

Unscented Multi-point Smoother for Fusion of Delayed Displacement Measurements: Application to Agricultural Robots

Mathias Hauan Arbo¹ Trygve Utstumo^{1,2} Edmund Brekke¹ Jan Tommy Gravdahl¹

¹*Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway. E-mail: {Mathias.Arbo, Edmund.Brekke, Jan.Tommy.Gravdahl}@itk.ntnu.no*

²*Adigo AS, N-1405 Langhus, Norway. E-mail: Trygve@adigo.no*

Abstract

Visual Odometry (VO) is increasingly a useful tool for robotic navigation in a variety of applications, including weed removal for agricultural robotics. The methods of evaluating VO are often computationally expensive and can cause the VO measurements to be significantly delayed with respect to a compass, wheel odometry, and GPS measurements. In this paper we present a Bayesian formulation of fusing delayed displacement measurements. We implement solutions to this problem based on the unscented Kalman filter (UKF), leading to what we term an unscented multi-point smoother. The proposed methods are tested in simulations of an agricultural robot. The simulations show improvements in the localization RMS error when including the VO measurements with a variety of latencies.

Keywords: Robot Navigation, Sensor fusion, Agricultural Robotics

1 INTRODUCTION



Figure 1: A prototype of the Asterix robot.

In recent years Agricultural Robotics has been an increasingly important research topic, and there are numerous publications presenting unmanned ground vehicles and robot platforms, (Bogue, 2016; Biber et al., 2012; Jørgensen et al., 2007; Molstad et al., 2014; Grim-

stad et al., 2015, 2016). Most agricultural platforms rely heavily on GPS for navigation. RTK GPS systems may provide localization accuracy of ± 0.02 m under ideal conditions. The cost of an RTK GPS system may hinder the adaption of mobile robots in agriculture, and the signal conditions often reduce the position accuracy.

Visual-aided navigation may provide several benefits for agricultural robots, and lessen the dependence on expensive RTK GPS systems. The work presented here has sprung out from a research project by Adigo AS, in cooperation with the Norwegian University of Science and Technology and the Norwegian Institute of Bioeconomy Research (NIBIO).

An agricultural robot for high precision drop-on-demand herbicide application for row crops is under development. The robot uses a downward facing camera to identify weeds and a nozzle array applies the

herbicide as the robot passes over them, (Urdal et al., 2014; Utstumo and Gravdahl, 2013; Utstumo et al., 2015).

Weed control is a vital part of agriculture, and autonomous robotic weed control has become an important research area. The review by Slaughter et al. (2008) illustrates the potential of robotic weed control, and presents several similar systems to the one which is the focus of this paper.

The same images used for plant classification may also be used to compute visual odometry (VO) measurements. In this paper, VO measurements are assumed to be frame-to-frame rotation and displacements that the robot has undergone between two overlapping images.

The VO technique considered finds a set of matching features in two subsequent images. These features are used to reconstruct the movement of the camera by minimization of the transformation that matches one set to the other. This is similar to wheel odometry (WO) which uses the wheel encoders to reconstruct the movement of the robot. Unlike the WO measurements, VO will be unaffected by any skidding and sliding of the wheels. Both VO and WO are relative displacements with respect to a previous state of the system. Absolute measurements, such as GPS, provide measurements with respect to a known coordinate frame.

The feature identification and matching algorithms introduces a latency between when a picture is taken, and when the measurement becomes available. We refer to this time as processing delay. This processing delay may be more than one second in non-optimized implementations, to an order of milliseconds in implementations as described in Forster et al. (2014), or with dedicated hardware and tight coupling of inertial and VO measurements as in Goldberg and Matthies (2011).

The processing delay for the VO measurement is characteristically different from the other available sensors such as compass, GPS, wheel encoders, which are considered instantaneous in this paper. The process of detecting and matching feature points, and solving for the displacement measurement, is not necessarily fixed. It may vary with the number of feature points processed.

The camera is triggered by a hardware GPIO line. This allows us to know when a picture is taken with the same time reference as the GPS, compass and other measurements.

For the fusion of relative displacement measurements, a method called stochastic cloning (Mourikis et al., 2007) was considered. This method “clones” the state estimates when a measurement should have arrived, augmenting the state vector used by the filter, thus maintaining the cross-covariances between

the current state of the system and the time when a measurement should have arrived. Practical examples of stochastic cloning can be found in Romanovas et al. (2013) for visual-inertial/magnetic data fusion, in Van der Merwe and Wan (2004) under the name latency compensation, and in Mourikis et al. (2009) for spacecraft entry, descent and landing.

Similar to Van der Merwe and Wan (2004), we use the UKF as opposed to the EKF. The EKF often suffers from providing covariance estimates that are lower than the actual covariance, something which can be very detrimental in precision agriculture where the decision to spray or not may be directly based on the covariance estimate. Instead we use the Unscented Kalman filter (UKF), which through the unscented transform (UT) deals more directly with non-linearities by choosing a strategic set of sample points (Julier and Uhlmann, 1997).

The main contributions of this paper are

- A presentation of a Bayesian framework for delayed displacement fusion, which build upon the method of stochastic cloning.
- An implementation of a novel fixed-point smoother termed unscented multi-point smoother.
- A description of the relation between stochastic cloning and the proposed framework.

The remainder of this paper is structured as follows: in Section 2 the Bayesian formulation of delayed displacement measurement fusion is presented. The dependencies in the basic model and the delayed displacement scenario are presented with Hidden Markov models. These are used to formulate a Bayesian filter capable of fusing the delayed displacement measurement. In Section 3 the unscented multi-point smoother is implemented. First Gaussian assumptions are applied to the filter algorithm, such that the algorithm is described using an augmented state vector for a multivariate Gaussian distribution. Then the unscented transform is briefly described for evaluating the necessary expectations and covariances. In Section 4 the robot simulation setup is described, followed by simulation results. In Section 5 potential future work is outlined.

2 BAYESIAN FORMULATION

In this section we present the Bayesian formulation of fusion with delayed displacement measurements, using graphical models to indicate stochastic dependencies.

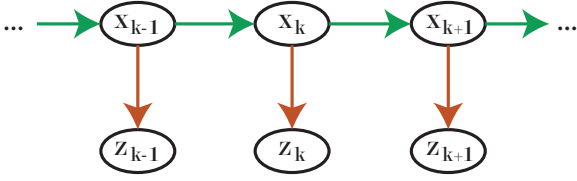


Figure 2: Hidden Markov Model of the basic system (1)-(2) with state vector \mathbf{x} , and measurement vector \mathbf{z} . Green arrows represent (1) and red represent (2).

2.1 Basic System

The basic underlying system considered in this paper is

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{w}_k \quad (1)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (2)$$

$$\mathbf{w}_k = \mathcal{N}(\mathbf{w}_k; 0, Q_k) \quad (3)$$

$$\mathbf{v}_k = \mathcal{N}(\mathbf{v}_k; 0, R_k) \quad (4)$$

where $\mathbf{x} \in \mathbb{R}^L$ is the state vector of pose, forward velocity, and rotational velocity, \mathbf{z} are the measurements from the fast sensors, compass, GPS, and WO. The process noise \mathbf{w}_k and measurement \mathbf{v}_k is considered additive Gaussian. In Fig. 2 a Hidden Markov model is presented where there is no delay on any measurements, and there are no displacement measurements. The green arrows indicate the transition model, (1), showing how the next state is only conditioned on the previous state. The red arrow indicates the measurement model, (2), and how the likelihood of the measurement is only conditioned on the current state. We desire to find the probability density function (pdf) of the current state given the measurements, $p(\mathbf{x}_k | \mathbf{z}_{0:k})$. For the basic system this can be done with a recursive Bayesian filter (Thrun et al., 2005).

2.2 Delayed Displacement Measurements

The delayed displacement measurement model is

$$\mathbf{d}_n = \tilde{h}(\mathbf{x}_l, \mathbf{x}_m) + \tilde{\mathbf{v}}_n \quad (5)$$

$$\tilde{\mathbf{v}}_n = \mathcal{N}(\tilde{\mathbf{v}}_n; 0, \tilde{R}_n) \quad (6)$$

where \mathbf{d}_n is the delayed displacement measurement dependent on two previous states \mathbf{x}_l and \mathbf{x}_m , and $\tilde{\mathbf{v}}_n$ is Gaussian noise. The Hidden Markov model of the system is given in Fig. 3. At time t_l the first picture is taken, and at time t_m the second picture is taken. The displacement measurement becomes available at time t_n . We desire to find the distribution $p(\mathbf{x}_n | \mathbf{z}_{0:n}, \mathbf{d}_n)$.

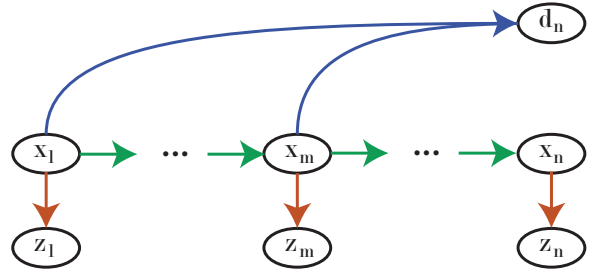


Figure 3: Hidden Markov model showing a delayed displacement measurement \mathbf{d}_n being dependent on the states \mathbf{x}_l and \mathbf{x}_m , but arriving at t_n . Green arrows represent (1), red arrows represent (2), and blue arrows represent (5).

As the delayed displacement measurement is dependent on two previous states, optimal fusion of \mathbf{d}_n requires a joint distribution of \mathbf{x}_l , \mathbf{x}_m and \mathbf{x}_n .

In reality, the displacement measurements are dependent on the underlying features’ poses relative to the camera’s pose at time t_l and t_m . The simplification of the displacement measurement as a function of the states is based on Mourikis et al. (2007).

With an initial distribution $p(\mathbf{x}_0)$, from t_0 until t_l , the measurements $\mathbf{z}_{0:l}$ are fused iteratively using the recursive Bayes filter up to $p(\mathbf{x}_l | \mathbf{z}_l)$. From there on the filter must maintain \mathbf{x}_l . This can be done by augmenting the state vector by a “clone” \mathbf{x}_l^c of \mathbf{x}_l at time t_l . This leads to a joint density as

$$p(\mathbf{x}_l, \mathbf{x}_l^c | \mathbf{z}_l) = p(\mathbf{x}_l^c | \mathbf{x}_l) p(\mathbf{x}_l | \mathbf{z}_l) \quad (7)$$

where $p(\mathbf{x}_l^c | \mathbf{x}_l)$ is a Dirac delta distribution. Thus, even if $p(\mathbf{x}_l | \mathbf{z}_l)$ is Gaussian, the joint density will not be Gaussian unless a regularization constant ϵ is used. Under Gaussian assumptions,

$$p(\mathbf{x}_l, \mathbf{x}_l^c | \mathbf{z}_l) = \mathcal{N}(\mathbf{x}_l^c; \mathbf{x}_l, \epsilon I) \mathcal{N}(\mathbf{x}_l; \hat{\mathbf{x}}_{l|l}, P_{l|l}). \quad (8)$$

As just as straightforward, and in principle more general, solution to the delayed fusion problem follows from “extending” the state vector when the first prediction after time t_l is done:

$$p(\mathbf{x}_l, \mathbf{x}_{l+1} | \mathbf{z}_l) = p(\mathbf{x}_{l+1} | \mathbf{x}_l) p(\mathbf{x}_l | \mathbf{z}_l) \quad (9)$$

which under Gaussian assumptions is

$$p(\mathbf{x}_l, \mathbf{x}_{l+1} | \mathbf{z}_l) = \mathcal{N}(\mathbf{x}_{l+1}; f(\mathbf{x}_l), Q_l) \mathcal{N}(\mathbf{x}_l; \hat{\mathbf{x}}_{l|l}, P_{l|l}). \quad (10)$$

This is referred to as extending, and occurs as an alternative to prediction at timestep t_{l+1} . In this spirit, we can make further extensions as required. In particular, we also extend at time t_{m+1} to arrive at the joint distribution $p(\mathbf{x}_l, \mathbf{x}_m, \mathbf{x}_{m+1} | \mathbf{z}_m)$, which, when iteratively fusing \mathbf{z} with a recursive Bayes filter leads to $p(\mathbf{x}_l, \mathbf{x}_m, \mathbf{x}_n | \mathbf{z}_n)$.

When a picture has been taken at a timestep, the next predict step in the filter is replaced with an extend step. A system where the camera is not synchronized with the filter will face the additional challenge of estimating the clock synchronization and trigger time.

With the joint distribution $p(\mathbf{x}_l, \mathbf{x}_m, \mathbf{x}_n | \mathbf{z}_n)$, the delayed displacement measurement \mathbf{d}_n is fused using Bayes theorem according to

$$p(\mathbf{x}_l, \mathbf{x}_m, \mathbf{x}_n | \mathbf{z}_n, \mathbf{d}_n) = \mu p(\mathbf{d}_n | \mathbf{x}_l, \mathbf{x}_m) p(\mathbf{x}_l, \mathbf{x}_m, \mathbf{x}_n | \mathbf{z}_n) \quad (11)$$

where μ is the normalization constant and $p(\mathbf{d}_n | \mathbf{x}_l, \mathbf{x}_m)$ is the likelihood of the displacement measurement. Note that the likelihood is specified conditional on both \mathbf{x}_l and \mathbf{x}_m , but not on \mathbf{x}_n . If no other displacement measurements depend on the states \mathbf{x}_l and \mathbf{x}_m , the states are marginalized from the joint distribution by

$$p(\mathbf{x}_n | \mathbf{z}_n, \mathbf{d}_n) = \int_{\mathbf{x}_l} \int_{\mathbf{x}_m} p(\mathbf{x}_l, \mathbf{x}_m, \mathbf{x}_n | \mathbf{z}_n, \mathbf{d}_n) d\mathbf{x}_m d\mathbf{x}_l \quad (12)$$

which under Gaussian assumptions is simply done by omitting the parts of the expectation vector and covariance matrix associated with \mathbf{x}_l and \mathbf{x}_m .

In Alg.1 the delayed displacement fusing filter algorithm is outlined. This algorithm is a Bayesian on-demand smoother. When a picture has been taken, the state vector is extended with the predicted state. And after updating with a displacement measurement, any unnecessary old states are marginalized. The list of state indices when the pictures have been taken is denoted K . The set N contains the current index i only if a displacement measurement is available. The augmented state vector \mathbf{x}_S contains all the states maintained in the joint distribution.

The purpose of the list S is to keep track of the corresponding state indices. In the algorithm \cup is used to append an index, and \setminus is used to remove indices. The function $ind(\mathbf{d})$ returns the indices associated with a displacement measurement \mathbf{d} (e.g. l and m) that can be marginalized. The function ind and the list S are used to handle the case when a picture is used for more than one displacement measurement. This is generally the case, as an image generates displacement

Algorithm 1 Bayesian delayed displacement fusion

Require: $p(\mathbf{x}_0)$, K , N

- 1: $S \leftarrow \{0\}$
- 2: **for** $i = 1$ **to** $i = \infty$ **do**
- 3: *Predict:*
- 4: **if** $i \in K$ **then** $\{picture\ taken\}$
- 5: $p(\mathbf{x}_S, \mathbf{x}_i | \mathbf{z}_{0:i-1}) = p(\mathbf{x}_i | \mathbf{x}_{i-1}) p(\mathbf{x}_S)$
- 6: $S \leftarrow S \cup \{i\}$
- 7: **else**
- 8: $p(\mathbf{x}_S, \mathbf{x}_i | \mathbf{z}_{0:i-1}) = \int_{\mathbf{x}_{i-1}} p(\mathbf{x}_i | \mathbf{x}_{i-1}) p(\mathbf{x}_S) d\mathbf{x}_{i-1}$
- 9: $S \leftarrow S \setminus \{i-1\}$
- 10: $S \leftarrow S \cup \{i\}$
- 11: **end if**
- 12: *Update:*
- 13: **if** $i \in N$ **then** $\{displacement\ available\}$
- 14: $p(\mathbf{x}_S | \mathbf{z}_{0:i}, \mathbf{d}_i) = \mu p(\mathbf{z}_i, \mathbf{d}_i | \mathbf{x}_S) p(\mathbf{x}_S | \mathbf{z}_{0:i-1})$
- 15: $A \leftarrow ind(\mathbf{d}_i)$
- 16: $p(\mathbf{x}_i | \mathbf{z}_{0:i}, \mathbf{d}_i) = \int_{\mathbf{x}_A} p(\mathbf{x}_S | \mathbf{z}_{0:i}, \mathbf{d}_i) d\mathbf{x}_A$
- 17: $S \leftarrow S \setminus A$
- 18: **else**
- 19: $p(\mathbf{x}_S | \mathbf{z}_{0:i}, \mathbf{d}_i) = \mu p(\mathbf{z}_i | \mathbf{x}_i) p(\mathbf{x}_S | \mathbf{z}_{0:i-1})$
- 20: **end if**
- 21: **end for**

measurements both together with the preceding and the succeeding image.

K is updated by the camera triggering before each iteration and N is updated before the update step by the VO module indicating a displacement measurement being ready. The filter acts similar to a fixed-point smoother with the capability of formulating the points to be smoothed on-demand, a ‘‘multi-point’’ smoother.

3 UNSCENTED MULTI-POINT SMOOTHER

In this section we look at an unscented multi-point smoother based on Alg.1. First we describe more thoroughly what the Gaussian assumptions means for our augmented state vector \mathbf{x}_S , then we present important properties of the unscented transformation, and show how the method in this paper relates to stochastic cloning.

3.1 Gaussian Assumption

To implement Alg.1 with a UKF as the underlying filter, one must define how the extension method, (10), under Gaussian assumptions, behaves for the augmented state vector \mathbf{x}_S . Consider the case where we

are extending at time t_k

$$p(\mathbf{x}_S, \mathbf{x}_k | \mathbf{z}_{k-1}) = p(\mathbf{x}_k | \mathbf{x}_S) p(\mathbf{x}_S | \mathbf{z}_{k-1}) = \mathcal{N} \left(\begin{bmatrix} \mathbf{x}_S \\ \mathbf{x}_k \end{bmatrix}; \begin{bmatrix} \mathbf{E}(\mathbf{x}_S) \\ \mathbf{E}(\mathbf{x}_k) \end{bmatrix}, \begin{bmatrix} \text{Var}(\mathbf{x}_S) & \text{Cov}(\mathbf{x}_S, \mathbf{x}_k) \\ \text{Cov}(\mathbf{x}_k, \mathbf{x}_S) & \text{Var}(\mathbf{x}_k) \end{bmatrix} \right). \quad (13)$$

The notation $\text{Cov}(\mathbf{x}_S, \mathbf{x}_k)$ is shorthand for

$$\text{Cov}(\mathbf{x}_S, \mathbf{x}_k) = \begin{bmatrix} \text{Cov}(\mathbf{x}_{S_1}, \mathbf{x}_k) \\ \vdots \\ \text{Cov}(\mathbf{x}_{k-1}, \mathbf{x}_k) \end{bmatrix} \quad (14)$$

where S_1 indicates the oldest index in the list of state indices S . The newest index is always $k-1$. The mean and covariance of \mathbf{x}_S is maintained by the filter. We must evaluate the parts associated with \mathbf{x}_k . In the following section we describe $\mathbf{E}(\mathbf{x}_k)$ and $\text{Var}(\mathbf{x}_k)$ with the unscented transform. For the cross-covariances between \mathbf{x}_S and \mathbf{x}_k , (14), consider S^+ to be a list of state indices containing the same indices as S , with the last index, $k-1$ replaced with k . Then the cross-covariance between \mathbf{x}_S and \mathbf{x}_{S^+} is by definition

$$\text{Cov}(\mathbf{x}_S, \mathbf{x}_{S^+}) = \begin{bmatrix} \text{Var}(\mathbf{x}_{S_1}) & \dots & \text{Cov}(\mathbf{x}_{S_1}, \mathbf{x}_k) \\ \vdots & \ddots & \vdots \\ \text{Cov}(\mathbf{x}_{k-1}, \mathbf{x}_{S_1}) & \dots & \text{Cov}(\mathbf{x}_{k-1}, \mathbf{x}_k) \end{bmatrix}. \quad (15)$$

Hence, evaluating the cross-covariance between \mathbf{x}_S and \mathbf{x}_{S^+} , and extracting the rightmost columns of size L , gives us (14). This augmented system keeps all but the last state the same between each iteration.

$$\mathbf{x}_{S^+} = f^a(\mathbf{x}_S) = \begin{bmatrix} \mathbf{x}_{S_1} \\ \mathbf{x}_{S_2} \\ \vdots \\ f(\mathbf{x}_{k-1}) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \mathbf{w}_{k-1} \end{bmatrix} \quad (16)$$

$$\mathbf{z}_k = h^a(\mathbf{x}_{S^+}) + \mathbf{v}_k = h(\mathbf{x}_k) + \mathbf{v}_k. \quad (17)$$

The augmented version of the displacement measurement model \tilde{h} must know which states in the augmented state vector \mathbf{x}_S correspond to the imminent displacement measurement

$$\mathbf{d}_n = \tilde{h}^a(\mathbf{x}_S) + \tilde{\mathbf{v}}_n. \quad (18)$$

3.2 Unscented Transform

In this section we briefly introduce the unscented transform and the properties that need to be evaluated for Alg.1.

The basic premise of the unscented transform is that it is easier to approximate a probability distribution than a nonlinear function (Julier and Uhlmann, 2002). This is done by propagating a set of deterministically chosen sigma-points through the nonlinear function. For a nonlinear transformation

$$\mathbf{y} = f(\mathbf{x}) \quad (19)$$

where $\mathbf{x} \in \mathbb{R}^L$ is a Gaussian random variable of dimension L . Sigma-points can be defined by the selection scheme

$$\mathcal{Y}_0 = \mathbf{E}(\mathbf{x}) \quad (20)$$

$$\mathcal{Y}_h = \mathbf{E}(\mathbf{x}) + \sqrt{(L+\lambda)\text{Var}(\mathbf{x})}_i \quad (21)$$

$$\mathcal{Y}_j = \mathbf{E}(\mathbf{x}) - \sqrt{(L+\lambda)\text{Var}(\mathbf{x})}_i \quad (22)$$

$$w_0 = 1/(L+\lambda) \quad (23)$$

$$w_{h,j} = 1/2(L+\lambda) \quad (24)$$

for $h = 1, \dots, L$, $i = 1, \dots, L$, and $j = L+1, \dots, 2L+1$, where λ is a tuning parameter defining the spread of the sigma-points around the expected value (Van der Merwe, 2004).

The subscript i of the matrix $\text{Var}(\mathbf{x})$ denotes its i th column vector. From Van der Merwe (2004), under the Gaussian assumption, the mean, covariance and cross-covariance are constructed from the sigma-points by

$$\mathbf{E}(\mathbf{y}) = \sum_{i=0}^{2L+1} w_i \mathcal{Y}_i, \quad (25)$$

$$\text{Var}(\mathbf{y}) = \sum_{i=0}^{2L+1} w_i (\mathcal{Y}_i - \mathbf{E}(\mathbf{y})) (\mathcal{Y}_i - \mathbf{E}(\mathbf{y}))^T, \quad (26)$$

and

$$\text{Cov}(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{2L+1} w_i (\mathcal{X}_i - \mathbf{E}(\mathbf{x})) (\mathcal{Y}_i - \mathbf{E}(\mathbf{y}))^T. \quad (27)$$

These methods allow us to evaluate the expectation, covariance, and cross-covariance of the augmented system described in the previous section, and the unscented multi-point smoother can be constructed.

3.3 Remark on Stochastic Cloning

If the underlying filter is an EKF, then $f(\mathbf{x}_k) \approx F_k \mathbf{x}_k$. The cloning procedure of (7) without a regularization constant constructs the augmented expectation state vector $\hat{\mathbf{x}}_{S|k}$, and covariance matrix $P_{k|k}^S$ of the filter according to

$$\hat{\mathbf{x}}_{S|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k|k} \end{bmatrix} \quad (28)$$

$$P_{S|k} = \begin{bmatrix} P_{k|k} & P_{k|k} \\ P_{k|k} & P_{k|k} \end{bmatrix}. \quad (29)$$

This does not describe a Gaussian distribution as the covariance matrix is singular. When performing the prediction step, an UKF will fail as it relies on the Cholesky factorization of the state covariance matrix, which is only unique on nonsingular matrices. On the other hand, an EKF can readily perform the prediction step with an augmented transition model giving the expectation state vector and covariance matrix given by

$$\hat{\mathbf{x}}_{S^+|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ F_k \hat{\mathbf{x}}_{k|k} \end{bmatrix} \quad (30)$$

and

$$P_{S^+|k} = \begin{bmatrix} P_{k|k} & P_{k|k} F_k^T \\ F_k P_{k|k} & F_k P_{k|k} F_k^T + Q_k \end{bmatrix}, \quad (31)$$

which is identical to the extension method described in this paper. It is interesting to note that by Schur's complement, we require non-zero process noise to ensure that the distribution remains Gaussian through an extend step.

3.4 Remark on Smoothing

The filter algorithm, Alg.1, performs optimal fixed-point smoothing of the augmented states under the assumption that the model can be described by (1), (2) and (5). If the transformations are highly nonlinear, the unscented transform will also encounter problems. As such, it is debatable whether the smoothing procedure is beneficial. When absolute measurements are available from e.g. GPS or a compass, it was observed that the covariance associated with an old augmented state diminished to the point where the delayed displacement measurement was assumed to be more accurate than it actually was. To remedy this, for the simulation with high process delay and low camera framerate, the covariance associated with the oldest augmented state had a lower threshold it could not decrease below. This was used on the pose of the robot as these were the states affected by the absolute measurements.

4 ROBOT SIMULATION

4.1 System

The robot is modeled as a unicycle-like robot with no-slip conditions. The kinematic model for the system is based on the model by Cruz and Carelli (2006), with a change where the castor wheel is at the rear end of the robot,

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{u} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} u \cos \psi - a\omega \sin \psi \\ u \sin \psi + a\omega \cos \psi \\ \omega \\ 0 \\ 0 \end{bmatrix} \quad (32)$$

where x and y is the robot's east-north position, ψ is yaw, u is the forward speed, and ω is the yaw rate. The kinematic model (32) was implemented by using the forward Euler method and Gaussian process noise was added:

$$\dot{\mathbf{x}} = f_c(\mathbf{x}) \quad (33)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + f_c(\mathbf{x}_k)(t_{k+1} - t_k) + \mathbf{w}_k. \quad (34)$$

We assume that the onboard sensors are not delayed, except for the visual odometry measurements, when setting up the observation mapping $h(\mathbf{x}_k)$, (2). We model the onboard RTK-GPS, compass, and wheel odometry, as direct measurements of the position, (x, y) , the heading (ψ), and linear and angular velocities (u, ω) respectively. The measurements are assumed to be affected by additive Gaussian measurement noise.

The measurement \mathbf{d}_n is rotated by the oldest picture frame,

$$\mathbf{d}_n = \begin{bmatrix} R(-\psi_k) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_m - x_k \\ y_m - y_k \\ \psi_m - \psi_k \end{bmatrix} + \tilde{\mathbf{v}}_n \quad (35)$$

where $R(\cdot)$ is the 2D rotation matrix. Using the simulink models and controller of Martins (2013) the robot was simulated with a timestep of 0.05 s, following a figure eight trajectory. As the robot is heavy and slow moving, the process noise is negligible. The acronym JUKF is used for the delayed displacement fusing UKF in reference to the joint distribution. The kinematic controller has an upper bound of 1.5 m/s in these simulations.

For the simulations, the errors of the filter estimates are

$$\begin{bmatrix} \tilde{x}_k \\ \tilde{y}_k \\ \tilde{\psi}_k \\ \tilde{u}_k \\ \tilde{\omega}_k \end{bmatrix} = \mathbf{x}_k - \hat{\mathbf{x}}_{k|k}. \quad (36)$$

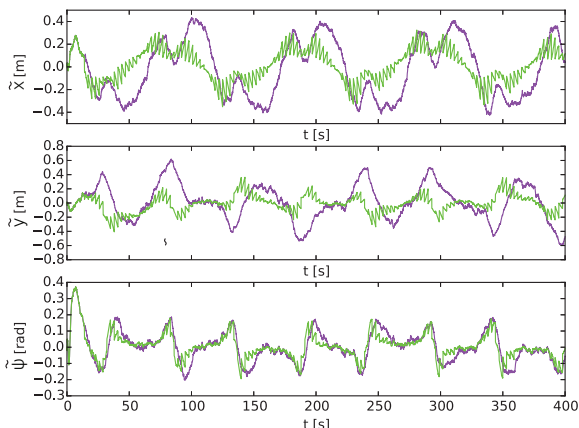


Figure 4: Simulation of the robot showing pose error with respect to time. Simulated with $\tau_C = 3s$ and $\tau_D = 2s$. Purple indicates UKF, green indicates JUKF.

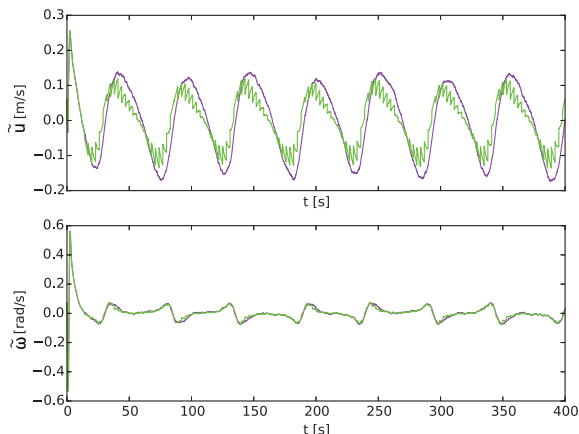


Figure 5: Simulation of the robot showing velocity error with respect to time. Simulated with $\tau_C = 3s$ and $\tau_D = 2s$. Purple indicates UKF, green indicates JUKF.

4.2 Simulation Results

In Test 1 the time between images, τ_C , is 0.25 s, and the processing delay, τ_D is 0.2 s. The RMS errors are given in Tab. 1. The JUKF fusing the delayed displacement measurements has a lower RMS error than the UKF not fusing the delayed displacement measurement.

In Test 2 the time between images is 2 s, and the processing delay is 1 s. The RMS errors are given in Tab. 2. The JUKF fusing the delayed displacement

Table 1: RMS errors of the simulation Test 1, $\tau_C = 0.25s$, $\tau_D = 0.2s$.

	x	y	ψ	u	ω
UKF	0.272287	0.239938	0.089938	0.100554	0.045322
JUKF	0.040214	0.043980	0.034718	0.017865	0.032077

Table 2: RMS errors of the simulation Test 2, $\tau_C = 2s$, $\tau_D = 1s$

	x	y	ψ	u	ω
UKF	0.262771	0.233008	0.098997	0.101097	0.058464
JUKF	0.093007	0.115435	0.065567	0.051329	0.053759

measurement has a lower RMS error than the UKF not fusing the delayed displacement measurement.

In Test 3 the time between images is 3 s, and the processing delay is 2 s. The RMS errors are given in Tab. 3. See Fig. 4 for the error in pose with respect to time, and Fig. 5 for error in velocities with respect to time. The purple line is the UKF, the green is the JUKF. Note that whenever a VO measurement is fused, the error in JUKF decreases, causing a sawtooth effect.

In Test 4 the time between images is 5 s, and the processing delay is 3 s. The RMS errors are given in Tab. 4. In this test a covariance threshold was applied to the covariances associated with the oldest x , y and ψ . Without the covariance threshold, the JUKF did not have lower RMS errors than the UKF. At this frame rate, the pictures are not expected to overlap for the robot under typical field conditions. This test was included to show that the algorithm show improvement in the localization RMS error for cases beyond the worst expected latencies.

5 Future Work and Discussion

The algorithm proposed in this paper handles a mild case of out-of-sequence measurements, in the sense that displacement measurements are only processed after several GPS and compass measurements have been processed, i.e. out-of-sequence. The key to handling this is the *ind* function, declaring which augmented states are to be marginalized.

In the simulations presented here, we have only considered GPS and compass measurements in addition to VO. In practice, most platforms will also have inertial

Table 3: RMS errors of the simulation Test 3, $\tau_C = 3s$, $\tau_D = 2s$

	x	y	ψ	u	ω
UKF	0.251910	0.249378	0.099966	0.101343	0.058235
JUKF	0.138441	0.117433	0.081633	0.072851	0.056340

Table 4: RMS errors of the simulation Test 4, $\tau_C = 5s$, $\tau_D = 4s$.

	x	y	ψ	u	ω
UKF	0.278994	0.252959	0.098941	0.102308	0.057945
JUKF	0.265563	0.244570	0.09856	0.097869	0.057328

sensors available, which may be included in the measurement model. When using MEMS sensors, it would also be natural to include the gyro and accelerometer biases in the filter.

Finally, this work employed a full-state representation as opposed to an error-state representation, which has been used for visual odometry as described in Mourikis and Roumeliotis (2007). In continuation of this research, it may be interesting to investigate how error-state smoothing relates to the Bayesian paradigm.

6 Conclusion

This paper has proposed a Bayesian framework for the fusion of delayed displacement measurements. This led to a multi-point smoother capable of defining fixed-points to be smoothed on demand according to the GPS, WO, and compass measurements. This filtering technique may prove useful in other scenarios where fixed-points to be smoothed are formulated while the filter is running.

By using the unscented transform, the filter was able to use the smoothing effects of the fusion method for its benefits. The filter maintains estimates of the state the robot was in when a picture was taken, smoothed by the absolute measurements. The VO fusing methods improved the localization RMS error for a variety of latencies.

References

- Biber, P., Weiss, U., Dorna, M., and Albert, A. Navigation System of the Autonomous Agricultural Robot "BoniRob". *Workshop on Agricultural Robotics: Enabling Safe, Efficient, and Affordable Robots for Food Production*, 2012. pages 1–7.
- Bogue, R. Robots poised to revolutionise agriculture. *Industrial Robot: An International Journal*, 2016. 43(5):450–456. doi:[10.1108/IR-05-2016-0142](https://doi.org/10.1108/IR-05-2016-0142).
- Cruz, C. D. L. and Carelli, R. Dynamic Modeling and Centralized Formation Control of Mobile Robots. In *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*. pages 3880–3885, 2006. doi:[10.1109/IECON.2006.347299](https://doi.org/10.1109/IECON.2006.347299).
- Forster, C., Pizzoli, M., and Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. *Proceedings - IEEE International Conference on Robotics and Automation*, 2014. pages 15–22. doi:[10.1109/ICRA.2014.6906584](https://doi.org/10.1109/ICRA.2014.6906584).
- Goldberg, S. B. and Matthies, L. Stereo and IMU assisted visual odometry on an OMAP3530 for small robots. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2011. doi:[10.1109/CVPRW.2011.5981842](https://doi.org/10.1109/CVPRW.2011.5981842).
- Grimstad, L., Pham, C. D., Phan, H. T., and From, P. J. On the design of a low-cost, light-weight, and highly versatile agricultural robot. *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts, ARSO*, 2016. 2016-March(October). doi:[10.1109/ARSO.2015.7428210](https://doi.org/10.1109/ARSO.2015.7428210).
- Grimstad, L., Phan, H. N. T., Pham, C. D., and Bjugstad, N. Initial field-testing of Thorvald, a versatile robotic platform for agricultural applications. *IEEE International Workshop on Advanced Robotics and its Social Impacts*, 2015.
- Jørgensen, R. N., Sørensen, C. G., Pedersen, J. M., Havn, I., Jensen, K., Søggaard, H. T., and Sørensen, L. Hortibot: A system design of a robotic tool carrier for high-tech plant nursing. *CIGR Ejournal*, 2007. IX:Manuscript ATOE 07 006.
- Julier, S. J. and Uhlmann, J. K. New extension of the kalman filter to nonlinear systems. 1997. doi:[10.1117/12.280797](https://doi.org/10.1117/12.280797).
- Julier, S. J. and Uhlmann, J. K. Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, 2002. 2:887–892 vol.2. doi:[10.1109/ACC.2002.1023128](https://doi.org/10.1109/ACC.2002.1023128).
- Martins, F. Velocity-based dynamic model and adaptive controller for differential steered mobile robot. *Mathworks Fileexchange 44850*, 2013.
- Van der Merwe, R. *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. Ph.D. thesis, Oregon Health & Science University, 2004.
- Van der Merwe, R. and Wan, E. A. Sigma-Point Kalman Filters for Integrated Navigation. In *Proceedings of the 60th Annual Meeting of the Institute of Navigation (ION)*. pages 641–654, 2004.

- Molstad, L., Reent Köster, J., Bakken, L., Dörsch, P., Lien, T., Overskeid, Ø., Utstumo, T., Løvås, D., and Brevik, A. A field robot for autonomous laser-based n₂o flux measurements. In *EGU General Assembly Conference Abstracts*, volume 16. page 16109, 2014.
- Mourikis, A. I. and Roumeliotis, S. I. A multi-state constraint Kalman filter for vision-aided inertial navigation. *Proceedings - IEEE International Conference on Robotics and Automation*, 2007. pages 3565–3572. doi:[10.1109/ROBOT.2007.364024](https://doi.org/10.1109/ROBOT.2007.364024).
- Mourikis, A. I., Roumeliotis, S. I., and Burdick, J. W. SC-KF mobile robot localization: A stochastic cloning Kalman filter for processing relative-state measurements. *IEEE Transactions on Robotics*, 2007. 23(4):717–730. doi:[10.1109/TRO.2007.900610](https://doi.org/10.1109/TRO.2007.900610).
- Mourikis, A. I., Trawny, N., Roumeliotis, S. I., Johnson, A. E., Ansar, A., and Matthies, L. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 2009. doi:[10.1109/TRO.2009.2012342](https://doi.org/10.1109/TRO.2009.2012342).
- Romanovas, M., Schwarze, T., Schwaab, M., and Traechtler, M. Stochastic Cloning Kalman Filter for Visual Odometry and Inertial / Magnetic Data Fusion. *Information Fusion (FUSION)*, 2013, 2013. pages 1434–1441.
- Slaughter, D. C., Giles, D. K., and Downey, D. Autonomous robotic weed control systems: A review. *Computers and Electronics in Agriculture*, 2008. 61(1):63–78. doi:[10.1016/j.compag.2007.05.008](https://doi.org/10.1016/j.compag.2007.05.008).
- Thrun, S., Burgard, W., and Fox, D. *Probabilistic robotics*. MIT press, 2005.
- Urdal, F., Utstumo, T., Vatne, J. K., Ellingsen, S. A. A., and Gravdahl, J. T. Design and control of precision drop-on-demand herbicide application in agricultural robotics. In *2014 13th International Conference on Control Automation Robotics and Vision, ICARCV 2014*. IEEE, Singapore, pages 1689–1694, 2014. doi:[10.1109/ICARCV.2014.7064570](https://doi.org/10.1109/ICARCV.2014.7064570).
- Utstumo, T., Berge, T. W., and Gravdahl, J. T. Non-linear model predictive control for constrained robot navigation in row crops. *2015 IEEE International Conference on Industrial Technology (ICIT)*, 2015. (July):357–362. doi:[10.1109/ICIT.2015.7125124](https://doi.org/10.1109/ICIT.2015.7125124).
- Utstumo, T. and Gravdahl, J. T. Implementation and comparison of attitude estimation methods for agricultural robotics. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2013. 4(PART 1):52–57. doi:[10.3182/20130828-2-SF-3019.00051](https://doi.org/10.3182/20130828-2-SF-3019.00051).