Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-32 (2022) 107–112

Sensor optimization for altitude estimation of spraying drones in vineyards

Thomas Severin [*] **Dirk Söffker** [**]

[*] *Department of Agricultural Engineering, Hochschule Geisenheim University, 65366 Geisenheim, Germany (e-mail: thomas.severin@hs-gm.de)*

[**] *Chair of Dynamics and Control, University of Duisburg-Essen, Duisburg, Germany (e-mail: soeffker@uni-due.de)*

Abstract: Current drones find a variety of applications, including in viticulture. The application of pesticides in steep slope viticulture is necessary to control pathogens. Due to the characteristics of the applied liquids, high demands are placed on the robustness and reliability of the drone and the selected sensors. For the application of the particle-laden fluid, a specific altitude is useful, which should be realized by a flight attitude control system. Steep slope situations generate increased demands on accurate relative altitude estimation. This study compares various low-cost sensors for ground distance measurement suitable for spray drones. Using the overflight of a vineyard as an example, the problem is demonstrated, sensor data are recorded for evaluation, and suitable parameters and strategies are set. Sensor signals based on ultrasound, range radar, and Doppler radar are filtered using a Kalman filter. Various measures are presented to optimize the estimated altitude to improve the reliability in terms of relative altitude estimation. The presented algorithm will be implemented on a companion computer parallel to the flight controller.

Keywords: Kalman filter, Filtering and smoothing, Agricultural robotics, Viticulture, UAV.

1. INTRODUCTION

1.1 Challenges of spraying drones in viticulture

Unmanned aerial vehicles (UAV) are becoming of increasing relevance in the field of agriculture. A reasonable application is plant protection spraying. In addition to applications for precision agriculture, an UAV (drone) can be used in areas that are difficult to access. This is especially true for vineyards with steep slopes (Fig. 1).

Vine monocultures are threatened by a wide range of pathogens (viruses, bacteria, fungi, and parasites) which either cause direct damage to plants and fruits or transmit pathogens. This can lead to a loss of crops and a decrease in the quality of the fruit and vines. For these reasons, winegrowers are advised to use plant protection products in their vineyards.

To increase efficiency and to reduce the amount of crop protection agent, it is necessary for drones to have accurate height and flight path control during crop protection application.

Drones rely on sensor readings for flight control. The adverse conditions during spraying applications pose a challenge. The leaves, dirt, and drops contaminating range sensors. Sensors must be durable and reliable.

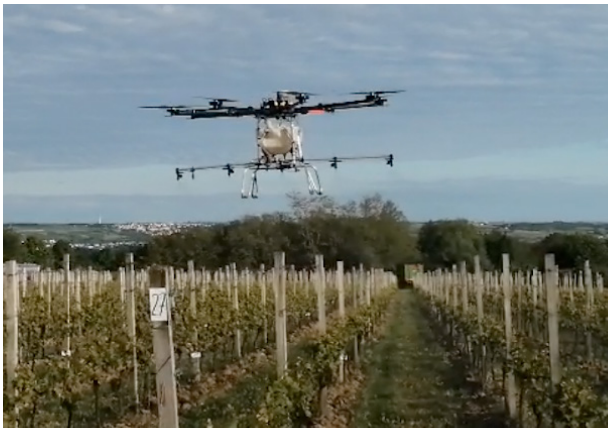

Fig. 1. Drone for plant protection application.

Machinery used to protect crops must be approved by EU Directive EU-2009/128. As few harmful chemicals as possible should get into the environment. In addition, UAVs and their operations are regulated by the EU Directives 2019/947 and 2019/945.

In the following, the focus is placed on altitude estimation, as this aspect is especially relevant for spraying drone applications in steep-slope viticulture. Range sensors are mounted on a drone to measure the distance to the ground. In this article, the focus is related to the quality of the filtered signal.

[★] The project is supported by funds of the Federal Ministry of Food and Agriculture (BMEL) based on a decision of the Parliament of the Federal Republic of Germany. The Federal Office for Agriculture and Food (BLE) provides coordinating support for digitalisation in agriculture as funding organisation, grant number 28DE105A18.

10.1016/j.ifacol.2022.11.123

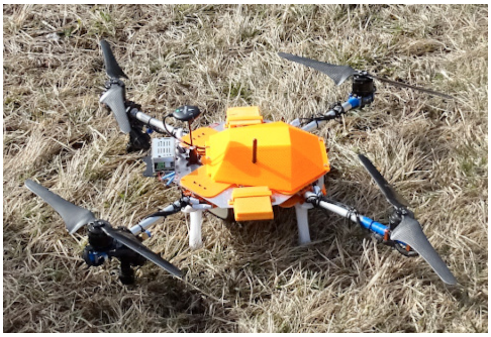

Fig. 2. Drone for measurements, all distance sensors are directed downward.

1.2 Altitude control and estimation

Modern drone flight controllers for the specified and open classes consist of estimators. The controller in the flight controller uses an extended Kalman filter algorithm to process sensor measurements and provide state estimates of

- Quaternions (defining the rotation in the local frame),
- X,Y,Z positioning and
- kinematic (velocity, acceleration)

To estimate these states, sensor magnetometers, barometers, GPS, gyroscopes, and accelerometers are typically integrated into the IMU (Initial Measurement Unit). A wide range of additional sensors are available that can be merged by the estimator. Flight controllers like the Open Source PX4 provide an API for companion computers (Meier et al. (2015)). Companion computers are therefore capable to provide additional functionalities.

The most essential sensors are designed to be redundant. Redundant means that a sensor fault or erroneous measurement like drift, outliers, bias, etc. is not leading to a malfunction of the drone. For example, the IMU contains sensors like magnetometers from different manufacturers.

Drones have no direct odometry (Achtelik et al. (2009)). Drones are dependent on the measurement of mostly low-cost, IC-integrated, sensors like accelerometers, magnetometers, gyroscopes, and GPS. In outdoor environments, it is difficult to achieve an exact flight with a precision of a few centimeters. Wind, changes in GPS accuracy, lower quality of MEMS (Micro-Electro-Mechanical Systems), and magnetometer bias are examples of such kinds of influences, see de Alteriis et al. (2021).

The aim of the sensor fusion is to improve the accuracy of the estimation of the local position. All sensors have different measurement rates. GPS, for example, has a sample rate of 1 Hz, while the sensors within the IMU use a sample rate of 100 Hz, see Meier (2021) and Li and Fu (2018). The different measurement frequencies and the conversion of global GPS positioning to a geodetic local frame provide additional effort.

The measurement values given by the altitude sensors must be filtered, weighted, and combined to get an accurate height estimate. The optimized height information is sent to the flight controller. The flight controller is capable of using this information as a relative altitude indicator. During the mission planning, a relative altitude can be defined, which will be used as a set point for the altitude control during flight operations.

In Driessen et al. (2018) an Extended Kalman filter for sensor fusion of an IMU, an optical-flow sensor (CMOS cams), and a sonar sensor for altitude and attitude estimation is used. The sonar is measuring the altitude, while the flow sensor is measuring the velocity in the horizontal x and y axis of the UAV body frame. A complete attitude controller is introduced to handle the drone. This kind of sensor combination is common for altitude estimation. It has the disadvantage that a failure of one sensor leads to a failure of the altitude estimation.

The authors of Yang et al. (2021) used a LiDar Lite V3 to implement terrain following control for an UAV. No sensor redundancy is given and a complete UAV odometry for altitude control is implemented. In this article, an already flight controller provided interface for altitude estimation is used.

An approach to increase accuracy is to implement outlier filters. Outliers are sensor data points that lie outside of a common range. A study on combining an outlier filter with a Kalman filter was conducted by Ting et al. (2007) on a walking robot dog. The Kalman filter is combined with a scalar Bayesian weighting factor (gamma distribution) for each measurement value. Such weighing is described as an "Expectation-Minimization-like (EM) learning problem", according to Dempster et al. (1977).

Thus, the aim of this study was to assess a simple and reliable sensor setup for altitude control of a viticulture spraying drone, especially suited for the requirements of steep-slope vineyards. As an additional practical requirement, the rows of the vine and its leaves must be ignored by the altitude estimation. The vineyards are systematically disturbing the altitude estimation using related sensors. A temporary non-working sensor must be detected and excluded. The remaining sensors must be able to generate a sufficient base for the altitude estimation. If an errornous sensor is not switching back to normal operation, the error messaging system must be informed about the reduced number of sensors.

In Section 2, the technical and mathematical methods for this work are briefly explained. Besides the presentation of the different sensors, the interface to the controller and the mathematical model of the filter are described. In Section 3, the experiment and its evaluation are described.

2. MATERIALS AND METHODS

2.1 Drone setup

The spraying drone considered is an individual-constructed drone for carrying four sensors using the PX4 flight controller and a Raspberry Pi companion computer. All distance measurement sensors for relative altitude estimations are oriented downwards. The specifications of these sensors are given in table 1.

The accuracy and range depend on further factors. For example, the sun light can reduce the accuracy of the laser probe. Enclosures and drone parts can also reduce the

Table 1. Sensors for distance measurement. The standard deviation is signed with σ.

Type	Freq.	Range	σ
Dopp. radar OPS241-A	15 Hz	$0\,\mathrm{m\,s^{-1}}$ to $5\,\mathrm{m\,s^{-1}}$	0.5 %
Range radar OPS241-B	15 Hz	0.1 m to 20 m	0.5 %
Ultrasonic HC-SR04	10 Hz	0.2 m to 3.5 m	3 mm
1D laser STM VL53L1X	5 Hz	0.0 m to 3.5 m	0.5 %

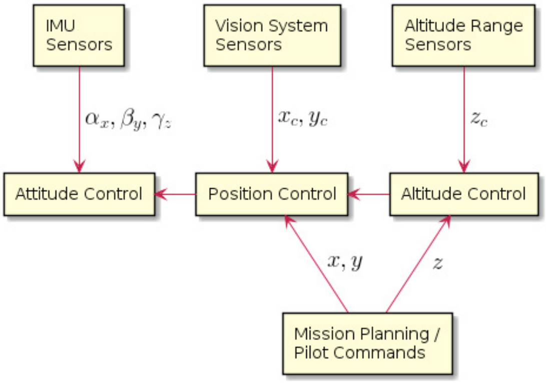

Fig. 3. Control schema of an UAV controller according to Bouabdallah and Siegwart (2007). The control units are today standard components in common flight controllers.

range and accuracy of the radar. Rectangular speed above $1.5\,\mathrm{m\,s^{-1}}$ reduces the range of the used ultrasonic sensor. The UAV is flying rectangular to the reflected signal.

Kalman Filter and system model A Kalman filter can be set up with a basic mathematical model of the drone's altitude behavior.

The prediction of the system state vector is the result of merging model knowledge with assumed or updated knowledge about sensors, model errors, and sensor signals. The Kalman filter consists of an update step and a prediction step. This article focuses on the interfaces between the already implemented control loops of the flight controller and the Kalman filter combined sensors for altitude control in Fig. 3. This algorithm is running on a companion computer. The flight controller provides an interface for the external altitude sensor system.

The system state space representation is described by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{w}, \qquad (1)$$

where $\mathbf{x}$ denotes the state vector, $\mathbf{A}$ the system matrix, $\mathbf{B}$ the input matrix and $\mathbf{w}$ the system noise.

A discrete description is used. With

$$\hat{\mathbf{x}}_k = \mathbf{F}\hat{\mathbf{x}}_{k-1} + \mathbf{B}_k\mathbf{u}_k \qquad (2)$$

the prediction is calculated for a time discrete system.

The state transition matrix $\mathbf{F}$ can be derived from the state space representation. The transformation matrix $\mathbf{F}$ computes the change in $\mathbf{A}$ over a discrete time step.

By substituting the power series

$$\mathbf{F} = e^{\mathbf{A}\Delta t} = \mathbf{I} + \mathbf{A}\Delta t + ... \qquad (3)$$

and using the system matrix

Table 2. System model variables

Variable	Description	Unit
d_1	Range radar Z-Axis	m
d_2	Laser probe Z-Axis	m
d_3	Ultra sonic sensor Z-Axis	m
d_4	Doppler radar Z-Axis	m/s
x_1	State variable distance Z-Axis	m
x_2	State variable speed Z-Axis	m/s
α_{roll}	Roll angle	degree
α_{pitch}	Pitch angle	degree
y_1	Z altitude set point	m
y_2	Drone speed Z direction	m

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad (4)$$

the transition matrix

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \qquad (5)$$

can be derived.

During flight operations, the derivation of the altitude should be zero. Further process-related changes are considered by a time discrete white noise. For a basic concept, the state vector and the state transition function are designed considering the kinds of available sensors, distance, and their derivation. The speed of the altitude is measured by the Doppler radar.

The state vector corresponds to the distance to the ground and its derivative

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix}. \qquad (6)$$

The state covariance $\mathbf{P}$ is an indication for the trust in the with $\mathbf{F}$ described system

$$\mathbf{P}_i = \begin{bmatrix} \sigma_{z_i}^2 & 0 \\ 0 & \dot{\sigma}_{\dot{z}_i}^2 \end{bmatrix}. \qquad (7)$$

The measurement vector and the measurement covariance matrix is described by

$$\mathbf{z}_i = \begin{bmatrix} \mathrm{z_i} \\ \dot{\mathrm{z}}_\mathrm{i} \end{bmatrix}, \quad \mathbf{R}_i = \begin{bmatrix} \sigma_{z_i}^2 & 0 \\ 0 & \sigma_{\dot{z}_i}^2 \end{bmatrix}. \qquad (8)$$

The set variance during the experiments in dependence of the sensor can be found in table 2.3.

Measurement data The modification of the measured values by consideration the pose of the UAV is necessary.

With the attitude angles provided by the flight controller (table 2), the measurement variables can be calculated with

$$z_i = d_i \cos(\alpha_{roll}) \cos(\alpha_{pitch}), \qquad (9)$$

where the indexes of z_i and d_i correspond to the radar, Doppler, ultrasonic, and laser sensor.

The asynchronous incoming measurements are fused regarding the algorithm 1. Sensor fusion means here, that the updated system state vector $\mathbf{x_k}$ is calculated using the sensor specific $\mathbf{R}_i$. In this sense, the update is triggered when any sensor signal is 'ready'. The prediction is calculated after each time step.

Algorithm 1 Asynchronous sensor fusion

```
procedure UPDATELOOP
    if (Sensor Radar ready) then
    UPDATE$(x, P, z_{Radar}, R_{Radar}, H)$
    if (Sensor Laser ready) then
    UPDATE$(x, P, z_{Laser}, R_{Laser}, H)$
    if (Sensor US ready) then
    UPDATE$(x, P, z_{UltraSonic}, R_{UltraSonic}, H)$
    PREDICT$(x, P, F, Q, B, u)$
```

The software package used is published by Labbe (2021).

Outlier and robust filter A Gaussian weighting factor is used in (11) to change the measurement covariance matrix $\mathbf{R}_i$ dynamically. The planned flight altitude is one parameter of the weighting factor. With this factor, the measurement covariance $\mathbf{R}_i$ is updated with regard to the weighted measurement value

$$\mathbf{R}_{w_i} = g(z_i, \mu_{alt}, p_w)\mathbf{R}_i \tag{10}$$

with

$$g(z_i, \mu_{alt}, p_w) = 1 - \frac{p_w}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(z - \mu_{alt})^2}{2\sigma^2}\right). \tag{11}$$

The parameters in (11) are

- Altitude mean value μ_{alt} given by flight controller interface / mission planning,
- Weighting factor p_w for range with higher trust. Altitude mean value is implemented as a value given from the flight controller,
- Measurement value z_i of the corresponding sensor.

.

With (11), an inverted Gaussian can be defined, which weights the sensor variance regarding the predefined flight altitude.

2.2 Fault detection

In this section, the implemented failure detection and handling is presented. The requirement is, that a sensor failure should not lead to errors in height detection.

In the algorithm 1, the condition for accepting the measurement (sensor ready) is therefore combined with further conditions. The first condition for accepting the measurement is that $x_1 - z_1$, the state height minus the measured value, has to be lower than a threshold. The threshold is set to the μ_{alt} altitude of the flight mission. Outliers due to height, roll, and pitch angles of the UAV are therefore considered.

The second condition is, that the measurement must be larger than 0.02 m. This value is especially relevant to the radar. For this sensor, reflections on the enclosure or UAV are assumed (protection foil), which generates iterations with a frequency between 6 and 8 Hz. This is half of the maximal measurement frequency. By filtering radar measurements below 0.02 m, this reflection can be filtered out.

Table 3. Variance for different covariance matrices.

	σ_z	$\sigma_{\dot{z}}$	$\sigma_{z\dot{z}}$	$\sigma_{\dot{z}z}$
R_{Radar}	9	6	0	0
$R_{Doppler}$	12	6	0	0
R_{US}	16	6	0	0
R_{Laser}	22	6	0	0
P	2	6	0	0

2.3 Experimental site

The goal is to optimize the relative altitude estimation to the ground as this information allows the flight controller to fly in an approximation parallel to the terrain. This problem is especially challenging in the case of flying perpendicular to vine rows. This has been considered in the chosen flight path over an experimental vineyard from the Hochschule Geisenheim University, Johannisberg. The selected vineyard for test flights is located at latitude 49.994588° and longitude 7.974138° at an elevation of 153 m asl (above sea level). The vines are oriented transversely to the slope. The flight path covers a height difference of 1 m. The distance between the rows is approximately 2 m. This is a common distance to realize the movement of narrow-gauge tractors.

The selection criteria for the choice of the test field are

- viticulture slopes,
- sufficient distance to infrastructure (150 m), and
- barrier possibilities of the access roads.

The relative flight altitude is set to 3.5 meters relative to the start position. The drone flies along a path defined by GPS positions. The height difference due to the slope of the cultivated area is approximately 1 meter. The take-off and landing points (below) are the overall lowest points, while the upper u-turn is the highest point. The speed is defined by the plant protection application. Typical values are in between 1.5 m s^{-1}and 2 m s^{-1}. In this experiment, 2 m s^{-1} was used.

The sensors continuously record measurement data during the flight, which is used for later evaluation. A log file is generated on the UAV in which the measured values are entered asynchronously for each range sensor. Height control by sensors is not required during recording. Only GPS altitude data was used for flying. This step requires prior recording of GPS positions along the planned flight path and is therefore only practical for an experimental setup (see section 2.4).

The altitude of the UAV is evaluated with GPS-RTK and compared with landmarks (measurement points) to determine the true distance between the ground and the UAV in the data evaluation. The pillars of the vine yard to connect the vineyard wires were used as reference points.

2.4 Terrain height calculation

Reference altitudes for flight path over the experimental field were measured at twenty positions along the path by setting up a GPS-RTK antenna around 2 m above the ground by placing it at the grapevine pillars (table 2). The altitude variance and the horizontal position for all measured points is below 0.02 m. For the calculation of

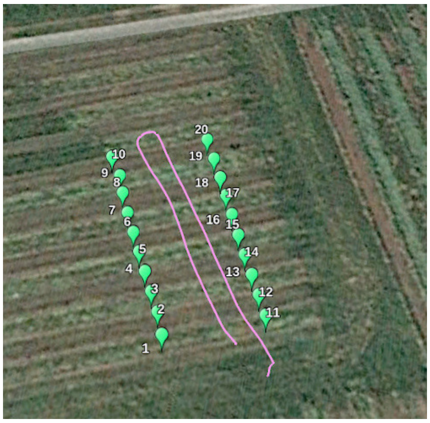

Fig. 4. Flight trajectory (purple) during the experiment with GPS-RTK reference positions for terrain measurement (green dots)

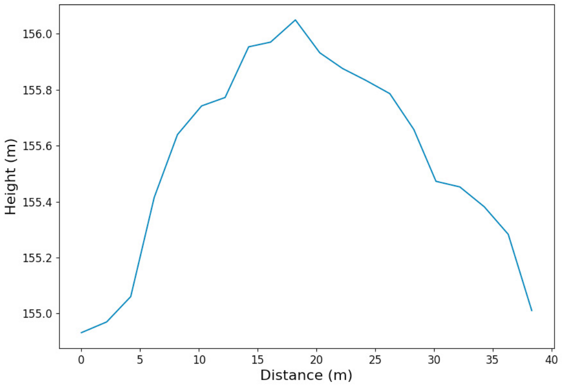

Fig. 5. GPS measurement points 1–20 absolute terrain height.

the distance between the GPS positions of the measuring points (pillar tips), the Haversine function was chosen, Alkan and Celebi (2019). This function is used to calculate the distances between two GPS coordinates. With the equation

$$y_1 = altitude_{UAV} - altitude_{terrain} \tag{12}$$

the reference relative altitude y_1 is calculated. The absolute $altitude_{UAV}$ (asl) is provided by the GPS-RTK of the UAV, while the $altitude_{terrain}$ is given by the interpolated terrain measurement points (Fig. 5).

3. RESULTS

3.1 Experimental results

In Fig. 4 the flight track of the drone during the experiment is visualized. The GPS-RTK measurement points are also shown. The absolute terrain height is visualized in diagram 5. In Fig. 6, the estimates of the Kalman filter are shown. The first two axes of the diagram of the experiment show the measurement of the sensor data (radar, ultrasonic and laser range sensor). The orange line is the true value of the UAV altitude y_1. The third horizontal axis shows the estimated altitude (distance to ground) from the Kalman filter in comparison to the true value y_1 (orange line). The fourth graph displays the measurements coming from the Doppler radar. In comparison to the GPS z-axis velocity (derivative of altitude change). The last diagram shows the angle estimation (pitch and roll angle) supplied by the flight controller.

Another field of investigation is the detection of faulty data. For example, another experience was made where the range radar got completely faulty data for a few seconds. One reason could be drops, dirt, or a cable that disturbs the measurement. Interference can also happen to other sensors installed on the drone. In Fig. 7, the range radar in this second experiment under the same conditions has two failures in the first 15 seconds, with inconsistent data. The measurement value is around 0.01 m due to drops or leaves in front of the sensor. The laser range sensor has also failed. There is no laser range data available up to 18 seconds. Despite these disturbances, the estimated flight height in the 3rd row is relatively accurate due to the different outlier and weighting approaches.

The ultrasonic radar has the slowest measurement frequency. At a speed of the UAV by 1.5 m s^{-1}, the signal reflection of the distance arrives the ultrasonic sensor 0.02 m behind the sensor. This characteristic reduces the maximal range when flying at a higher speed than 1.5 m s^{-1}. Laser sensors can be disturbed by sunlight, which can also harm the range. The 1D range laser STM VL53L1X has the ability to reduce the frequency automatically for accuracy improvements in the case of measurement bias. This behavior results in a reduced update frequency during flight. The variance between the real relative altitude y_1 and the estimation x_1 was for both experiments below 0.1 m, which is reasonable enough for an approximate flight parallel to the terrain. Generally, the smoothness of the filter should be more important than a strict parallel flight to the terrain.

With a standard Kalman filter, a more smooth estimation of altitude is possible. The dynamic weighting factor of the sensor values and a few simple outlier filters show that even a failure of one sensor in a combination of minor disturbances of a second sensor does not yield a faulty and useless altitude estimation. The different used sensors have different characteristics regarding possible issues (reflections, sunlight). Therefore, a combination of different kinds of sensors contributes to more robust and accurate altitude estimation.

ACKNOWLEDGEMENTS

The authors acknowledge Prof. Kai Velten, Professorship for Mathematics, Statistics, Mathematical Modeling and Simulation, Hochschule Geisenheim University for additional support.

REFERENCES

Achtelik, M., Bachrach, A., He, R., Prentice, S., and Roy, N. (2009). Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. In *Unmanned Systems Technology XI*, volume 7332, 733219. International Society for Optics and Photonics.

Alkan, H. and Celebi, H. (2019). The implementation of positioning system with trilateration of haversine distance. In *2019 IEEE 30th annual international sym-*

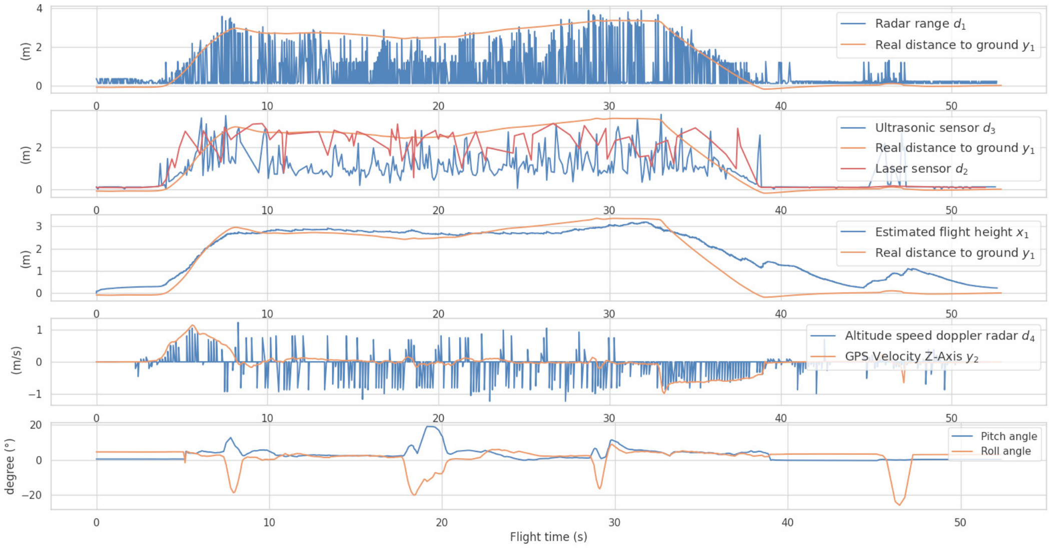

Fig. 6. Experiment 1: Relative altitude estimation x_1 based on various sensors.

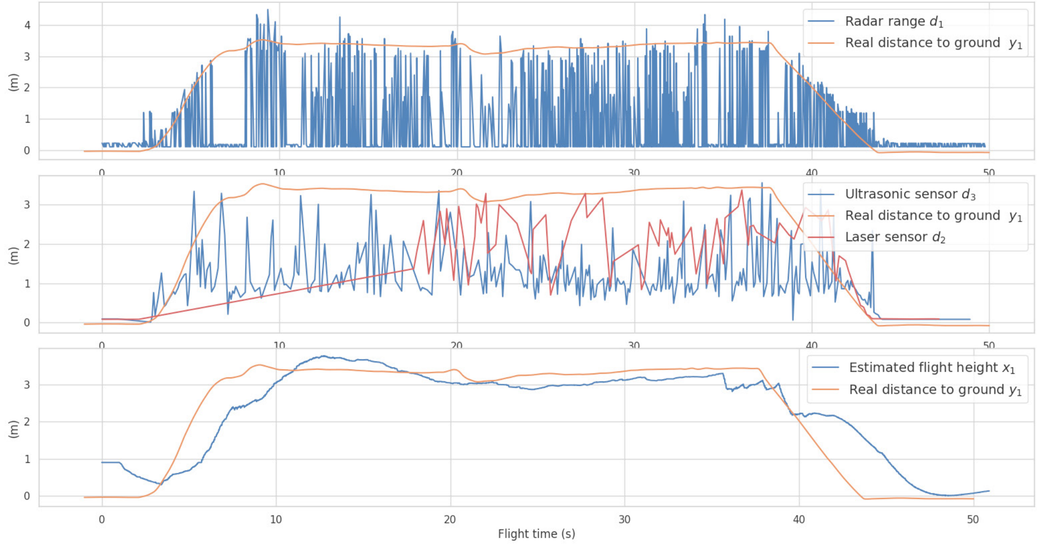

Fig. 7. Experiment 2: Errornous sensor data given by range radar and laser sensors.

posium on personal, indoor and mobile radio communications (PIMRC), 1–6. IEEE.

Bouabdallah, S. and Siegwart, R. (2007). Full control of a quadrotor. In *2007 IEEE/RSJ international conference on intelligent robots and systems*, 153–158.

de Alteriis, G., Accardo, D., Conte, C., and Schiano Lo Moriello, R. (2021). Performance enhancement of consumer-grade mems sensors through geometrical redundancy. *Sensors*, 21(14), 4851.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1–22.

Driessen, S., Janssen, N., Wang, L., Palmer, J., and Nijmeijer, H. (2018). Experimentally validated extended kalman filter for uav state estimation using low-cost sensors. *IFAC-PapersOnLine*, 51(15), 43–48.

Labbe, R. (2021). Kalman and bayesian filters in python. URL `https://filterpy.readthedocs.io/en/latest/`. Visited on 2022-02-19.

Li, W. and Fu, Z. (2018). Unmanned aerial vehicle positioning based on multi-sensor information fusion. *Geo-Spatial Information Science*, 21(4), 302–310.

Meier, L. (2021). Pixhawk standards. URL `https://pixhawk.org/standards/`. Visited on 2021-03-20.

Meier, L., Honegger, D., and Pollefeys, M. (2015). Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *2015 IEEE international conference on robotics and automation (ICRA)*, 6235–6240.

Ting, J.A., Theodorou, E., and Schaal, S. (2007). A kalman filter for robust outlier detection. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1514–1519.

Yang, Y., Huang, Y., Yang, H., Zhang, T., Wang, Z., and Liu, X. (2021). Real-time terrain-following of an autonomous quadrotor by multi-sensor fusion and control. *Applied Sciences*, 11(3), 1065.