf{P}_k = (\mathbf{I} - \tilde{\mathbf{K}}_k\mathbf{H})\mathbf{P}_{k-1}(\mathbf{I} - \tilde{\mathbf{K}}_k\mathbf{H})^T + \tilde{\mathbf{K}}_k\mathbf{R}_k\tilde{\mathbf{K}}_k^T$

Outputs: $\bar{\mathbf{x}}_k, \mathbf{P}_k$.

$$\bar{\mathbf{Y}}_{1 \times N} = [\bar{y}_1 \dots \bar{y}_N] \quad (3)$$

- 2: The process involves calculating the eigenvectors and eigenvalues of the covariance matrix. Initially, the covariance matrix is derived from $\bar{\mathbf{Y}}_{1 \times N} = \mathbf{Y} - \bar{\mathbf{Y}}$. The transformation occurs through the eigenvectors of this covariance matrix, also referred to as loadings or weights. Each eigenvector comprises loadings for the original data, facilitating their transformation into the new dataset. The information captured by the new data is quantified by their corresponding eigenvalues. Consequently, the new data represented by the first eigenvector, associated with the largest eigenvalue, encapsulates the most information from the original dataset. Mathematically, this can be expressed as:

$$\text{cov} = \frac{\bar{\mathbf{Y}}\bar{\mathbf{Y}}^T}{N-1} \quad (4)$$

$$(\text{cov} - \lambda \mathbf{I})\mathbf{P} = 0$$

- 3: The focus is on selecting eigenvectors and constructing a feature matrix. As previously mentioned, the eigenvalues indicate the amount of information contained in the new data, with the greatest information captured by the eigenvectors associated with the largest eigenvalues. These eigenvectors are the principal components of the dataset. To extract the original data from the presence of noise, an appropriate number of eigenvectors must be chosen, effectively discarding the noise. This is achieved by selecting (m) eigenvectors with the largest eigenvalues and forming a matrix, $(\hat{P}_{1 \times m})$, with these eigenvectors as columns. The first column corresponds to the eigenvector with the largest eigenvalue (z_1), and the one with the smallest eigenvalue is in the last column (z_m).

$$\hat{P}_{1 \times m} = \{z_1, z_2, \dots, z_m\} \quad (5)$$

- 4: After selecting the principal components to retain and forming the feature matrix $(\hat{P})$, the next step involves projecting the data $\bar{\mathbf{Y}}$ into the subspace (R^m) spanned by $\hat{P}$. The resulting projected data $(\hat{P}_{m \times N})$ is expressed as:

$$\hat{\mathbf{Y}} = \hat{P}^T \bar{\mathbf{Y}} \quad (6)$$

This transformation effectively represents the original data in a reduced-dimensional space spanned by the chosen principal components.

- 5: The objective is not merely to reduce dimensionality but to eliminate noise from the original data. To achieve this, the data ($\mathbf{Y}$) is reconstructed in the space $\mathbf{R}^1$ using the projected data set ($\hat{\mathbf{Y}}$). The final data in the $\mathbf{R}^1$ space can be computed as follows:

$$\hat{\mathbf{Y}}_{1 \times N} = \hat{P}\hat{\mathbf{Y}} + \bar{\mathbf{Y}} \quad (7)$$

This step completes the process of denoising and reconstructing the original data in a lower-dimensional subspace, resulting in a final dataset that effectively represents the noise-free information.

- 6: The process involves extracting the modified sensor measurements at the N^{th} time step ($\bar{y}(N)$) from the N^{th} column of the modified data set ($\hat{\mathbf{Y}}$).
- 7: Entails executing the Kalman estimator based on the modified data. Specifically, the noiseless sensor measurements at the N th time step are utilized to estimate states and propagate the error covariance matrix through the standard Kalman or Maximum Correntropy Kalman Filter (MCKF) estimator. This step completes the integration of denoised sensor measurements into the estimation process, contributing to improved accuracy and reliability.

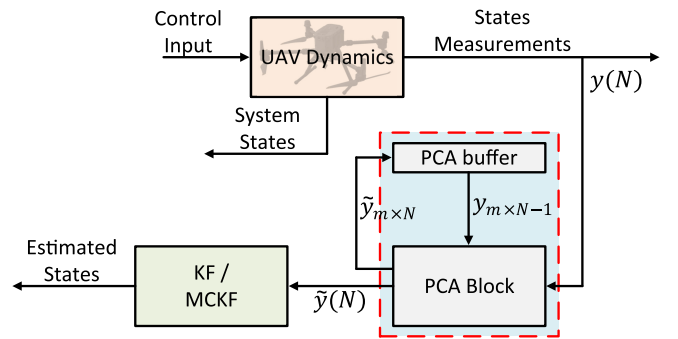


Fig. 2. PCA Block diagram

IV. EXPERIMENTAL RESULTS

For the experiments, Webots Real-time simulation and Matlab simulation environments have been used to compare the conventional Kalman Filter (KF) and the proposed method. The University of Idaho Campus has been used for

the UAV spraying area. Moreover, the DJI Matrice 350RTK solid model has been provided by the DJI company, and then it has been remodelled to apply to Webots. After that, all systems have been tested under dependent noises. That means the defined noises are not based on the Markov chain. It directly depends on previous data value. The first test results show that KF has a faster computation time, but its performance results are not good on the Matlab simulation. The computation time results in 0.002s and 0.01s, respectively. However, computation is the negligible criterion for the UAV. Therefore, total task time and root mean square error (RMSE) have been selected for the comparison criteria. Under different levels of noise, PCA-MCKF gives better RMSE value and faster task compilation. Moreover, a different number of samples have been applied to show performance results.

A. Numerical Experiment Results

For numerical experiments, a constant velocity motion model has been used. The sampling time (T_s) has been set as 0.05s. The selected motion model has been written in Eq. 8. It is noted that, Matlab environment has been used for the numerical analysis [17].

$$\mathbf{F} = \begin{bmatrix} 1 & 0.05 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.05 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Then, the model process covariance $\mathbf{Q}$ have been written in Eq. 9

$$\mathbf{Q} = \begin{bmatrix} 1.562e-05 & 0 & 0 & 0 \\ 0 & 0.025 & 0 & 0 \\ 0 & 0 & 1.562e-05 & 0 \\ 0 & 0 & 0 & 0.025 \end{bmatrix} \quad (9)$$

The last matrices, measurement and measurement noise covariance, have been defined in Eq. 10 and Eq. 11.

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (10)$$

$$\mathbf{R} = \begin{bmatrix} 10 & 0 \\ 0 & 100 \end{bmatrix} \quad (11)$$

The outcomes of the numerical analysis are depicted in Fig. 3. In the figure below, the Coverage Path Planning (CPP) reference trajectory is depicted by the black line. Subsequently, various levels and types of noise were introduced (Grey line). The proposed approach (PCA-MCKF) and the compared methods (KF, MCKF, and PCA-KF) are also presented. The findings indicate that PCA-MCKF exhibits a superior reference trajectory and smoother performance compared to the other methods. Furthermore, the results of KF and MCKF without PCA reveal a higher degree of oscillation compared to the others.

The numerical experiments were conducted in both numerical and offline modes, where the effects of the controller and

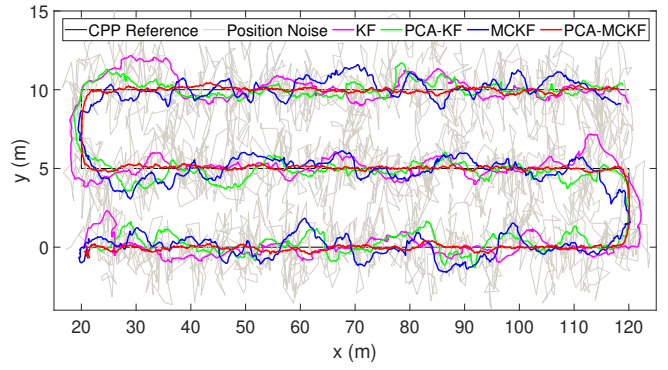


Fig. 3. Numerical Experiments Results

other uncertainties were eliminated. Consequently, the proposed and compared methods underwent testing in Webots Simulation environments to assess their performance under more realistic conditions.

B. Webots Environment Results

In this section, the proposed and the compared estimation methods have been tested on Webots environments. Unlike numerical experiments, the drone has been modelled and the interval type-2 Fuzzy PID (IT2-FPID) controller has been implemented. After the implementation, coefficients of the controllers has been tuned using the Big Bang-Big Crunch optimisation algorithm [10]. For experiments, the selected UAV, DJI Matrice 350RTK has been converted to URDF model [8] and imported to Webots [9]. In Fig. 4, the designed UAV, its position and its orientation can be seen.

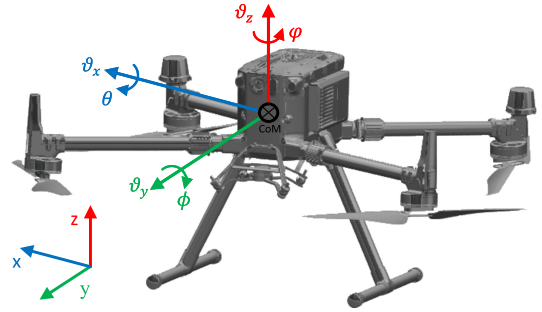


Fig. 4. DJI Matrice 350RTK URDF Model for Webots

Considering the UAV dynamical equations [3], the position ($\mathbf{p}$) of the UAV is defined as $[x, y, z]^T$. After that, the acceleration ($\dot{\mathbf{v}}$) is written in Eq. 12. It is noticed that UAV motion and dynamical equations have been determined as $\times$ form.

$$\dot{\mathbf{v}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{b}{m} \sum_{i=1}^4 \Omega_i^2, \quad (12)$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi) \\ -s(\phi)c(\theta) + c(\phi)s(\theta)s(\psi) \\ c(\phi)c(\theta) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}.$$

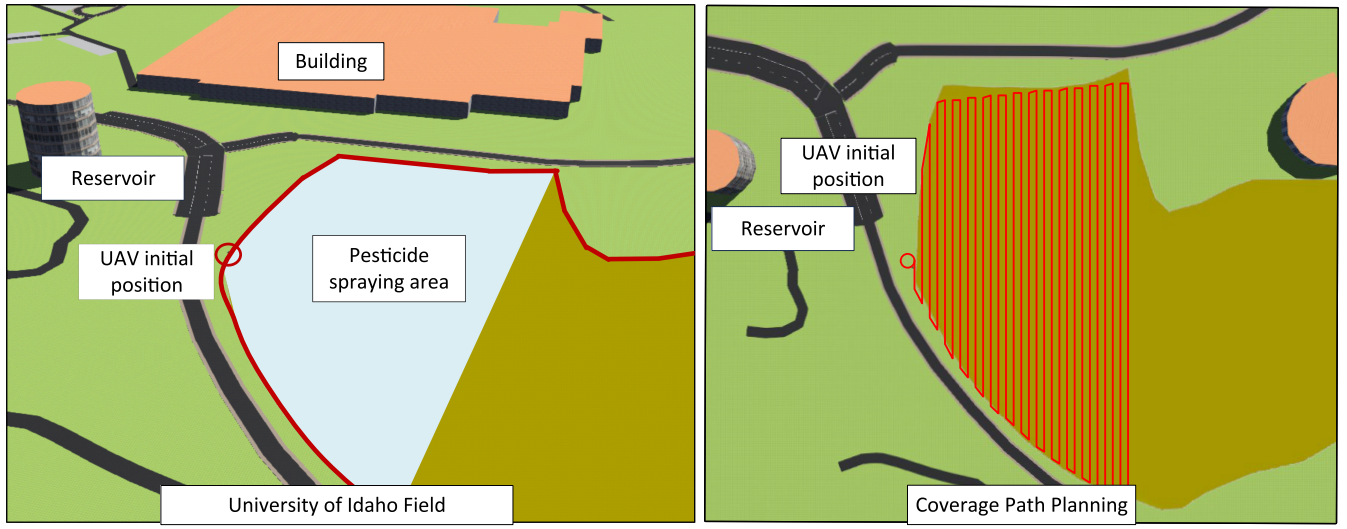


Fig. 5. Simulation Scenario on Webots

It is remarked that $s(\cdot)$ and $c(\cdot)$ are abbreviated from sine and cosine, respectively. Moreover, v is the velocity of the UAV rotor, b is the thrust coefficient, m is the mass of the UAV and also, g defines gravity.

The above equations have been used to find optimal control parameters for the Webots simulation environment. In this paper, IT2-FPID has been used and implemented into the defined UAV as a high-level controller. To find optimal parameters of the controller, the Big-Bang Big-Crunch algorithm has been used [10] and the tuned parameter has been shown in Table I.

TABLE I
THE OPTIMISED IT2-FPID CONTROLLER PARAMETERS

	K_e	K_d	K_0	K_1	α	ζ
IT2-FPID	0.0034	0.003	0.001	65	0.25	0.5

In Fig. 5, Webots simulation environment has been represented. The selected area is the golf zone in the University of Idaho. The selected UAV have had pesticide spraying tasks in this area. For spraying, coverage path planning (CPP) has been applied and each point has been generated as reference signals. In Fig. 5, pesticide spraying area and designed coverage path planning have been shown.

In Fig. 6, the simulation results of the pesticide spraying have been plotted. When focusing one of edges on CPP, the proposed method, PCA-MCKF, has the most accurate results in terms of tracking the reference. Also, without PCA, MCKF is better than both KFs.

C. Performance Results

The designed numerical and simulation environments have been tested 100 times repeated independently. After that, root mean square (RMSE) values have been calculated as performance criteria. In Table II, the performance results and computation times have been represented. The results show that PCA-MCKF and MCKF have better results than PCA-KF and KF under dependent noise and mixed distributions.

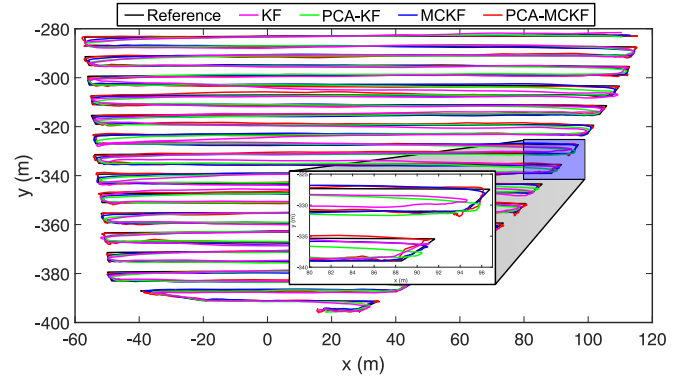


Fig. 6. Simulation Results on Webots Simulation Environment

Both simulation parts, PCA-MCKF is three times better accuracy than PCA-KF and four times better than KF. However, computation time is a big problem for MCKFs. PCA-MCKF is the slowest and it is slower three times than KF.

TABLE II
TEST RESULTS OF KF AND PCA-MCKF

Model	Webots		Computer (Matlab)	
	Time	RMSE	Time	RMSE
MCKF	7850 sec	0.198	6880 sec	0.137
KF	3250 sec	0.415	2415 sec	0.328
PCA-MCKF	9615 sec	0.113	7530 sec	0.068
PCA-KF	3580 sec	0.365	2865 sec	0.343

V. CONCLUSION AND FUTURE WORKS

This paper addresses challenges in agricultural unmanned aerial vehicles (A-UAVs), particularly in low-altitude flight control and accurate position estimation essential for coverage path planning. The focus is on handling coloured noise, specifically noise dependent on the previous value and influenced by factors like wind disturbance and GPS sensor

acceleration. The proposed solution involves a PCA-based maximum correntropy Kalman filter (MCKF) to enhance stability, mitigate oscillations, and address uncertainties. The method is compared with traditional Kalman filter (KF), PCA- KF, and PCA- MCKF. The study involves the design of four rotor-wing UAVs with linear and nonlinear dynamical models, and a Fuzzy PID controller is chosen as the intelligent control method. Constant acceleration and constant velocity manoeuvre models are employed for estimation to simplify model effects and reduce computation time. Performance criteria are evaluated using root mean square error, and real-time simulations in the Webots environment demonstrate that the proposed PCA- MCKF outperforms other methods, showcasing its potential for improving accuracy in agricultural UAV applications.

REFERENCES

- [1] M. Afshari, A. Tavasoli, and J. Ghaisari, "A pca-based kalman estimation approach for system with colored measurement noise," in *20th Iranian Conference on Electrical Engineering (ICEE2012)*. IEEE, 2012, pp. 969–973.
- [2] A. D. Boursianis, M. S. Papadopoulou, P. Diamantoulakis, A. Liopa-Tsakalidi, P. Barouchas, G. Salahas, G. Karagiannidis, S. Wan, and S. K. Goudos, "Internet of things (iot) and agricultural unmanned aerial vehicles (uavs) in smart farming: A comprehensive review," *Internet of Things*, vol. 18, p. 100187, 2022.
- [3] F. Candan, A. Beke, and T. Kumbasar, "Design and deployment of fuzzy pid controllers to the nano quadcopter crazyflie 2.0," in *2018 Innovations in Intelligent Systems and Applications (INISTA)*. IEEE, 2018, pp. 1–6.
- [4] F. Candan, A. Beke, C. Shen, and L. Mihaylova, "An interacting multiple model correntropy kalman filter approach for unmanned aerial vehicle localisation," in *2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 2022, pp. 1–6.
- [5] F. Candan, O. F. Dik, T. Kumbasar, M. Mahfouf, and L. Mihaylova, "Real-time interval type-2 fuzzy control of an unmanned aerial vehicle with flexible cable-connected payload," *Algorithms*, vol. 16, no. 6, p. 273, 2023.
- [6] B. Chen, X. Liu, H. Zhao, and J. C. Principe, "Maximum correntropy kalman filter," *Automatica*, vol. 76, pp. 70–77, 2017.
- [7] B. Chen, L. Xing, H. Zhao, N. Zheng, J. C. Pri et al., "Generalized correntropy for robust adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3376–3387, 2016.
- [8] D. Company, "Matrice 350rtk," <https://enterprise.dji.com/matrice-350-rtk>, accessed: 2023-12-17.
- [9] Cyberbotics, "Webots simulation ide," <https://cyberbotics.com>, accessed: 2023-12-17.
- [10] O. K. Erol and I. Eksin, "A new optimization method: big bang–big crunch," *Advances in engineering software*, vol. 37, no. 2, pp. 106–111, 2006.
- [11] A. Fakharian, T. Gustafsson, and M. Mehrfam, "Adaptive kalman filtering based navigation: An imu/gps integration approach," in *2011 International Conference on Networking, Sensing and Control*. IEEE, 2011, pp. 181–185.
- [12] K. Feng, J. Li, X. Zhang, C. Shen, Y. Bi, T. Zheng, and J. Liu, "A new quaternion-based kalman filter for real-time attitude estimation using the two-step geometrically-intuitive correction algorithm," *Sensors*, vol. 17, no. 9, p. 2146, 2017.
- [13] P. K. Freeman and R. S. Freeland, "Agricultural uavs in the us: potential, policy, and hype," *Remote Sensing Applications: Society and Environment*, vol. 2, pp. 35–43, 2015.
- [14] S. Li, D. Shi, W. Zou, and L. Shi, "Multi-kernel maximum correntropy kalman filter," *IEEE Control Systems Letters*, vol. 6, pp. 1490–1495, 2021.
- [15] C. Liu, G. Wang, X. Guan, and C. Huang, "Robust m-estimation-based maximum correntropy kalman filter," *ISA transactions*, vol. 136, pp. 198–209, 2023.
- [16] X. Liu, Z. Ren, H. Lyu, Z. Jiang, P. Ren, and B. Chen, "Linear and nonlinear regression-based maximum correntropy extended kalman filtering," *IEEE transactions on systems, man, and cybernetics: Systems*, vol. 51, no. 5, pp. 3093–3102, 2019.
- [17] M. MATLAB, "Mathworks matlab, control system toolbox," <https://www.mathworks.com/products/control.html>, accessed: 2023-12-17.
- [18] M. Roth, E. Özkan, and F. Gustafsson, "A student's t filter for heavy tailed process and measurement noise," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 5770–5774.
- [19] S. Saha and F. Gustafsson, "Particle filtering with dependent noise processes," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4497–4508, 2012.
- [20] A. Singh and J. C. Principe, "Using correntropy as a cost function in linear adaptive filters," in *2009 International Joint Conference on Neural Networks*. IEEE, 2009, pp. 2950–2955.