Target Localization using Multiple UAVs with Sensor Fusion via Sigma-Point Kalman Filtering

Gregory L. Plett[*], Pedro DeLima[†], and Daniel J. Pack[‡]

United States Air Force Academy, USAFA, CO 80840, USA

This paper proposes a sensor-fusion methodology based on sigma-point Kalman filtering (SPKF) techniques, where the SPKF is applied to the particular problem of localizing a mobile target using uncertain nonlinear measurements relating to target positions. This SPKF-based sensor fusion is part of our larger research program in which the particular target localization solution of interest must address a number of challenging requirements: (a) covert/passive sensing means must be used; (b) the dynamic characteristics of the target are unknown; (c) the target is episodically mobile; and (d) the target is intermittently occluded from one or more sensors. Our research efforts seek to meet these requirements using a flight of small autonomous unmanned aerial vehicles (UAVs) with heterogeneous sensing capabilities. Each UAV carries one or more sensors, which include: radio frequency sensors, infrared sensors, and image sensors. Due to cost considerations and to payload constraints, it is not desirable for every UAV to have a full complement of sensing capability. In our implementation, the UAVs first search a predefined region for targets. Upon target detection, a small formation of UAVs comprising complementary heterogeneous sensors is autonomously assembled to localize the target. The UAVs then cooperatively locate the target by combining the sensor information collected by heterogeneous sensors onboard the UAVs. The output of the localization process is an estimate of the target's position and velocity as well as error bounds on that estimate.

The primary purpose of this paper is to describe SPKF-based sensor-fusion for target localization using sensors that only provide direction-of-arrival information. Results of simulations are compared to prior work where linear Kalman filters were used to localize stationary targets. The SPKF gives much more accurate and consistent results, due mostly to its better handling of the nonlinear measurement relationship. Results are also given for the application of SPKF to mobile targets.

Nomenclature

AT	Approach Target state of cooperative control state machine
DOA	Direction of arrival
FDOA	Frequency difference of arrival
GS	Global Search state of cooperative control state machine
LT	Locate Target state of cooperative control state machine
RF	Radio frequency
Σ	A covariance matrix corresponding to its subscripted variable
SPKF	Sigma-point Kalman filter
TDOA	Time difference of arrival
TR	Target Re-acquisition state of cooperative control state machine
UAV	Unmanned aerial vehicle
UKF	Unscented Kalman filter
x_k	The true target state at time index k
$\hat{x}_k$	The filter's estimate of the target state at time index k
$\mathcal{X}$	Sigma points corresponding to the target state estimate
y_k	A sensor measurement corresponding to the true target state at time index k
$\mathcal{Y}$	Sigma points corresponding to the target output estimate

[*] Visiting Research Professor, DFEC, Suite 2F6 (Associate Professor from the University of Colorado at Colorado Springs, CO 80918, USA)

[†] Researcher, DFEC, Suite 2F6

[‡] Professor, DFEC, Suite 2F6

I. Introduction

THIS paper proposes a sensor-fusion methodology based on sigma-point Kalman filtering (SPKF) techniques, where the SPKF is applied to the particular problem of localizing a mobile target using uncertain nonlinear measurements relating to target positions. This SPKF-based sensor fusion is part of our larger research program in which the particular target localization solution of our interest must address a number of challenging requirements: (a) covert/passive sensing means must be used (e.g., active radar may not be used); (b) the dynamic characteristics of the target are unknown; (c) the target is intermittently mobile; and (d) the target is intermittently occluded from one or more sensors. Our research efforts seek to meet these requirements using a flight of multiple small autonomous cooperating UAVs with heterogeneous sensing capabilities.[1,2] These UAVs first search a predefined region for targets. Upon target detection, the localization effort begins. When the target is localized by reducing its position estimate uncertainty to an acceptable level, the UAV(s) resume searching for additional targets.

Each UAV carries one or more sensors: radio frequency (RF) sensors to determine direction-of-arrival (DOA), frequency-difference-of-arrival (FDOA), and/or time-difference-of-arrival (TDOA); infrared sensors to detect heat signatures; and image sensors. The UAVs cooperatively locate the target by combining the sensor information collected by heterogeneous sensors onboard the different UAVs. The UAVs collectively estimate the target's position and velocity and compute error bounds on the estimated values.

We achieve cooperative target localization through sensor fusion by using an SPKF[3–5] to intelligently combine measurements to form a dynamic estimate of target position and (optionally) velocity. SPKFs are a generalization of the ubiquitous Kalman filter[6,7] to problems with nonlinear descriptions—an alternate approach might be to use an extended Kalman filter (EKF), but the EKF produces poorer estimates at the same computational complexity as SPKF, so there is no incentive to use it. SPKFs are a superset of methods that include: the unscented Kalman filter (UKF),[8–10] which has been used to locate targets using TDOA measurements,[11] and the central difference Kalman filter (CDKF).[12,13] We use the latter form here since it has slightly higher theoretic accuracy[13] and has fewer algorithm parameters that require tuning.

This paper proceeds by first introducing the cooperative UAV control architecture we apply during the search, detection, and localization tasks. It then shows how the localization may be performed using Kalman-filtering techniques, and introduces the sigma-point Kalman filter. Some practical details such as models that may be used to describe target motion and how to initialize the filtering effort are described next, followed by results and conclusions.

II. Cooperative UAV Control Architecture

We have developed a behavior-based distributed control architecture[2,14] to maximize the capabilities of multiple UAVs to cooperatively search, detect and localize multiple ground targets. The control architecture dictates that each UAV governs its own operation through a decision state machine composed of four states: Global Search (GS), Approach Target (AT), Locate Target (LT) and Target Re-acquisition (TR). Figure 1 illustrates the state machine, where the decision to change from one state to another is represented by directional connectors triggered by the different events summarized in Table 1. We now describe each state in more detail.

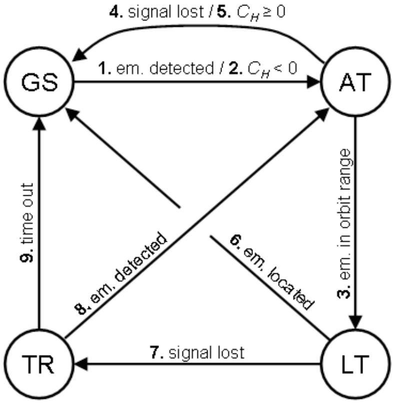


Figure 1. Decision state machine for UAV state selection. The numbered events that trigger each particular directional connector are listed in Table 1.

Table 1. List of events that trigger decisions to change states in the decision state machine of each UAV.

#	From	To	Event
1	GS	AT	New target (e.g., RF emitter) detected by UAV's sensor.
2	GS	AT	Decision to cooperate with an ongoing localization effort.
3	AT	LT	UAV arrives at orbit range from target's estimated position.
4	AT	GS	Target becomes occluded from all available sensors.
5	AT	GS	Decision to cooperate with an ongoing localization effort no longer supported.
6	LT	GS	Target successfully located.
7	LT	TR	Target becomes occluded from all available sensors.
8	TR	AT	Target detected by UAV's sensor.
9	TR	GS	Maximum time for TR reached.

II.A. Global Search (GS)

While in the GS state, a UAV flies with the primary goal of detecting targets not yet found by any other UAV. We assume all UAVs are equipped with at least one sensor with known sensing geometry: e.g., an omni-directional RF sensor with a limited sensing range, or a camera with a limited field of view. The effectiveness of the UAV group is judged by the total area covered by the UAVs for a fixed time period. One must also consider the impact of such search patterns toward individual UAV fuel consumption and the predictability of the UAV paths—an unpredictable search path is desirable when we are dealing with ground targets who are actively trying to remain undetected.

As a distributed control system, cooperative search patterns are generated by having each UAV in the GS state determine its own flying path. Each UAV attempts to fly to destinations within the search boundary that have not been inspected for the longest time, while keeping maximum distances from other UAVs and not turning too much, therefore maximizing the overall search area and minimizing incremental fuel usage. Note that this strategy directly addresses one of the challenges of our research since, after one UAV flies over an area, the interest of other UAVs in revisiting the same location increases over time, allowing for the detection of mobile targets that transmit intermittently.

II.B. Approach Target (AT)

A UAV may change its operational state from GS to AT if it detects a target currently not detected by any other UAVs (Event 1 in Table 1), or if it decides to stop contributing to the global search and join another UAV in a combined target localization effort (Event 2). Approach Target is a state in which the UAV seeks to move into a circular orbit around the estimated position of a detected target at a desired radial distance. If a UAV detects a target, it always switches its operation state to AT in order to position itself to start the target localization effort. If, on the other hand, a target is detected by another UAV, the UAV in question must decide whether to approach the detected target to lend assistance. A UAV decides to help based on considering the distance between itself and the newly detected target, the maximum number of UAVs considered helpful in a localization effort, the number of UAVs presently helping, the number of targets estimated to be present in the search area, and the number of targets that have already been located.

A decision to help may be reversed based on actions taken by other UAVs (Event 5). In this case, the UAV will return to the GS state. A UAV also returns to the GS state if the target of interest becomes occluded from all the sensors it possesses while the UAV is approaching the target (Event 4). The UAV switches its operating state to the LT state when it reaches the orbit (Event 3).

II.C. Locate Target (LT)

A UAV operating in the Locate Target state orbits around the estimated target position, while taking new sensor readings. The current implementation allows up to three UAVs to join in the effort to locate a target, providing simultaneous direction-of-arrival sensor readings from different view-points. The primary purpose of this paper is to propose a Kalman-filter based sensor-fusion process to localize the target by combining information provided by different sensors on different UAVs. In particular, using Kalman filtering techniques, the associated covariance matrix functions as an estimator for the localization error, helping each UAV to determine when the target localization estimate has reached the desired level of confidence. Once a target is accurately located, all UAVs orbiting the target return to the GS state (Event 6). Otherwise, if a target becomes occluded to all available sensors, all UAVs orbiting the target switch to the fourth state, TR (Event 7).

II.D. Target Re-acquisition (TR)

Target Re-acquisition is a state that is only reached by orbiting UAVs operating in LT when their target becomes fully occluded before it is accurately located. UAVs operate in one of two phases within the TR state. During the initial phase, UAVs continue to orbit around the last estimated target position with an increasing orbit radius over time. The resulting flying path is a growing spiral with a rate of expansion dictated by the estimated target's maximum speed. Once a predefined time period has passed without a reacquisition of the target, UAVs operating in TR enter into the second phase.

In the second phase, UAVs perform a local search similar to the global search, except in a limited area around the target's last estimated position. If a target is still not detected after a user-defined maximum amount of time, the associated UAVs discontinue the local search and switch to operate in the GS state (Event 9). A TR effort is successful if during one of the two phases a target is detected. In such circumstance, UAVs switch to the AT state (Event 8).

III. Three Approaches to "Locate Target" Mode

The target we wish to locate may be modeled as having an unknown state that we wish to estimate (e.g., the target's position and/or velocity) using noisy measurements related to that state. Section IV describes how this general problem may be solved using a class of algorithms based on Kalman filtering (KF). Kalman filters maintain a state estimate and a covariance matrix corresponding to the Gaussian error distribution of that state estimate.

For many sensor types, the sensed variable is the result of a nonlinear relationship between the target's state and the UAV's state (e.g., the UAV's state might include its position and orientation). For example, any direction-of-arrival sensor will necessarily compute a trigonometric function of the two; any time-of-arrival or signal-strength indicator will produce a relationship involving radial distance, and so forth. Traditional Kalman filters cannot be used directly since they assume linear measurements, although ad hoc approaches may be taken to linearize the measurements first, and then use the KF. SPKFs, on the other hand, account for nonlinear dynamics and/or nonlinear sensor relationships directly in their theory, so offer the promise of providing much better state estimates.

In this paper we focus on localizing a target using sequential direction-of-arrival (DOA) measurements from the different viewpoints in time and space collected by the formation of UAVs orbiting the target in LT mode. DOA measurements might be collected for an RF emission using a phased-array antenna, or might be sensed as a bearing to the target using a camera-based system (Two cameras can triangulate a target position in three-dimensional space, but a single camera can only provide a bearing to the target). At least three approaches to fusing this data using some form of Kalman filter might be taken by modeling the target with: (1) the filter state being the target's position estimate, and the output equation also being the position estimate; (2) the filter state being the angle to target, and the output equation also being the angle to target; and (3) the filter state being the target's position estimate, and the output equation being the angle to target.

In the first approach (cf. Ref. 15), the state is natural; the position of the target is the desired output of LT mode. The state covariance matrix also provides bounds on the state estimation error. However, measurements must be mapped through some transformation to achieve a position estimate for KF update—this loses the information on sensor uncertainty, and requires complex triangulation methods be used. Another disadvantage of this method is that measurements are not independent; a partial fix is to add artificial process noise to compensate, but this destroys the meaning of the state covariance matrix, and it can no longer be used to provide error bounds on the state.

In the second approach (cf. Ref. 16), the actual sensor-error distribution is taken into account. However the KF state must be constantly updated to reflect a moving UAV and target dynamics are not simple to add (except by increasing process noise). It is also necessary to run several KFs and combine angle results via triangulation to estimate the target's position. Error bounds on the position are not directly known.

In the third approach, taken here, the state is natural, and the output equation is also natural. The sensor uncertainty information may still be used, and the state covariance matrix is a true representation of the error bounds on target position. Target dynamics are easily added, and the solution dynamically scales with the number of UAVs in orbit. However, since DOA is a nonlinear combination of target and UAV positions, a nonlinear KF must be used. The SPKF is one such nonlinear KF.[3–5] Since it may be unfamiliar to the reader, we define the SPKF in the next sections.

IV. Sequential Probabilistic Inference

Generally, any causal dynamic system (*e.g.*, a motion model for the target we wish to localize) generates its outputs as a function of its past and present inputs. Often, we can define a *state vector* for the system whose values together summarize the effect of all past inputs. Present system output is a function of present input and present state only; past input values need not be stored.

For some applications, we desire to estimate the state-vector quantities in real time, as the system operates. *Probabilistic inference* and *optimal estimation theory* are names given to the field of study that concerns estimating these hidden variables in an optimal and consistent fashion, given noisy or incomplete observations. For example, in this paper we will be concerned with estimating the position and velocity of a target. Observations are available to us at sampling points and might include DOA, TDOA, or FDOA measurements.

In the following, we will assume that the motion of the target being localized may be modeled using a discrete-time state-space model of the form

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}, k-1) \tag{1}$$

$$y_k = h(x_k, u_k, v_k, k). \tag{2}$$

Here, $x_k \in \mathbb{R}^n$ is the system state vector at time index k, and Eq. (1) is called the "state equation" or "process equation". The state equation captures the evolving system dynamics. System stability, dynamic controllability and sensitivity to disturbance may all be determined from this equation. The known/deterministic input to the system is $u_k \in \mathbb{R}^p$, and $w_k \in \mathbb{R}^n$ is stochastic "process noise" or "disturbance" that models some unmeasured input which affects the state of the system. The output of the system is $y_k \in \mathbb{R}^m$, computed by the "output equation" (2) as a function of the states, input, and $v_k \in \mathbb{R}^m$, which models "sensor noise" that affects the measurement of the system output in a memoryless way, but does not affect the system state. $f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)$ is a (possibly nonlinear) state transition function and $h(x_k, u_k, v_k, k)$ is a (possibly nonlinear) measurement function.

With this model structure, the evolution of unobserved states and observed measurements may be visualized as shown in Fig. 2. The conditional probability $p(x_k \mid x_{k-1})$ indicates that the new state is a function of not only the deterministic input u_{k-1}, but also the stochastic input w_{k-1}, so that the unobserved state variables do not form a deterministic sequence. Similarly, the conditional probability density function $p(y_k \mid x_k)$ indicates that the observed output is not a deterministic function of the state, due to the stochastic input v_k.

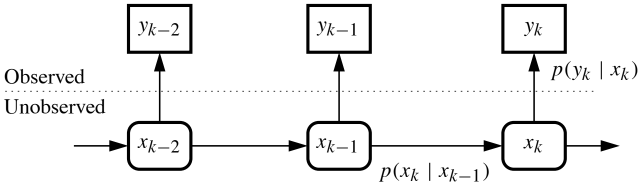


Figure 2. Sequential probabilistic inference.

The goal of probabilistic inference is to create an estimate of the system state given all observations $\mathbb{Y}_k = \{y_0, y_1, \cdots, y_k\}$. A frequently used estimator is the conditional mean

$$\hat{x}_k = \mathbb{E}\left[x_k \mid \mathbb{Y}_k\right] = \int_{R_{x_k}} x_k p(x_k \mid \mathbb{Y}_k)\, dx_k,$$

where R_{x_k} is the set comprising the range of possible x_k, and $\mathbb{E}[\cdot]$ is the statistical expectation operator. The optimal solution to this problem computes the posterior probability density $p(x_k \mid \mathbb{Y}_k)$ recursively with two steps per iteration.[4] The first step computes probabilities for predicting x_k given all past observations

$$p(x_k \mid \mathbb{Y}_{k-1}) = \int_{R_{x_{k-1}}} p(x_k \mid x_{k-1}) p(x_{k-1} \mid \mathbb{Y}_{k-1})\, dx_{k-1},$$

and the second step updates the prediction via

$$p(x_k \mid \mathbb{Y}_k) = \frac{p(y_k \mid x_k) p(x_k \mid \mathbb{Y}_{k-1})}{p(y_k \mid \mathbb{Y}_{k-1})},$$

which is a simple application of Bayes' rule and the assumption that the present observation y_k is conditionally independent of previous measurements given the present state x_k. The relevant probabilities may be computed as

$$p(y_k \mid \mathbb{Y}_{k-1}) = \int_{R_{x_k}} p(y_k \mid x_k) p(x_k \mid \mathbb{Y}_{k-1})\, dx_k$$

$$p(x_k \mid x_{k-1}) = \sum_{\{w\,:\,x_k = f(x_{k-1}, u_{k-1}, w, k-1)\}} p(w)$$
$$p(y_k \mid x_k) = \sum_{\{v\,:\,y_k = h(x_k, u_k, v, k)\}} p(v).$$

Although this is the optimal solution, finding a formula solving the multi-dimensional integrals in a closed-form is intractable for most real-world systems. For applications that justify the computational expense, the integrals may be closely approximated using Monte Carlo methods such as particle filters.[4, 17–20] However, for our application, we do not believe that present economics make this approach feasible.

A simplified solution to these equations may be obtained if we are willing to make the assumption that all probability densities are Gaussian—this is the basis of the original Kalman filter, the extended Kalman filter, and the sigma-point Kalman filters to be discussed. Then, rather than having to propagate the entire density function through time, we need only to evaluate the conditional mean and covariance of the state (and parameters, perhaps) once each sampling interval. It can be shown that the recursion becomes:[21]

$$\hat{x}_k^+ = \hat{x}_k^- + L_k\big(y_k - \hat{y}_k\big) = \hat{x}_k^- + L_k \tilde{y}_k \tag{3}$$
$$\Sigma_{\tilde{x},k}^+ = \Sigma_{\tilde{x},k}^- - L_k \Sigma_{\tilde{y},k} L_k^T, \tag{4}$$

where the superscript T is the matrix/vector transpose operator, and

$$\hat{x}_k^+ = \mathbb{E}\big[x_k \mid \mathbb{Y}_k\big] \tag{5}$$
$$\hat{x}_k^- = \mathbb{E}\big[x_k \mid \mathbb{Y}_{k-1}\big] \tag{6}$$
$$\hat{y}_k = \mathbb{E}\big[y_k \mid \mathbb{Y}_{k-1}\big] \tag{7}$$
$$\Sigma_{\tilde{x},k}^- = \mathbb{E}\big[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T\big] = \mathbb{E}\big[(\tilde{x}_k^-)(\tilde{x}_k^-)^T\big] \tag{8}$$
$$\Sigma_{\tilde{x},k}^+ = \mathbb{E}\big[(x_k - \hat{x}_k^+)(x_k - \hat{x}_k^+)^T\big] = \mathbb{E}\big[(\tilde{x}_k^+)(\tilde{x}_k^+)^T\big] \tag{9}$$
$$\Sigma_{\tilde{y},k} = \mathbb{E}\big[(y_k - \hat{y}_k)(y_k - \hat{y}_k)^T\big] = \mathbb{E}\big[(\tilde{y}_k)(\tilde{y}_k)^T\big] \tag{10}$$
$$L_k = \mathbb{E}\big[(x_k - \hat{x}_k^-)(y_k - \hat{y}_k)^T\big]\Sigma_{\tilde{y},k}^{-1} = \Sigma_{\tilde{x}\tilde{y},k}^- \Sigma_{\tilde{y},k}^{-1}. \tag{11}$$

While this is a linear recursion, we have not directly assumed that the system model is linear. In the notation we use, the decoration "circumflex" indicates an estimated quantity (*e.g.,* $\hat{x}$ indicates an estimate of the true quantity x). A superscript "$-$" indicates an *a priori* estimate (*i.e.,* a prediction of a quantity's present value based on past data) and a superscript "$+$" indicates an *a posteriori* estimate (*e.g.,* $\hat{x}_k^+$ is the estimate of true quantity x at time index k based on all measurements taken up to and including time k). The decoration "tilde" indicates the error of an estimated quantity. The symbol $\Sigma_{xy} = \mathbb{E}[xy^T]$ indicates the auto- or cross-correlation of the variables in its subscript. (Note that often these variables are zero-mean, so the correlations are identical to covariances). Also, for brevity of notation, we often use Σ_x to indicate the same quantity as Σ_{xx}.

Equations (3) through (11) and approximations thereof can be used to derive the Kalman filter, the extended Kalman filter, and sigma-point Kalman filters.[21] All members of this family of filters comply with a structured sequence of six steps per iteration, as outlined here:

GENERAL STEP 1: STATE ESTIMATE TIME UPDATE. For each measurement interval, the first step is to compute an updated prediction of the present value of x_k, based on *a priori* information and the system model. This is done using Eqs. (1) and (6) as

$$\hat{x}_k^- = \mathbb{E}\big[x_k \mid \mathbb{Y}_{k-1}\big] = \mathbb{E}\big[f(x_{k-1}, u_{k-1}, w_{k-1}, k-1) \mid \mathbb{Y}_{k-1}\big].$$

GENERAL STEP 2: ERROR COVARIANCE TIME UPDATE. The second step is to determine the predicted state-estimate error covariance matrix $\Sigma_{\tilde{x},k}^-$ based on *a priori* information and the system model. We compute $\Sigma_{\tilde{x},k}^- = \mathbb{E}\big[(\tilde{x}_k^-)(\tilde{x}_k^-)^T\big]$ using Eq. (8), knowing that $\tilde{x}_k^- = x_k - \hat{x}_k^-$.

GENERAL STEP 3: ESTIMATE SYSTEM OUTPUT y_k. The third step is to estimate the system's output using present *a priori* information and Eqs. (2) and (7)

$$\hat{y}_k = \mathbb{E}\big[y_k \mid \mathbb{Y}_{k-1}\big] = \mathbb{E}\big[h(x_k, u_k, v_k, k) \mid \mathbb{Y}_{k-1}\big].$$

GENERAL STEP 4: ESTIMATOR GAIN MATRIX L_k. The fourth step is to compute the estimator gain matrix L_k by evaluating $L_k = \Sigma^{-}_{\tilde{x}\tilde{y},k}\Sigma^{-1}_{\tilde{y},k}$.

GENERAL STEP 5: STATE ESTIMATE MEASUREMENT UPDATE. The fifth step is to compute the *a posteriori* state estimate by updating the *a priori* estimate using the estimator gain and the output prediction error $y_k - \hat{y}_k$ using (3). There is no variation in this step in the different Kalman filter methods; implementational differences between Kalman approaches do manifest themselves in all other steps, however.

GENERAL STEP 6: ERROR COVARIANCE MEASUREMENT UPDATE. The final step computes the *a posteriori* error covariance matrix using Eq. (4). The estimator output comprises the state estimate $\hat{x}^+_k$ and error covariance estimate $\Sigma^+_{\tilde{x},k}$. The estimator then waits until the next sample interval, updates k, and proceeds to step 1.

The general sequential probabilistic inference solution is summarized in Table 2. The following sections describe specific applications of this general framework.

Table 2. Summary of the general sequential probabilistic inference solution.

General state-space model:

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)$$
$$y_k = h(x_k, u_k, v_k, k),$$

where w_k and v_k are independent zero-mean white Gaussian noise processes of covariance matrices Σ_w and Σ_v, respectively.

Definitions: Let

$$\tilde{x}^-_k = x_k - \hat{x}^-_k, \qquad \tilde{y}_k = y_k - \hat{y}_k.$$

Initialization: For $k = 0$, set

$$\hat{x}^+_0 = \mathbb{E}\left[x_0\right]$$
$$\Sigma^+_{\tilde{x},0} = \mathbb{E}\left[(x_0 - \hat{x}^+_0)(x_0 - \hat{x}^+_0)^T\right].$$

Computation: For $k = 1, 2, \ldots$ compute:

State estimate time update: $\hat{x}^-_k = \mathbb{E}\left[f(x_{k-1}, u_{k-1}, w_{k-1}, k-1) \mid \mathbb{Y}_{k-1}\right].$

Error covariance time update: $\Sigma^-_{\tilde{x},k} = \mathbb{E}\left[(\tilde{x}^-_k)(\tilde{x}^-_k)^T\right].$

Output estimate: $\hat{y}_k = \mathbb{E}\left[h(x_k, u_k, v_k, k) \mid \mathbb{Y}_{k-1}\right].$

Estimator gain matrix: $L_k = \mathbb{E}\left[(\tilde{x}^-_k)(\tilde{y}_k)^T\right]\left(\mathbb{E}\left[(\tilde{y}_k)(\tilde{y}_k)^T\right]\right)^{-1}.$

State estimate measurement update: $\hat{x}^+_k = \hat{x}^-_k + L_k\left(y_k - \hat{y}_k\right).$

Error covariance measurement update: $\Sigma^+_{\tilde{x},k} = \Sigma^-_{\tilde{x},k} - L_k \Sigma_{\tilde{y},k} L^T_k.$

V. Sigma-Point Kalman Filters

The extended Kalman filter is probably the best known and most widely used nonlinear Kalman filter. However, it has a number of shortcomings that affect the accuracy of state estimation. These problems reside in two assumptions made in order to propagate a Gaussian random state vector x through some nonlinear function: one assumption concerns the calculation of the output random variable mean, the other concerns the output random variable covariance.

First, we note that in step 1, the EKF attempts to determine an output random-variable mean from the state-transition function $f(\cdot)$ assuming that the input state is a Gaussian random variable. EKF step 3 makes a similar calculation for the output function $h(\cdot)$. The EKF makes the simplification

$$\mathbb{E}\left[\text{fn}(x)\right] \approx \text{fn}(\mathbb{E}\left[x\right]),$$

which is not true in general. The SPKF to be described will make an improved approximation to the means in steps 1 and 3.

Secondly, in EKF steps 2 and 4, a Taylor-series expansion is performed as part of a calculation designed to find the output-variable covariance. Nonlinear terms are dropped from the expansion, resulting in a loss of accuracy.

Sigma-point Kalman filtering (SPKF) is an alternate approach to generalizing the Kalman filter to state estimation for nonlinear systems. Rather than using Taylor-series expansions to approximate the required covariance matrices, a number of function evaluations are performed whose results are used to compute an estimated covariance matrix. This has several advantages: (1) derivatives do not need to be computed (which is one of the most error-prone steps when implementing EKF), also implying (2) the original functions do not need to be differentiable, and (3) better covariance approximations are usually achieved, relative to EKF, allowing for better state estimation, (4) all with comparable computational complexity to EKF.

SPKF estimates the mean and covariance of the output of a nonlinear function using a small fixed number of function evaluations. A set of points (*sigma points*) is chosen to be input to the function so that the (possibly weighted) mean and covariance of the points exactly match the mean and covariance of the *a priori* random variable being modeled. These points are then passed through the nonlinear function, resulting in a transformed set of points. The *a posteriori* mean and covariance that are sought are then approximated by the mean and covariance of these points. Note that the sigma points comprise a fixed small number of vectors that are calculated deterministically—not like the Monte Carlo or particle filter methods.

Specifically, if the input random vector x has dimension L, mean $\bar{x}$, and covariance $\Sigma_{\tilde{x}}$, then $p + 1 = 2L + 1$ sigma points are generated as the set

$$\mathcal{X} = \left\{\bar{x}, \bar{x} + \gamma\sqrt{\Sigma_{\tilde{x}}}, \bar{x} - \gamma\sqrt{\Sigma_{\tilde{x}}}\right\},$$

with columns of $\mathcal{X}$ indexed from 0 to p, and where the matrix square root $S = \sqrt{\Sigma}$ computes a result such that $\Sigma = SS^T$. Usually, the efficient *Cholesky decomposition*[22,23] is used, resulting in lower-triangular S. The reader can verify that the weighted mean and covariance of $\mathcal{X}$ equal the original mean and covariance of random vector x for a specific set of $\{\gamma, \alpha^{(\mathrm{m})}, \alpha^{(\mathrm{c})}\}$ if we define the weighted mean as $\bar{x} = \sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} \mathcal{X}_i$, the weighted covariance as $\Sigma_{\tilde{x}} = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{X}_i - \bar{x})(\mathcal{X}_i - \bar{x})^T$, $\mathcal{X}_i$ as the ith column of $\mathcal{X}$, and both $\alpha_i^{(m)}$ and $\alpha_i^{(\mathrm{c})}$ as real scalars with the necessary (but not sufficient) conditions that $\sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} = 1$ and $\sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} = 1$. The various sigma-point methods differ only in the choices taken for these weighting constants. Values for the two most common methods—the unscented Kalman filter (UKF)[8–10,20,24,25] and the central difference Kalman filter (CDKF)[4,12,13]—are summarized in Table 3. The UKF is derived from the point of view of estimating covariances with data rather than a Taylor series. The CDKF is derived quite differently—it uses Stirling's formula to approximate derivatives rather than using a Taylor series—but the final method is essentially identical. The CDKF has only one "tuning parameter" h, which makes implementation simpler. It also has marginally higher theoretic accuracy than UKF,[13] so we focus on this method in the application sections later.

Table 3. Weighting constants for two sigma-point methods.

	γ	$\alpha_0^{(\mathrm{m})}$	$\alpha_k^{(\mathrm{m})}$	$\alpha_0^{(\mathrm{c})}$	$\alpha_k^{(\mathrm{c})}$
UKF	$\sqrt{L+\lambda}$	$\frac{\lambda}{L+\lambda}$	$\frac{1}{2(L+\lambda)}$	$\frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta)$	$\frac{1}{2(L+\lambda)}$
CDKF	h	$\frac{h^2-L}{h^2}$	$\frac{1}{2h^2}$	$\frac{h^2-L}{h^2}$	$\frac{1}{2h^2}$

$\lambda = \alpha^2(L + \kappa) - L$ is a scaling parameter, with $(10^{-2} \le \alpha \le 1)$.
Note that this α is different from $\alpha^{(\mathrm{m})}$ and $\alpha^{(\mathrm{c})}$. κ is either 0 or $3 - L$.
β incorporates prior information. For Gaussian RVs, $\beta = 2$.
h may take any positive value. For Gaussian RVs, $h = \sqrt{3}$.

To use SPKF in an estimation problem, we first define an augmented random vector x^a that combines the randomness of the state, process noise, and sensor noise. This augmented vector is used in the estimation process as described below.

SPKF STEP 1: STATE ESTIMATE TIME UPDATE. For each measurement interval, the state estimate time update is computed by first forming the augmented *a posteriori* state estimate vector for the previous time interval: $\hat{x}_{k-1}^{a,+} = \left[(\hat{x}_{k-1}^{+})^T, \bar{w}, \bar{v}\right]^T$, and the augmented *a posteriori* covariance estimate: $\Sigma_{\tilde{x},k-1}^{a,+} = \mathrm{diag}\left(\Sigma_{\tilde{x},k-1}^{+}, \Sigma_w, \Sigma_v\right)$. These factors are used to generate the $p + 1$ sigma points

$$\mathcal{X}_{k-1}^{a,+} = \left\{\hat{x}_{k-1}^{a,+}, \hat{x}_{k-1}^{a,+} + \gamma\sqrt{\Sigma_{\tilde{x},k-1}^{a,+}}, \hat{x}_{k-1}^{a,+} - \gamma\sqrt{\Sigma_{\tilde{x},k-1}^{a,+}}\right\}.$$

From the augmented sigma points, the $p+1$ vectors comprising the state portion $\mathcal{X}^{x,+}_{k-1}$ and the $p+1$ vectors comprising the process-noise portion $\mathcal{X}^{w,+}_{k-1}$ are extracted. The process equation is evaluated using all pairs of $\mathcal{X}^{x,+}_{k-1,i}$ and $\mathcal{X}^{w,+}_{k-1,i}$ (where the subscript i denotes that the ith vector is being extracted from the original set), yielding the *a priori* sigma points $\mathcal{X}^{x,-}_{k,i}$ for time step k. That is,

$$\mathcal{X}^{x,-}_{k,i} = f(\mathcal{X}^{x,+}_{k-1,i}, u_{k-1}, \mathcal{X}^{w,+}_{k-1,i}, k-1).$$

Finally, the *a priori* state estimate is computed as

$$\begin{aligned}\hat{x}^-_k &= \mathbb{E}\big[f(x_{k-1}, u_{k-1}, w_{k-1}, k-1) \mid \mathbb{Y}_{k-1}\big] \\ &\approx \sum_{i=0}^{p} \alpha^{(\mathrm{m})}_i f(\mathcal{X}^{x,+}_{k-1,i}, u_{k-1}, \mathcal{X}^{w,+}_{k-1,i}, k-1) \\ &= \sum_{i=0}^{p} \alpha^{(\mathrm{m})}_i \mathcal{X}^{x,-}_{k,i}.\end{aligned}$$

SPKF STEP 2: ERROR COVARIANCE TIME UPDATE. Using the *a priori* sigma points from step 1, the *a priori* covariance estimate is computed as

$$\Sigma^-_{\tilde{x},k} = \sum_{i=0}^{p} \alpha^{(\mathrm{c})}_i \big(\mathcal{X}^{x,-}_{k,i} - \hat{x}^-_k\big)\big(\mathcal{X}^{x,-}_{k,i} - \hat{x}^x_{k,i}\big)^T.$$

SPKF STEP 3: ESTIMATE SYSTEM OUTPUT y_k. The system output is estimated by evaluating the model output equation using the sigma points describing the spread in the state and noise vectors. First, we compute the points $\mathcal{Y}_{k,i} = h(\mathcal{X}^{x,-}_{k,i}, u_k, \mathcal{X}^{v,+}_{k-1,i}, k)$. The output estimate is then

$$\begin{aligned}\hat{y}_k &= \mathbb{E}\big[h(x_k, u_k, v_k, k) \mid \mathbb{Y}_{k-1}\big] \\ &\approx \sum_{i=0}^{p} \alpha^{(\mathrm{m})}_i h(\mathcal{X}^{x,-}_{k,i}, u_k, \mathcal{X}^{v,+}_{k-1,i}, k) \\ &= \sum_{i=0}^{p} \alpha^{(\mathrm{m})}_i \mathcal{Y}_{k,i}.\end{aligned}$$

SPKF STEP 4: ESTIMATOR GAIN MATRIX L_k. To compute the estimator gain matrix, we must first compute the required covariance matrices.

$$\begin{aligned}\Sigma_{\tilde{y},k} &= \sum_{i=0}^{p} \alpha^{(\mathrm{c})}_i \big(\mathcal{Y}_{k,i} - \hat{y}_k\big)\big(\mathcal{Y}_{k,i} - \hat{y}_k\big) \\ \Sigma^-_{\tilde{x}\tilde{y},k} &= \sum_{i=0}^{p} \alpha^{(\mathrm{c})}_i \big(\mathcal{X}^{x,-}_{k,i} - \hat{x}^-_k\big)\big(\mathcal{Y}_{k,i} - \hat{y}_k\big).\end{aligned}$$

Then, we simply compute $L_k = \Sigma^-_{\tilde{x}\tilde{y},k}\Sigma^{-1}_{\tilde{y},k}$.

SPKF STEP 6: ERROR COVARIANCE MEASUREMENT UPDATE. The final step is calculated directly from the optimal formulation: $\Sigma^+_{\tilde{x},k} = \Sigma^-_{\tilde{x},k} - L_k\Sigma_{\tilde{y},k}L^T_k$. The SPKF solution is summarized in Table 4.

VI. Example Models of Target Motion

In order to use any Kalman filtering technique to localize a ground target, we require a model of the target's dynamics. Due to the non-cooperative nature of the target assumed in our present research, this model cannot be known a priori; therefore, we must employ an approximate model. For example, we might assume that the target is nearly stationary and decide to employ a "nearly constant position" (NCP) model. Alternately, we might assume that

Table 4. Summary of the nonlinear sigma-point Kalman filter.

Nonlinear state-space model:

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}, w_{k-1}, k-1) \\ y_k &= h(x_k, u_k, v_k, k), \end{aligned}$$

where w_k and v_k are independent zero-mean white Gaussian noise processes of covariance matrices Σ_w and Σ_v, respectively.

Definitions: Let

$$x_k^a = \left[x_k^T,\ w_k^T,\ v_k^T\right]^T, \quad \mathcal{X}_k^a = \left[(\mathcal{X}_k^x)^T,\ (\mathcal{X}_k^w)^T,\ (\mathcal{X}_k^v)^T\right]^T, \quad p = 2 \times \dim(x_k^a).$$

Initialization: For $k = 0$, set

$$\begin{aligned} \hat{x}_0^+ &= \mathbb{E}\left[x_0\right] & \hat{x}_0^{a,+} &= \mathbb{E}\left[x_0^a\right] = \left[(\hat{x}_0^+)^T,\ \bar{w},\ \bar{v}\right]^T. \\ \Sigma_{\tilde{x},0}^+ &= \mathbb{E}\left[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T\right] & \Sigma_{\tilde{x},0}^{a,+} &= \mathbb{E}\left[(x_0^a - \hat{x}_0^{a,+})(x_0^a - \hat{x}_0^{a,+})^T\right] \\ & & &= \operatorname{diag}\left(\Sigma_{\tilde{x},0}^+,\ \Sigma_w,\ \Sigma_v\right). \end{aligned}$$

Computation: For $k = 1, 2, \ldots$ compute:

State estimate time update:

$$\begin{aligned} \mathcal{X}_{k-1}^{a,+} &= \left\{\hat{x}_{k-1}^{a,+},\ \hat{x}_{k-1}^{a,+} + \gamma\sqrt{\Sigma_{\tilde{x},k-1}^{a,+}},\ \hat{x}_{k-1}^{a,+} - \gamma\sqrt{\Sigma_{\tilde{x},k-1}^{a,+}}\right\}. \\ \mathcal{X}_{k,i}^{x,-} &= f(\mathcal{X}_{k-1,i}^{x,+}, u_{k-1}, \mathcal{X}_{k-1,i}^{w,+}, k-1). \\ \hat{x}_k^- &= \textstyle\sum_{i=0}^p \alpha_i^{(\mathrm{m})} \mathcal{X}_{k,i}^{x,-}. \end{aligned}$$

Error covariance time update:

$$\Sigma_{\tilde{x},k}^- = \textstyle\sum_{i=0}^p \alpha_i^{(\mathrm{c})} \left(\mathcal{X}_{k,i}^{x,-} - \hat{x}_k^-\right)\left(\mathcal{X}_{k,i}^{x,-} - \hat{x}_k^-\right)^T.$$

Output estimate:

$$\begin{aligned} \mathcal{Y}_{k,i} &= h(\mathcal{X}_{k,i}^{x,-}, u_k, \mathcal{X}_{k-1,i}^{v,+}, k). \\ \hat{y}_k &= \textstyle\sum_{i=0}^p \alpha_i^{(\mathrm{m})} \mathcal{Y}_{k,i}. \end{aligned}$$

Estimator gain matrix:

$$\begin{aligned} \Sigma_{\tilde{y},k} &= \textstyle\sum_{i=0}^p \alpha_i^{(\mathrm{c})} \left(\mathcal{Y}_{k,i} - \hat{y}_k\right)\left(\mathcal{Y}_{k,i} - \hat{y}_k\right)^T. \\ \Sigma_{\tilde{x}\tilde{y},k}^- &= \textstyle\sum_{i=0}^p \alpha_i^{(\mathrm{c})} \left(\mathcal{X}_{k,i}^{x,-} - \hat{x}_k^-\right)\left(\mathcal{Y}_{k,i} - \hat{y}_k\right)^T. \\ L_k &= \Sigma_{\tilde{x}\tilde{y},k}^- \Sigma_{\tilde{y},k}^{-1}. \end{aligned}$$

State estimate measurement update:

$$\hat{x}_k^+ = \hat{x}_k^- + L_k\left(y_k - \hat{y}_k\right).$$

Error covariance measurement update:

$$\Sigma_{\tilde{x},k}^+ = \Sigma_{\tilde{x},k}^- - L_k \Sigma_{\tilde{y},k} L_k^T.$$

the target is mobile and decide to use a “nearly constant velocity” (NCV) model of dynamics. These two models are now introduced.

The NCP model uses state vector $x(t) = [p_x(t), p_y(t)]^T$, where $p_x(t)$ is the “x” position coordinate of the target, and $p_y(t)$ is the “y” position coordinate of the target on the ground. The state dynamics are then:

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} x(t) + w(t) \\ y(t) &= h(x(t), u(t)) + v(t), \end{aligned}$$

where the stochastic signals $w(t)$ and $v(t)$ are assumed to be Gaussian and white, and sensor noise $v(t)$ has covariance matrix Σ_v and process noise $w(t)$ has covariance matrix $\Sigma_w(t)$. One example for the process noise covariance matrix is $\Sigma_w(t) = \operatorname{diag}(\sigma^2, \sigma^2)$, assuming equal noise for both the “x” velocity and “y” velocity. The output equation depends on $h(\cdot)$, which itself depends on the sensor being used to produce a measurement and perhaps a measurable input signal $u(t)$. This model says, in effect, that the target position is generally constant except via perturbations to its velocity through $w(t)$, and that measurements may be taken that somehow relate to the position and velocity of the target. For a direction-of-arrival sensor, $h(t) = \pi + \operatorname{atan2}(y_{\mathrm{uav}}(t) - p_y(t), x_{\mathrm{uav}}(t) - p_x(t))$, where angles are assumed measured in a Cartesian sense (due East is zero angle).

The NCV model uses state vector $x(t) = [p_x(t), p_y(t), v_x(t), v_y(t)]^T$, where $p_x(t)$ and $p_y(t)$ are defined as in the NCP model, $v_x(t)$ is the velocity of the target along the “x” axis, and $v_y(t)$ is the velocity of the target along the “y”

axis:

$$\begin{aligned}\dot{x}(t) &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x(t) + w(t) \\ y(t) &= h(x(t), u(t)) + v(t),\end{aligned}$$

where the stochastic signals $w(t)$ and $v(t)$ are assumed to be Gaussian and white, and sensor noise $v(t)$ has covariance matrix Σ_v and process noise $w(t)$ has covariance matrix $\Sigma_w(t)$. In this paper, we assume $\Sigma_w(t) = \text{diag}(0, 0, \sigma^2, \sigma^2)$. The output equation is the same as for the NCP model. The NCV model says, in effect, that the target velocity is generally constant except via perturbations to its acceleration through $w(t)$, and that measurements may be taken that somehow relate to the position and velocity of the target. The agility of the target is modeled by selecting appropriate values of σ^2 for $\Sigma_w(t)$.

We will be updating the Kalman filter at regularly separated discrete points in time. Therefore, we need to be able to integrate the effect of $w(t)$ on $x(t)$ over a period t_0 (the integral essentially performing a convolution) and result in a model of the form:

$$\begin{aligned}x(t + t_0) &= A_{t_0} x(t) + w_{t_0}(t) \\ y(t) &= h(x(t), u(t)) + v(t).\end{aligned}$$

Note that the output equation is unchanged. We compute $A_{t_0} = e^{At_0}$, where $e^{(\cdot)}$ is the matrix-exponential function. We further compute[26]

$$\Sigma_{w_{t_0}} = \int_0^{t_0} e^{A(t_0 - \tau)} \Sigma_w e^{A^T(t_0 - \tau)} \, d\tau$$

to evaluate the equivalent discrete-time noise covariance based on the continuous-time noise covariance. For the NCP model we have

$$x(t + t_0) = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{A_{t_0}} x(t) + w_{t_0}(t), \qquad \text{where} \qquad \Sigma_{w_{t_0}} = \begin{bmatrix} \frac{\sigma^2}{2}\left(e^{2t_0} - 1\right) & 0 \\ 0 & \frac{\sigma^2}{2}\left(e^{2t_0} - 1\right) \end{bmatrix}.$$

For the NCV model we have

$$x(t + t_0) = \Sigma_{w_{t_0}} \underbrace{\begin{bmatrix} 1 & 0 & t_0 & 0 \\ 0 & 1 & 0 & t_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A_{t_0}} x(t) + w_{t_0}(t), \qquad \text{where} \qquad \Sigma_{w_{t_0}} = \begin{bmatrix} \frac{t_0^3\sigma^2}{3} & 0 & \frac{t_0^2\sigma^2}{2} & 0 \\ 0 & \frac{t_0^3\sigma^2}{3} & 0 & \frac{t_0^2\sigma^2}{2} \\ \frac{t_0^2\sigma^2}{2} & 0 & t_0\sigma^2 & 0 \\ 0 & \frac{t_0^2\sigma^2}{2} & 0 & t_0\sigma^2 \end{bmatrix}.$$

VII. Initializing the SPKF State

The SPKF algorithm requires that we determine the initial state estimate $\hat{x}_0^+ = \mathbb{E}[x_0]$ and the initial covariance estimate $\Sigma_{\tilde{x},0}^+ = \mathbb{E}[(x - \hat{x}_0^+)(x - \hat{x}_0^+)^T]$. This section describes how to do this with a single UAV, and an extension to the method that allows measurements from multiple UAVs to estimate these values.

For one UAV, if the only sensor reading available is the direction of arrival, there is a great deal of uncertainty in the actual target position due to a lack of information regarding the range to the target. We model the range to target as the random variable R, and the angle to the target as random variable Θ. Then,

$$\begin{aligned}p_x &= R\cos\Theta + x_{\text{uav}} \\ p_y &= R\sin\Theta + y_{\text{uav}},\end{aligned}$$

where $x = [p_x, p_y]^T$ target position (as in the NCP model) and $[x_{\text{uav}}, y_{\text{uav}}]^T$ is the UAV position. (The target position aspect of the NCV model is handled in the same way, and the target velocity estimate is generally initialized to zero).

We assume a uniform distribution on $R \sim \mathcal{U}(0, r_0)$, where r_0 is the sensor range. We model the sensor reading $\hat{\Theta} = \Theta + \Theta_{\text{noise}}$ where Θ_{noise} is a Gaussian distribution with zero mean and standard deviation σ_v known by the sensor. Then, assuming that R and Θ_{noise} are independent,

$$\hat{x}_0^+ = \mathbb{E}\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \mathbb{E}[R]\mathbb{E}\begin{bmatrix} \cos(\hat{\Theta} - \Theta_{\text{noise}}) \\ \sin(\hat{\Theta} - \Theta_{\text{noise}}) \end{bmatrix} + \begin{bmatrix} x_{\text{uav}} \\ y_{\text{uav}} \end{bmatrix}.$$

Without loss of generality, we can assume that the sensor reading $\hat{\Theta} = 0$, and then rotate the final result by the true sensor reading to compensate. For the above assumptions, the final answer is:

$$\hat{x}_0^+ = \frac{r_0 \exp(-\sigma_v^2/2)}{2}\begin{bmatrix} \cos\hat{\Theta} \\ \sin\hat{\Theta} \end{bmatrix} + \begin{bmatrix} x_{\text{uav}} \\ y_{\text{uav}} \end{bmatrix}.$$

Using similar reasoning, the covariance matrix may be found to be:

$$\Sigma_{\tilde{x},0}^+ = \frac{r_0^2}{24}\begin{bmatrix} \cos(\hat{\Theta}) & -\sin(\hat{\Theta}) \\ \sin(\hat{\Theta}) & \cos(\hat{\Theta}) \end{bmatrix}\begin{bmatrix} 4 + 4\exp(-2\sigma_v^2) - 6\exp(-\sigma_v^2) & 0 \\ 0 & 1 - \exp(-2\sigma_v^2) \end{bmatrix}\begin{bmatrix} \cos(\hat{\Theta}) & \sin(\hat{\Theta}) \\ -\sin(\hat{\Theta}) & \cos(\hat{\Theta}) \end{bmatrix}.$$

The result is illustrated in the drawing of Fig. 3 (not to scale). The true uncertainty region for the target is the gray wedge shape, and the approximate uncertainty region is indicated by the ellipse denoting a constant-probability contour of the Gaussian distribution with mean $\hat{x}_0^+$ and covariance $\Sigma_{\tilde{x},0}^+$.

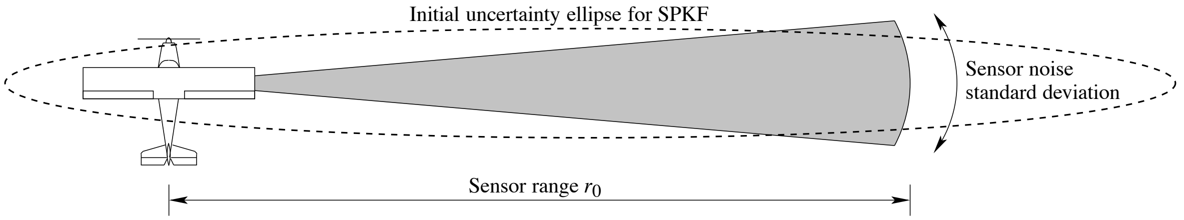

Initial uncertainty ellipse for SPKF
Sensor noise standard deviation
Sensor range r_0

Figure 3. Initializing the SPKF state vector.

When multiple UAVs simultaneously detect a target, we must modify this procedure to take into account the totality of the measurements taken[a]. Our approach is to first compute a separate state estimate and covariance estimate for each UAV's sensor reading, and to then combine the resulting estimates and covariances in a weighted least-squares sense. We illustrate the result for three initial state estimates and corresponding covariance matrices. The state estimates at time 0 from UAVs 1, 2, and 3 (respectively) are denoted $\hat{x}_{0:1}^+$, $\hat{x}_{0:2}^+$, and $\hat{x}_{0:3}^+$. The covariance matrices are denoted as $\Sigma_{\tilde{x},0:1}^+$, $\Sigma_{\tilde{x},0:2}^+$, and $\Sigma_{\tilde{x},0:3}^+$. Define $X = [\hat{x}_{0:1}^+, \hat{x}_{0:2}^+, \hat{x}_{0:3}^+]^T$, $W = \text{diag}(\Sigma_{\tilde{x},0:1}^+, \Sigma_{\tilde{x},0:2}^+, \Sigma_{\tilde{x},0:3}^+)$, and $H = [I_2, I_2, I_2]^T$ where I_2 is the 2×2 identity matrix. Then, we have,

$$\begin{aligned} \hat{x}_0^+ &= (H^T W^{-1} H)^{-1} H^T W^{-1} X \\ \Sigma_{\tilde{x},0}^+ &= (H^T W^{-1} H)^{-1}. \end{aligned}$$

VIII. Results

In order to generate results of target localization using SPKF, we used the USAFA Multiple-UAV Simulation System. This is the same program that was used to generate the results in Ref. 16, so we believe the results to be directly comparable. For the purpose of clarity, the simulator was run only in the LT mode, with the targets continuously emitting a detectable signal, and with the UAV positions initialized by randomly placing them within a 5 km radius of the target, and sensor ranges set to 5 km. All UAVs are assumed to be outfitted with global-positioning-system units, whose inaccuracy was not considered in the simulations. The UAV flight paths were then controlled to converge to an orbit of 2 km distance from the target, with inter-UAV angles of 90° for a two-UAV simulation and ±120° for a three-UAV simulation. Nominal UAV velocity was 100 km/h. Sensor measurements were simulated and fused once per ten seconds (of simulated time) for stationary-target simulations and once per second for moving-target

[a] This is admittedly a rare case, but may occur in practice if a target becomes visible when several UAVs are in range, which is the situation considered in the results section below to eliminate the effect of startup transients caused by UAVs requiring variable time to fly within sensor range.

simulations. For the mobile-target scenarios, the target moved according to the NCV model, but with its velocity limited to a maximum of 20 km/h.

Figure 4 shows a screenshot of the simulator in action. The emitter location is indicated by a white cross in the center of the large circle—the emitter leaves behind it a fading trail showing its path. The large blue circle indicates the desired orbiting radius of the UAVs. Orbiting with precisely the desired radius is not possible in practice due to the random motion of the targets, but is approximated by the UAV control algorithm. The small blue-green squares denote the UAV positions (two in this case)—the UAVs also leave behind fading trails showing their paths. Magenta lines are drawn between the UAVs and their target-position estimates, with the SPKF state three-sigma uncertainty denoted by the magenta ellipse centered on the target position estimate.

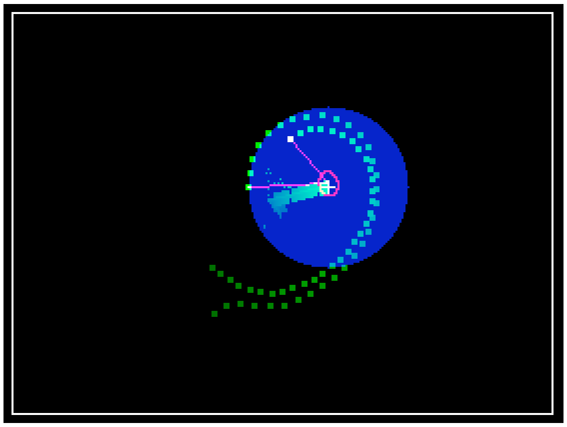
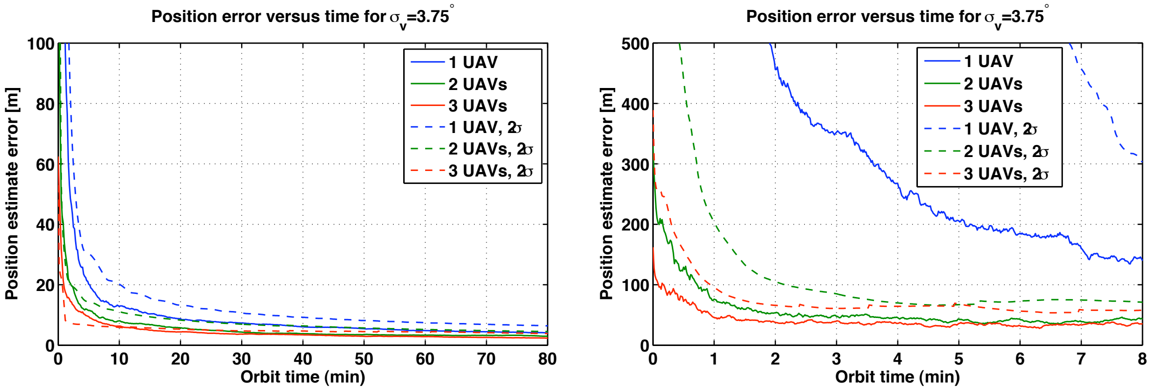


Figure 4. Screenshot of simulation system in action.

For the purpose of this paper, each UAV had one sensor that sensed DOA to the target, and the target was always visible to that sensor. Several different sensor-noise variances were simulated. Some representative results are presented in Fig. 5, where the output of 100 simulations is averaged in each curve. In frame (a), we see the localization error decreasing as the number of sensor measurements increases for localizing a target known to be stationary using the NCP model. As expected, the more UAVs in orbit, the better the localization result. The dashed lines in the figure denote the 2σ error bounds, which bound the true error in most cases. The three-UAV case is somewhat problematic at the beginning due to the approximate initialization method of the covariance matrix (using least-squares, as discussed in section VII), but once the UAVs have settled into orbit the error bounds are consistent with the actual error. (The SPKF algorithm is not as sensitive to poor initial covariance values as EKF is.) In frame (b), we see the localization error obtained while localizing a target known to be mobile using the NCV model. In this case, the results were (understandably) not as good since it is much more difficult to track a moving target. However, we see many of the same trends. In the single UAV scenario, that UAV had much more trouble using only DOA measurements since the target's velocity was a large fraction of the UAV velocity, and it was challenging for the single UAV to move enough to get sufficiently independent views of the target for the SPKF to determine a good location estimate.



(a)

(b)

Figure 5. Example learning curves while localizing a target for sensor with noise standard deviation $3.75°$. In frame (a), the target is stationary and a NCP model of motion is used; in frame (b) the target is moving and a NCV model is used.

Table 5 compares the results using SPKF to the results using KF, as presented in Ref. 16 (localizing a stationary target), where the average final localization error and standard deviation thereof from 100 simulation runs are tabulated. We see that in every scenario the SPKF outperforms the KF. The results are particularly improved for the single-UAV scenario, indicating that a single UAV may be able to achieve a good initial target estimate while waiting for other UAVs to arrive and assist with the effort. Table 6 gives results for a mobile target. As expected, the ultimate localization errors and their respective variances were higher when attempting to localize a mobile target with the same resources. Nevertheless, we see several reasonable trends: (1) the greater the number of UAVs participating in the effort, the lower the ultimate error; (2) the localization error is almost always bounded by the 2σ covariance bound estimate; and (3) the lower the sensor-error covariance, the lower the ultimate position error estimate.

Table 5. DOA-KF versus DOA-SPKF for locating a stationary target. Average final localization error E, and standard deviations of final localization errors σ given in [m].

Sensor		DOA-KF Method			DOA-SPKF Method	
Noise σ_v	1 UAV	2 UAVs	3 UAVs	1 UAV	2 UAVs	3 UAVs
15°	$E = 162.70$	$E = 50.70$	$E = 59.30$	$E = 44.53$	$E = 30.79$	$E = 23.75$
	$\sigma = 111.30$	$\sigma = 39.45$	$\sigma = 86.03$	$\sigma = 25.69$	$\sigma = 19.07$	$\sigma = 11.71$
7.5°	$E = 153.34$	$E = 20.40$	$E = 40.80$	$E = 19.93$	$E = 15.22$	$E = 11.57$
	$\sigma = 111.25$	$\sigma = 27.67$	$\sigma = 144.10$	$\sigma = 9.11$	$\sigma = 8.28$	$\sigma = 5.92$
3.75°	$E = 134.60$	$E = 9.90$	$E = 15.80$	$E = 10.28$	$E = 7.64$	$E = 6.22$
	$\sigma = 111.52$	$\sigma = 10.54$	$\sigma = 34.03$	$\sigma = 5.70$	$\sigma = 4.20$	$\sigma = 3.35$
1.4°	$E = 125.30$	$E = 5.24$	$E = 5.22$	$E = 3.94$	$E = 2.88$	$E = 2.29$
	$\sigma = 97.20$	$\sigma = 4.94$	$\sigma = 5.58$	$\sigma = 2.00$	$\sigma = 1.62$	$\sigma = 1.21$

Table 6. DOA-SPKF for locating a mobile target. Average final localization error E, and standard deviations of final localization errors σ given in [m].

Sensor		DOA-SPKF Method	
Noise σ_v	1 UAV	2 UAVs	3 UAVs
15°	$E = 187.78$	$E = 108.01$	$E = 97.22$
	$\sigma = 112.10$	$\sigma = 53.97$	$\sigma = 50.59$
7.5°	$E = 157.05$	$E = 64.90$	$E = 58.91$
	$\sigma = 131.09$	$\sigma = 32.61$	$\sigma = 30.12$
3.75°	$E = 144.43$	$E = 39.39$	$E = 31.62$
	$\sigma = 112.22$	$\sigma = 19.98$	$\sigma = 17.67$
1.4°	$E = 105.49$	$E = 20.05$	$E = 14.99$
	$\sigma = 87.99$	$\sigma = 12.59$	$\sigma = 8.11$

IX. Conclusions

In this paper, we have presented a method for using sigma-point Kalman filters for the data-fusion aspect of the target-localization problem using multiple UAVs. SPKF is a generalization of the ubiquitous Kalman filter to problems with nonlinear description. They perform better than the more common extended Kalman filter, with the same computational complexity. We were able to apply them successfully to the problem at hand, and achieve better localization results than those reported elsewhere using the KF for stationary target localization. We were also able to easily adapt the methodology for mobile target localization. As would be expected, the ultimate location errors and their respective variances were higher when attempting to localize a moving target using the same resources, but we expect that they can be improved by: (1) sampling the sensor more quickly (the ratio of sample rate to target velocity is important), (2) improving sensor accuracy (decreasing sensor noise), (3) using multiple sensors on the same UAV, and/or (4) orbiting the target at a closer radius.

References

[1] York, G. and Pack, D. J., "Comparative Study of Time-Varying Target Localization Methods Using Multiple Unmanned Aerial Vehicles: Kalman Estimation and Triangulation Techniques," *Proc. IEEE International Conference on Networking, Sensing and Control*, (Tuscon, AZ), March 2005, pp. 305–10.

[2] Pack, D. J. and York, G., "Developing a Control Architecture for Multiple Unmanned Aerial Vehicles to Search and Localize RF Time-

Varying Mobile Targets: Part I," *Proc. IEEE International Conference on Robotics and Automation*, (Barcelona, Spain), April 2005, pp. 3965–70.

[3] van der Merwe, R. and Wan, E., "Efficient Derivative-Free Kalman Filters for Online Learning," *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, April 2001.

[4] van der Merwe, R. and Wan, E., "Sigma-point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models," *Proceedings of the Workshop on Advances in Machine Learning*, (Montreal: June 2003), Available at http://choosh.ece.ogi.edu/spkf/spkf_files/WAML2003.pdf. Accessed 20 May 2004.

[5] van der Merwe, R., *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*, Ph.D. thesis, OGI School of Science & Engineering at Oregon Health & Science University, April 2004.

[6] Kalman, R., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME—Journal of Basic Engineering*, Vol. 82, Series D, 1960, pp. 35–45.

[7] "The Seminal Kalman Filter Paper (1960)," http://www.cs.unc.edu/~welch/kalman/kalmanPaper.html, Accessed 20 May 2004.

[8] Julier, S., Uhlmann, J., and Durrant-Whyte, H., "A new approach for filtering nonlinear systems," *Proceedings of the American Control Conference*, 1995, pp. 1628–32.

[9] Julier, S. and Uhlmann, J., "A New Extension of the Kalman Filter to Nonlinear Systems," *Proceedings of the 1997 SPIE AeroSense Symposium*, SPIE, Orlando, FL, April 21–24, 1997.

[10] Julier, S. and Uhlmann, J., "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, March 2004, pp. 401–22.

[11] Savage, C., Cramer, R., and Schmitt, H., "TDOA Geolocation with the Unscented Kalman Filter," *Proc. 2006 IEEE International Conference on Networking, Sensing and Control*, April 2006, pp. 602–6.

[12] Nørgaard, M., Poulsen, N., and Ravn, O., "Advances in Derivative-Free State Estimation for Nonlinear Systems," Technical rep. IMM-REP-1998-15, Dept. of Mathematical Modeling, Tech. Univ. of Denmark, 28 Lyngby, Denmark, April 2000.

[13] Nørgaard, M., Poulsen, N., and Ravn, O., "New Developments in State Estimation for Nonlinear Systems," *Automatica*, Vol. 36, No. 11, November 2000, pp. 1627–38.

[14] Pack, D. and Mullins, B., "Toward finding a universal search algorithm for swarm robots," *Proceedings of the IEEE/RJS Conference on Intelligent Robotic Systems*, October, (Las Vegas, NV) 2003, pp. 1945–50.

[15] York, G. and Pack, D., "Comparative Study of Time-Varying Target Localization Methods Using Multiple Unmanned Aerial Vehicles: Kalman Estimation and Triangulation Techniques," *Proceedings of the 2005 Networking, Sensing, and Control Conference*, 2005, pp. 305–310.

[16] Toussaint, G. J., De Lima, P., and Pack, D. J., "Localizing RF Targets with Cooperative Unmanned Aerial Vehicles," *Proceedings of the 2007 American Control Conference*, 2007.

[17] Gordon, N., Salmond, D., and Smith, A., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings, Part F*, Vol. 140, 1993, pp. 107–113.

[18] Doucet, A., "On sequential simulation-based methods for Bayesian filtering," Tech. Rep. CUED/F-INFENG/TR 310, Cambridge University Engineering Department, 1998.

[19] Doucet, A., de Freitas, J., and Gordon, N., "Introduction to sequential Monte Carlo methods," *Sequential Monte Carlo Methods in Practice*, edited by A. Doucet, J. de Freitas, and N. Gordon, Berlin: Springer-Verlag, 2000.

[20] Wan, E. and van der Merwe, R., "The Unscented Kalman Filter," *Kalman Filtering and Neural Networks*, edited by S. Haykin, chap. 7, Wiley Inter-Science, New York, 2001, pp. 221–82.

[21] Plett, G. L., "Sigma-Point Kalman Filtering for battery management systems of LiPB-based HEV battery packs, Part 1: Introduction and state estimation," *Journal of Power Sources*, Vol. 161, No. 2, September 2006, pp. 1356–68.

[22] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 2nd ed., 1992.

[23] Stewart, G., *Matrix Algorithms*, Vol. I: Basic Decompositions, SIAM, 1998.

[24] Julier, S. and Uhlmann, J., "A general method for approximating nonlinear transformations of probability distributions," Tech. rep., RRG, Department of Engineering Science, Oxford University, November 1996.

[25] Wan, E. and van der Merwe, R., "The Unscented Kalman Filter for Nonlinear Estimation," *Proc. of IEEE Symposium 2000 (AS-SPCC)*, Lake Louise, Alberta, Canada, October 2000.

[26] Burl, J. B., *Linear Optimal Control: H_2 and H_∞ Methods*, Addison-Wesley, Menlo Park, CA, 1999.