State Estimation of Multi-Agent Systems under Impulsive Noise and Disturbances

Luis Rodolfo García Carrillo, Wm. Joshua Russell,
João P. Hespanha, and Gaemus E. Collins

Abstract

We address the problem of estimating the state of a multi-agent system based on measurements corrupted by impulsive noise and whose dynamics are subject to impulsive disturbances. The qualifier "impulsive" refers to the fact that noise and disturbances are relatively small most of the time, but occasionally take large values. Noise and disturbances are modeled as mixtures of Gaussian and a Laplacian processes, leading to a maximum likelihood estimator that can be computed by solving a convex sum-of-norms optimization that can be solved online very efficiently. The approach has been validated both in simulation using synthetic data and in real hardware using a team of Unmanned Air Vehicles (UAVs) equipped with an onboard video cameras, inertial sensors, and GPS to cooperatively geolocate and track a ground-moving target agent.

I. Introduction

The Kalman filter is widely employed to estimate the state of dynamical systems from noisy measurements [1]. Its success can be attributed to a combination of two factors: First, it permits the incorporation of statistical information about measurement noise and the process dynamics into the estimation processes, often allowing the construction of good estimates with relatively little measurement data. Second, from a computational perspective, the estimation problem is reduced to a least-squares optimization that can be solved very efficiently. In fact, the Kalman filter actually performs the computations in a recursive fashion so that every new measurement that becomes available improves the previous estimate with relative little computation (on the order of the square of the size of the state).

A weakness of the Kalman filter, which is shared with other algorithms that minimize sums of squared residuals [2], is that it is especially sensitive to impulsive noise, i.e., noise that most of the time is relatively small, but occasionally takes large values. This can be explained by the fact that the Kalman filter is derived under the

L.R. García Carrillo, and J.P. Hespanha are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560, USA. {lrgc, hespanha}@ece.ucsb.edu

W.J. Russell is available via email at: josh@wjruss.com

G. Collins is with Toyon Research Corporation, 6800 Cortona Drive, Goleta, CA 93117. gcollins@toyon.com

This material is based upon work supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-D-0001 from the U.S. Army Research Office.

assumption of Gaussian noise, whose probability density function decays exponentially with the square of the noise, which makes large noise exceptionally unlikely. This problem arises in any least-squares algorithm since squaring the residuals amplifies large deviations that, instead, should be ignored in the presence of impulsive noise [3]. In the derivation of the Kalman filter, one also makes the assumption that the process dynamics are corrupted by Gaussian disturbances, which can be problematic for systems whose dynamics are affected by impulsive disturbances. Under a Gaussian assumption, large disturbances are exceedingly unlikely and the filter forces the estimates to be much smoother than what they should be.

Measurements from global positioning system (GPS) are often used by multi-agent system to determine the position and orientation of individual agents, but when some agents lack GPS information (either permanently or temporarily), one can still obtain information about the agents' positions using alternative sensor modalities, which include vision based-sensors [4, 5], RF sensors [6, 7], and acoustic sensors [8, 9]. It turns out that these sensor modalities typically exhibit impulsive noise: vision-based sensors can exhibit large errors when a visual landmark is temporarily obstructed or mis-interpreted, and RF/acoustic sensors are prone to reporting false measurements due to multi-path reflections. Impulsive disturbances in multi-agent systems arise for agents that generally move along fairly smooth paths but occasionally engage in sharp turns or evasive maneuvers.

In this paper, we seek to overcome the weaknesses of Kalman filtering and least-squares algorithms mentioned above, while still preserving the key features that make such algorithms useful. Our motivation stems from scenarios where one seeks to estimate the state of multi-agent systems with measurements corrupted by impulsive noise and dynamics subject to impulsive disturbances. Key contributions of this work include (i) the development of a maximum likelihood estimator that captures the statistical information about impulsive noise/disturbances in multi-agent systems; (ii) the representation of the multi-agent state estimation problem as a graph that effectively encodes the state of the estimator over time and facilitates the online implementation of the estimation algorithm; and (iii) the demonstration of the effectiveness of our approach both in simulation and in hardware experiments.

Impulsive noise/disturbance can be modelled by the sum of two terms: a Gaussian term that captures small noise that is present at essentially every time instant and a Laplacian term that is often zero (or very small), but that occasionally exhibits large magnitude. We recall that the probability density function (pdf) of the Laplacian distribution is highly peaked at zero but exhibits a tail that is heavier than a Gaussian distribution (cf. Figure 1). We show in Section III that computing the maximum likelihood estimate that results from this model involves minimizing the sum of two appropriately defined terms: one corresponding to an ℓ_2 norm and another to an ℓ_1 norm. The two terms are weighted by a positive term that encodes how much of the noise/disturbances covariance is due to the Gaussian versus the Laplacian components. While it is not possible to solve the resulting optimization in closed form, fast solvers have become available, which allow us to solve the optimization online in just a few milliseconds [10, 12]. This is sufficiently fast to process, e.g., vision measurements at frame rate.

Multi-agent dynamic estimation problems can be conveniently represented by graphs whose nodes correspond to the states of the agents at different time instants and whose edges correspond to the constraints that are either

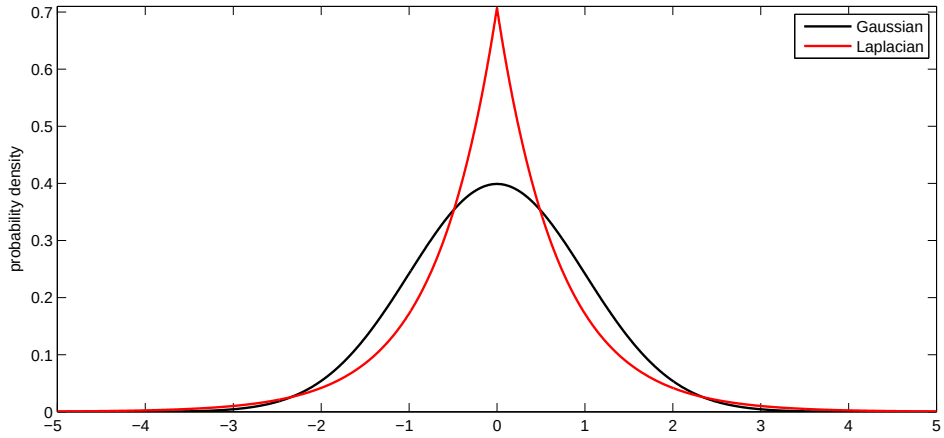


Fig. 1. Probability density functions (pdf) of the Gaussian and Laplacian distributions with zero mean and unit variance. The pdf of the Laplacian distribution is impulsive in the sense that it is highly peaked at zero but falls off more slowly than the Gaussian distribution as the distance from zero increases, exhibiting a heavier tail.

imposed by measurements or by the agents' dynamics (cf. Section II). As time evolves, new nodes are necessary to represent the (unknown) agents' states at the most recent time instants and new edges become available. The latter can be due both to the new measurements and to the constraints imposed by the agents' dynamics. To keep the computation and memory requirements limited, one typically drops nodes and edges associated with agents' states in the far past. The graph structure just described can be viewed as an efficient representation of the state of the estimator that is flexible and convenient for the online implementation of the estimator: as new measurements are collected one simply adds and drops a few nodes/edges, while maintaining most of the graph unchanged.

The estimation approach proposed has been validated through extensive simulations and hardware tests for scenarios where a team of Unmanned Air Vehicles (UAVs) equipped with cameras, GPS, and inertial sensors cooperate to track and geolocate a ground moving target. A summary of these results is presented in Section IV, which confirm that the proposed algorithms are robust to impulsive noise and disturbances. The results also show that one can find parameters for the estimator that perform effectively over wide ranges of impulsive noise/disturbances, which greatly facilitates the use of this approach in practice. The hardware experiments highlight the need to have effective schemes to deal with outliers and clearly show the performance benefits of using multiple UAVs for aerial target tracking.

Related work

Since the pioneering work of [13] on the Least Absolute Shrinkage and Selection operator (LASSO), ℓ_1-norm regularized versions of classical least squares estimators have been widely used to estimate sparse vectors of parameters, i.e., vectors that are known to have a large number of entries equal to zero. LASSO-like estimators

generally do not have closed-form solutions but can be found by solving well conditioned convex optimizations [14], which permits their use in numerous large-scale problems, including denoising [15], image decomposition [16], signal recovery from incomplete measurements [17, 18], sensor selection [19], and fault identification [20].

In the last few years, estimators based on ℓ_1-norm minimizations have also been proposed to deal with non-Gaussian measurement noise and to decide wether borderline observations should be removed from a set of measurements. In [21], the authors present several cost functions derived based on different noise models and show that a cost function based on an ℓ_1-norm metric is not only robust to outliers, but also computationally tractable. The performance of an ℓ_1-norm estimator is also studied in [22], where the problem of estimating a state from noisy and corrupted linear measurements is addressed. The ℓ_1-norm estimator is shown to outperform a Kalman filter, but no results are provided for impulsive disturbances in the process model.

The use of ℓ_1 or a combination of ℓ_1/ℓ_2 criteria to estimate the state of dynamical systems is less common in the literature, but appeared recently in [23] to deal with processes subject to abrupt changes in the state due to impulsive disturbances. Our work builds upon [23], by also considering impulsive noise and providing an efficient graph-based algorithm to produce online estimates for multi-agents systems.

The use of graphs to represent estimation problems arising in sensor networks and multi-agent systems can provide important insights into the quality of the estimates. In fact, it was shown in [24] that the topology of the graph plays a crucial role in how the variance of the estimation errors scales with the number of variables to be estimated. Such graphs are also convenient for the development of distributed estimation algorithms, as in [25], which addresses cooperative localization in mobile sensor networks with intermittent communication, where each agent updates its prediction based on the agents it encounters. Similarly, [26] considers the problem of decentralized Kalman filtering for multi-agent localization in a sensor network, where each sensor node implements a local Kalman filter based on its own measurements and the information exchanged with its neighbors. In [27, 28], the problem of estimating the position of a mobile agent based on a stream of noisy measurements was also reduced to a graph representation that facilitated the development of a distributed estimation algorithms that emulated the performance of a Kalman filter [27] and that explored the graph structure to accelerate convergence [28]. While the references above were mostly focused on least-squares estimation problems arising from Gaussian measurement noise models, graph representations are useful beyond that restrictive setting. In [29], estimation methods under Gaussian noise assumptions are derived, but modifications for Laplacian noise are also considered to model the presence of outliers in sensor networks' localization applications.

II. Problem Formulation

We consider the problem of estimating the state of an N-agent dynamical system of the form

$$\dot{x}_i = A_i\big(u_i(t)\big)x_i + b_i\big(u_i(t)\big) + B_i\big(u_i(t)\big)d_i, \quad \forall i \in \{1, 2, \ldots, N\}, \tag{1}$$

where $x_i \in \mathbb{R}^{n_{xi}}$ denotes the state of agent i, $u_i \in \mathbb{R}^{n_{ui}}$ the control input of agent i, and $d_i \in \mathbb{R}^{n_{ui}}$ an unmeasured disturbance affecting this agent. This type of model captures large classes of vehicle dynamics, ranging from simple

double integrator dynamics in which x_i would contain the agents' position and velocity, to nonlinear 6DOF rigid body dynamics [30]. Over an interval $[\tau, t]$, $0 \leqslant \tau \leqslant t$ during which the control input u_i is known, (1) establishes a constraint between the value of the state x_i at times τ and t through the variation of constants formula

$$x_i(t) = \Phi_i(t,\tau)x_i(\tau) + \bar{b}_i(t,\tau) + \bar{d}_i(t,\tau), \tag{2}$$

where $\Phi_i(t,\tau)$ denotes the state transition matrix of the time-varying homogeneous system $\dot{z} = A_i\big(u_i(t)\big)z$, and

$$\bar{b}_i(t,\tau) \coloneqq \int_\tau^t \Phi_i(t,s)b_i\big(u_i(s)\big)ds, \qquad \bar{d}_i(t,\tau) \coloneqq \int_\tau^t \Phi_i(t,s)B_i\big(u_i(s)\big)d_i(s)ds.$$

Knowledge of u_i permits the computation of $\bar{b}_i(t,\tau)$ in (2), but $\bar{d}_i(t,\tau)$ is unknown since it depends on the unmeasured disturbance d_i and needs to be estimated.

We consider two types of measurements available to the agents: agent-specific measurements of the form

$$y_i(t) = C_i\big(u_i(t)\big)x_i(t) + D_i\big(u_i(t)\big)n_i(t) \tag{3}$$

and inter-agent relative measurements of the form

$$y_{ij}(t) = C_i\big(u_i(t)\big)x_i(t) - C_j\big(u_j(t)\big)x_j(t) + D_{ij}\big(u_i(t),u_j(t)\big)n_{ij}(t), \tag{4}$$

where the terms n_i in (3) and n_{ij} in (4) represent measurement noise.

Our goal is to construct estimates of the states x_i, $i \in \{1,2,\ldots,N\}$ of the different agents based on

1) a set of motion constraints of the form (2) for different intervals (τ,t) for which $u_i(s)$, $\forall s \in [\tau,t)$ is known, and
2) a set of measurement equations of the form (3)–(4) for a set of times t for which measurements are available.

We are interested in estimation problems that are *asynchronous* in the sense that the measurements (3)–(4) produced by the agents' sensors may not be generated in a periodic fashion and, even if periodic, the different agents may not use the same period. Nevertheless, we do assume that the measurements are associated with time stamps that indicate when the measurement was collected. We are also interested in *heterogeneous* problems, where only some agents may have the ability to generate absolute measurements like (3), and the relative measurements like (4) may only be available between some pairs of agents. Moreover, which measurements are available may change over time. This level of generality is crucial for implementation in real multi-agent systems because different measurements are generally produced by sensors that operate independently and at different rates. Vision-based sensors are especially prone to asynchronous operation since the image processing software often operates at a variable frame rate.

For simplicity of notation, in the remainder of the paper we assume that all measurements (3)–(4) are actually of the form (4) by introducing a *reference agent* $x_0(t) \coloneqq 0$, $\forall t \geqslant 0$.

A. Graph Representation of the Multi-Agent System

The multi-agent estimation problem formulated above can be conveniently represented by a directed graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$, like the one shown in Figure 2. Each node in the *node set* $\mathcal{V}$ corresponds to the state $x_i(t_k)$ of some agent

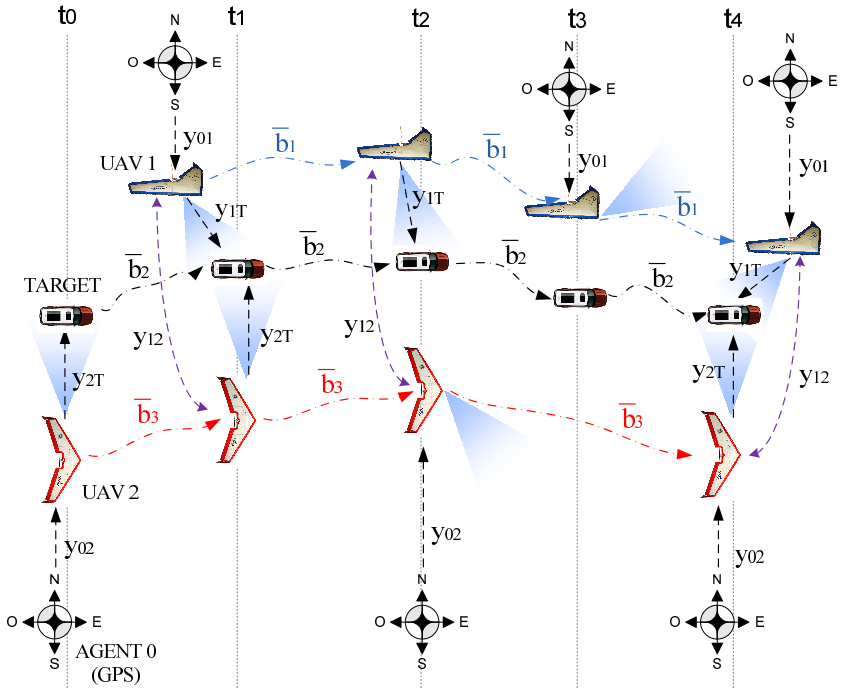


Fig. 2. A target tracking scenario considering a pair of UAVs (labeled UAV 1 and UAV 2) cooperatively tracking a target vehicle (labeled TARGET) moving on ground, with respect to a reference coordinate system (labeled AGENT 0). The estimation problem can be represented by a graph where the nodes correspond to the states x_i, $i \in \{0, 1, 2, 3\}$ of the four agents AGENT 0, UAV 1, UAV 2, TARGET at 5 consecutive time instants $\{t_0, t_1, t_2, t_3, t_4\}$. The edges of the graph correspond to measurements (dashed arrows) and models contraints (dash-dotted arrows). This example highlights the ability to represent heterogenous sensing: at times t_1, t_4 both UAVs acquire relative measurements between their positions and the target, at time t_0 only UAV 2 can "see" the target, and at time t_3 none of the UAVs "sees" the target; the UAVs only "see" each other at times t_1, t_2, t_4.

$i \in \{0, 1, \ldots, N\}$ at some time $t_k \geqslant 0$ that appears in one of the equations (2) or (4). Since the state of each agent i typically appears in several equations (2), (4) at different time instants, many nodes in $\mathcal{V}$ are typically associated with the same agent, each corresponding to a distinct time instant. Only the reference agent x_0 corresponds to a single node, since its state has the same (zero) value at every time instant.

Each edge $(\mu, \nu) \in \mathcal{V} \times \mathcal{V}$ in the *edge set* $\mathcal{E}$ is associated with an *edge equation* like (2) or (4) that can be written generically as

$$z = M\mu - N\nu + \epsilon. \tag{5}$$

For edges associated with equations of the form (2), the variables μ and ν correspond to the nodes in $\mathcal{V}$ that are associated with the states $x_i(\tau)$ and $x_i(t)$ of the same agent i at the two time instants τ and t, respectively. In this case, the edge equation (5) matches with (2) using the following associations

$$z = \bar{b}_i(t, \tau), \qquad M = I_{n_{xi} \times n_{xi}}, \qquad \mu = x_i(t), \qquad N = \Phi_i(t, \tau), \qquad \nu = x_i(\tau), \qquad \epsilon = -\bar{d}_i(t, \tau).$$

For edges associated with equations of the form (4), the nodes μ and ν correspond to the nodes in $\mathcal{V}$ that are

associated with the states $x_i(t)$ and $x_j(t)$ of two agents i and j, respectively, at the same time instant t. In this case, the edge equation (5) matches with (4) using the following associations

$$z = y_{ij}(t), \quad M \coloneqq C_i\big(u_i(t)\big), \quad \mu = x_i(t), \quad N = C_j\big(u_j(t)\big), \quad \nu = x_j(t), \quad \epsilon = D_{ij}\big(u_i(t), u_j(t)\big)n_{ij}(t).$$

In either case, the node variables μ and ν always correspond to quantities that we want to estimate, the variable z to a quantity that can be computed or directly measured, and the variable ϵ to unmeasured disturbances or noise. The matrices M and N are also known or computable.

Stacking all the unknown node variables in a *node vector* $\mathbf{x}$, all the z vectors that appear in the edge equations (5) in a vector $\mathbf{z}$, and all the ϵ vectors that appear in the edge equations (5) in a vector $\boldsymbol{\epsilon}$, we can write all the edge equations as

$$\mathbf{z} = H\mathbf{x} + \boldsymbol{\epsilon}. \tag{6}$$

The reference agent $x_0(t) = 0$, $\forall t \geqslant 0$ does not need to be introduced in the node vector $\mathbf{x}$ since this node variable is known. We shall refer to (6) as the *measurement model equation*, even though it encodes both the measurement equations (3)–(4) and the motion constraints equation (2). Consistently, the vectors $\mathbf{z}$ and $\boldsymbol{\epsilon}$ will be referred to as the *measurement vector* and the *noise vector*, respectively.

For simplicity of presentation, unless otherwise noted, we shall assume that the noise vector $\boldsymbol{\epsilon}$ has zero mean and its covariance matrix is equal to the identity matrix, i.e., all its entries are uncorrelated and have unit variance. If this was not the case and we had

$$\mathrm{E}[\boldsymbol{\epsilon}] = \mu, \qquad \mathrm{E}[(\boldsymbol{\epsilon} - \mu)\,(\boldsymbol{\epsilon} - \mu)'] = P, \tag{7}$$

we could define a normalized error

$$\bar{\boldsymbol{\epsilon}} = P^{-\frac{1}{2}}(\boldsymbol{\epsilon} - \mu)$$

for which we would have

$$\mathrm{E}[\bar{\boldsymbol{\epsilon}}] = 0, \qquad \mathrm{E}[\bar{\boldsymbol{\epsilon}}\,\bar{\boldsymbol{\epsilon}}'] = P^{-\frac{1}{2}}\,\mathrm{E}[(\boldsymbol{\epsilon} - \mu)\,(\boldsymbol{\epsilon} - \mu)']P^{-\frac{1}{2}} = I,$$

and we could re-write the measurement equation in terms of this normalized error as

$$\mathbf{z} = H\mathbf{x} + \boldsymbol{\epsilon} = H\mathbf{x} + P^{\frac{1}{2}}\bar{\boldsymbol{\epsilon}} + \mu \quad \Leftrightarrow \quad P^{-\frac{1}{2}}(\mathbf{z} - \mu) = P^{-\frac{1}{2}}H\mathbf{x} + \bar{\boldsymbol{\epsilon}}, \tag{8}$$

which is still of the form (6), provided that we replace $\mathbf{z}$ by $P^{-\frac{1}{2}}(\mathbf{z} - \mu)$ and H by $P^{-\frac{1}{2}}H$.

In scenarios where new measurements are constantly being generated, the graph-based representation of the multi-agent estimation problem described above is especially convenient for real-time online implementations. Each new measurement typically requires the addition of a graph edge (corresponding to the new measurement) and one or two nodes (corresponding to state(s) of the agent(s) involved, at the time the measurement was made). However, all remaining graph nodes and edges remain the same, which greatly facilitates maintaining appropriate data structures

to represent the data needed to solve the estimation problem. In practice, the addition of new graph edges and nodes corresponds to the adding new rows and columns, respectively, to the matrix H in (6). To keep the computation and memory requirements limited, one typically removes from the graph edges and nodes corresponding to old data, which amounts to removing rows and columns to the matrix H.

III. Maximum Likelihood Estimation for Impulsive Noise

We are interested in problems where the noise vector $\boldsymbol{\epsilon}$ can be decomposed as the sum of two independent components: one that is essentially Gaussian and another that is impulsive in the sense that its distribution is highly peaked at zero, but that falls off more slowly than the Gaussian distribution as the distance from zero increases. The latter components permits the modeling of "outliers" that are zero (or very small) most of the time, but that occasionally take large values. Specifically, we assume that the noise vector can be expressed as

$$\boldsymbol{\epsilon} = \lambda \mathbf{n} + (1-\lambda)\boldsymbol{\ell}, \tag{9}$$

where each entry $\mathbf{n}_i$ of $\mathbf{n}$ is a standard Gaussian random variable with zero mean and unit variance, each entry $\boldsymbol{\ell}_i$ of $\boldsymbol{\ell}$ is a Laplace random variable also with zero mean and unit variance, and λ is a constant in the interval $(0,1)$. All the $\mathbf{n}_i$ and $\boldsymbol{\ell}_i$ are assumed independent so that the covariance matrices of $\mathbf{n}$, $\boldsymbol{\ell}$, and $\boldsymbol{\epsilon}$ are all equal to the identity matrix and the probability density functions (pdfs) of $\mathbf{n}$ and $\boldsymbol{\ell}$ are equal to

$$f_{\mathbf{n}}(n) = (2\pi)^{-n_\epsilon/2} e^{-\frac{1}{2}\sum_i n_i^2} = (2\pi)^{-n_\epsilon/2} e^{-\frac{1}{2}\|n\|_2^2}, \qquad f_{\boldsymbol{\ell}}(\ell) = 2^{-n_\epsilon/2} e^{-\sqrt{2}\sum_i |\ell_i|} = 2^{-n_\epsilon/2} e^{-\sqrt{2}\|\ell\|_1},$$

respectively, where n_ϵ is the dimension of the vector $\boldsymbol{\epsilon}$, and $\|\cdot\|_2$ and $\|\cdot\|_1$ denote the ℓ_2 and ℓ_1 norms, respectively.

For $\lambda \approx 1$, the entries of $\boldsymbol{\epsilon}$ are essentially Gaussian, whereas for λ close to 0 they essentially follow the much more impulsive Laplacian distribution. Figure 1 shows a comparison between the distributions of the entries of $\mathbf{n}$ versus those of $\boldsymbol{\ell}$.

We recall that the components of $\boldsymbol{\epsilon}$ are of the form:

$$\bar{d}_i(t,\tau) \coloneqq \int_\tau^t \Phi_i(t,s) B_i\big(u_i(s)\big) d_i(s) ds, \qquad D_i\big(u_i(t)\big) n_i(t), \qquad D_{ij}\big(u_i(t), u_j(t)\big) n_{ij}(t),$$

in the first case arising from unmeasured disturbances that affect the agent's dynamics (1), and in the latter cases arising from the measurement equations (3)–(4). Impulsiveness in the $\bar{d}_i(t,\tau)$ term permits the modeling of agents that typically move with fairly constant velocity but occasionally exhibit strong accelerations (e.g., dues to sharp turns or evasive maneuvers). Impulsiveness in the absolute measurements $D_i\big(u_i(t)\big) n_i(t)$ permits, e.g., the modeling of sudden loss of GPS accuracy dues to a satellite loss. Finally, impulsiveness in the relative measurements $D_{ij}\big(u_i(t), u_j(t)\big) n_{ij}(t)$ permits the modeling of impulsive noise, e.g., due to the false detections in a computer vision algorithm that is used to obtain relative position measurements. All these source of impulsive noise/disturbances will be present in the application problem discussed in Section IV.

A. Maximum Likelihood Estimator

For a given value x of the node vector $\mathbf{x}$ and realizations z, n, and ℓ of the random variables $\mathbf{z}$, $\mathbf{n}$, and $\boldsymbol{\ell}$, respectively, the joint likelihood function $f_{\mathbf{z},\mathbf{n},\boldsymbol{\ell}|\mathbf{x}}$ of $\mathbf{z}$, $\mathbf{n}$, and $\boldsymbol{\ell}$ in (6), (9) can be computed using Bayes' rule according to the formula

$$f_{\mathbf{z},\mathbf{n},\boldsymbol{\ell}|\mathbf{x}}(z,n,\ell|x) = \begin{cases} f_{\mathbf{n}|\boldsymbol{\ell},\mathbf{x}}(n|\ell,x) f_{\boldsymbol{\ell}|\mathbf{x}}(\ell|x) & z = Hx + \lambda n + (1-\lambda)\ell \\ 0 & z \neq Hx + \lambda n + (1-\lambda)\ell, \end{cases}$$

where $f_{\mathbf{n}|\boldsymbol{\ell},\mathbf{x}}$ denotes the conditional pdf of $\mathbf{n}$ given $\boldsymbol{\ell}$ and $\mathbf{x}$ and $f_{\boldsymbol{\ell}|\mathbf{x}}$ the conditional pdf of $\mathbf{n}$ given $\mathbf{x}$. However, since $\mathbf{n}$, $\boldsymbol{\ell}$, and $\mathbf{x}$ are independent, we have that $f_{\mathbf{n}|\boldsymbol{\ell},\mathbf{x}} = f_{\mathbf{n}}$ and $f_{\boldsymbol{\ell}|\mathbf{x}} = f_{\boldsymbol{\ell}}$, leading to

$$f_{\mathbf{z},\mathbf{n},\boldsymbol{\ell}|\mathbf{x}}(z,n,\ell|x) = \begin{cases} f_{\mathbf{n}}(n) f_{\boldsymbol{\ell}}(\ell) = (4\pi)^{-n_\epsilon/2} e^{-\frac{1}{2}\|n\|_2^2 - \sqrt{2}\|\ell\|_1} & z = Hx + \lambda n + (1-\lambda)\ell \\ 0 & z \neq Hx + \lambda n + (1-\lambda)\ell. \end{cases}$$

This means that maximum likelihood estimators for $\mathbf{x}$, $\mathbf{n}$, and $\boldsymbol{\ell}$ can be obtained by solving the following optimization

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|n\|_2^2 + \sqrt{2}\|\ell\|_1 \\ \text{with respect to} & x, n, \ell \\ \text{subject to} & z = Hx + \lambda n + (1-\lambda)\ell. \end{array}$$

By eliminating the optimization variable n using the constraint $z = Hx + \lambda n + (1-\lambda)\ell$ and making the change of optimization variable $\bar{\ell} \coloneqq (1-\lambda)\ell$, we obtain the following equivalent sum-of-norms optimization that can also be used to obtain the maximum likelihood estimator.

$$\begin{array}{ll} \text{minimize} & \frac{1}{2\lambda^2}\left\|Hx - z + \bar{\ell}\right\|_2^2 + \frac{\sqrt{2}}{1-\lambda}\|\bar{\ell}\|_1 \\ \text{with respect to} & x, \bar{\ell}. \end{array} \tag{10}$$

Hereafter, we refer to (10) as the *sum-of-norms* estimator. A global solution to this minimization can be efficiently computed due to the convexity of the objective function. In fact, its special structure allows it to be solved in $O(m)$ operations, permitting its online solution even for large values of m (number of measurements) [23].

The sum-of-norms optimization in (10) favors impulsive solutions for the vector $\bar{\ell} = (1-\lambda)\ell$, i.e., solutions for which many/most of the entries of ℓ are equal to zero. In fact, one can show that there is a value $\lambda_{\max} \in (0,1)$ for which the estimated impulsive noise is identically zero if and only if $\lambda \geqslant \lambda_{\max}$ [14]. In other words, $\lambda_{\max}$ provides a threshold above which the estimated impulsive noise ℓ becomes zero.

Remark 1: If the measurement vector $\boldsymbol{\epsilon}$ had mean μ and covariance P, as in (7), in view of (8), we would solve

$$\begin{array}{ll} \text{minimize} & \frac{1}{2\lambda^2}\left\|P^{-\frac{1}{2}}Hx - P^{-\frac{1}{2}}(z-\mu) + \bar{\ell}\right\|_2^2 + \frac{\sqrt{2}}{1-\lambda}\|\bar{\ell}\|_1 \\ \text{with respect to} & x, \bar{\ell}. \end{array}$$

instead of (10). □

B. Solution algorithms and software

Convex optimization problems can be solved efficiently with theoretical performance guarantees and well-developed practical methods and tools. Among the tools available, YALMIP [31] and CVX [32, 33] are especially appealing because they greatly simplify the process of specifying and solving convex optimizations. These tools are ideal for prototyping algorithms based on convex optimization and can readily handle sum-of-norms optimizations. Unfortunately, these general purpose tools can be fairly slow, leading to optimization times measured in seconds or even minutes, which precludes their use in online in real-time systems. Furthermore, they require extensive use of libraries and commercial software, making them unsuitable for some embedded applications.

For the results reported in this paper, we used CVXGEN [10] for solving the sum-of-norms optimization (10). Using an online interface (http://www.cvxgen.com), CVXGEN takes a high-level description of a convex optimization problem and automatically generates flat, library-free C-code that can be compiled into a high-speed custom solver. For small and medium sized problems (with up to hundreds of optimization variables), CVXGEN generates solvers with optimization times measured in microseconds or milliseconds. The optimization problem is specified through a Matlab-like programming language that includes problem dimensions, parameters, variables, cost functions, and constraints. The generated C-code solver takes as inputs the problem specific parameters and returns optimal values of the optimization variables. For the specific case under consideration, the optimization variables correspond to the vector of the agent states x and the Laplacian noise variables $\bar{\ell}$; whereas the parameters are the weight λ, the matrix H, and the measurement vector z. The Open Source Computer Vision Library (OpenCV) [11] was used to perform linear algebra operations external to the C-code solver (matrix and vector multiplications, matrix inversions, matrix and transposing). The resulting C-coded optimization requires about 2ms per iteration in a Windows 7 laptop PC equipped with an Intel Core i5 CPU @ 2.4 GHz and 4 GB of RAM memory, amply permitting the real-time generation of estimates at frame rate.

An alternative fast solver has been introduced in [12], where the authors present efficient interior point methods tailored to convex multistage problems and superior numerical stability. While not required for this application, [12] allows for quadratic constraints, which are not supported by other fast solvers. For a detailed discussion regarding why custom solvers like [10] and [12] can outperform general purpose solvers the reader is referred to [34].

IV. Simulation and Hardware Results

The estimation approach developed in the previous sections was tested in simulation and in hardware for a scenario where a team of UAVs equipped with vision systems and navigation sensors try to geolocate and track a target vehicle moving on the ground. Much like in the scenario depicted in Figure 2, we regard the ground moving target and all the UAVs as agents whose positions we want to estimate.

The UAVs determine the relative position of the target with respect to their own positions using image-plane measurements provided by onboard video sensors. The image plane measurements are projected into the earth's surface using the camera's intrinsic and extrinsic parameters to obtain 3-dimensional relative positions. These

measurements correspond to inter-agent relative measurement equations of the form (4). Most of the time, the noise in these equations is relatively small, but false detections (outliers) due to image processing errors create impulsive noise.

The UAVs positions and orientations with respect to an earth fixed coordinate frame are obtained from a combined GPS/IMU unit operating at a 25 Hz rate, corresponding to agent-specific measurement equations of the form (3). Also here, the noise is often small, but sometimes GPS unavailability produces large errors and impulsive noise.

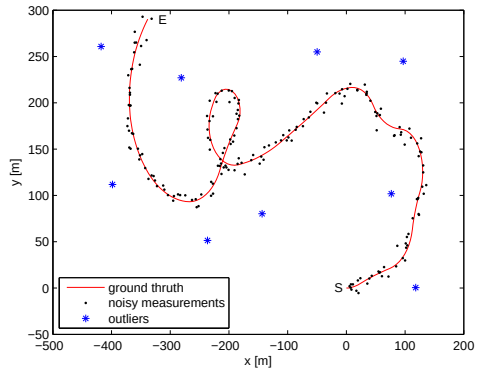
While the UAV dynamics can be written as (1), they are afflicted by fairly large disturbance (mostly due to wind). In contrast, the GPS/IMU measurement are fairly good, so we chose to ignore the motion constraints (2) for the UAVs — in essence we did not include such edges in the graph. For the target motion, we cannot afford to do this because the UAV-to-target relative position measurements are noisy with many outliers and one needs a target model to obtain reasonable estimates. For the target dynamics (1), we used a double integrator driven by a noise process that is independent in the x and y directions. This noise was taken to be impulsive to capture the fact that most of the time the target maintains a fairly constant velocity, but occasionally undergoes sharp turns.

Both for the simulation and the hardware results, the measurements taken by the UAVs (absolute and relative) were collected in an asynchronous fashion at a rate of about 10Hz, limited by the maximum frame rate supported by the computer vision system. For the purpose of estimating the target position, every time a new measurement become available, a graph like the one in Figure 2 was constructed based on the lastest 15 UAV-to-target relative position measurements, which could all come from the same UAV or from multiple UAVs, depending on which UAVs had the target in their field of view.

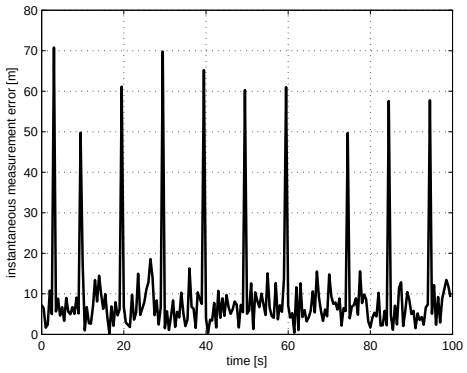
A. Synthetic-data simulation results

For the simulation results discussed below, the ground target motion was generated according to a kinematic unicycle model [35]. The simulation environment used for the generation of the synthetic data corresponds to a rectangular area of $600m \times 300m$, in the x and y dimensions, respectively. The UAV-to-target relative position measurements were contaminated with zero-mean Gaussian noise with a standard deviation of around $10m$. Additionally, to simulate the presence of outliers, some of the measurements were corrupted with uniformly distributed false detections, producing noises ranging from $40m$ to $70m$. We chose not to use Laplacian noise in the synthetically generated data to demonstrate the robustness of our algorithms with respect to the modeling assumptions used to derive the maximum-likelihood estimator.

The simulations were performed under five different percentages of outliers, representing 1%, 5%, 10%, 15%, and 20% of the total number of measurements. The RMS sensor measurement errors corresponding to each one of those cases are $9.73m$, $15.48m$, $20.56m$, $22.99m$, and $35.24m$, respectively. For illustrative purposes, one of these scenarios is shown in Figure 3.



(a) The ground truth of the target displacement is represented by the solid line staring at $(x, y) = (0m, 0m)$ at time $t = 0$ and ending at $(x, y) = (-338m, 290m)$ at time $t = 100$sec. The noisy measurements are represented by black dots and the $*$ symbols represent the outliers, i.e., erroneous measurements wrongly identified to be the target.



(b) Instantaneous sensor measurement error. The target position measurements are corrupted with zero-mean Gaussian noise with a standard deviation of around $10m$. The measurement error exhibits impulsive noises ranging from $40m$ to $70m$, which are associated with corrupted measurements. In the present scenario, the outliers represent 5% of the total number of measurements, and are clearly visible as instantaneous measurement error spikes.

Fig. 3. Simulation environment used for the generation of the synthetic data showing a target sample path 3(a) and the corresponding instantaneous measurement error 3(b).

1) Selection of the weight parameter λ: The sum-of-norms estimator (10) depends on the parameter λ introduced in (9) to characterize the relative weights of the Gaussian and Laplace (impulsive) components of the noise. Figure 4 shows the value of the RMS estimation error as a function of λ for different percentages of outliers in the UAV-to-target relative position measurements. The crucial observation to be made regarding this figure is that, while the RMS estimation error varies with λ, the RMS curves are relatively flat close to their minimum and values of λ close to 0.02 provide excellent values for the RMS estimation error for a wide range of outlier percentages. Notice that for $\lambda \geqslant 0.3$ the RMS estimation errors do not depend on λ, which is consistent with the conclusions of [14].

2) Comparison with conventional estimation methods : To demonstrate the benefits of the sum-of-norms estimator, we compare it with a Kalman filter and a Batched Least Squares (BLS) estimator under the same target tracking scenario. The BLS estimator uses the same 15-measurement window as the sum-of-norms estimator, whereas the (recursive) Kalmar filter uses all measurements collected up to the present time. Figure 5 shows typical simulation results comparing the instantaneous estimation errors obtained using the three methods for 20% of outliers measurements. A quantitative comparison of the three methods for different levels of outlier measurements is summarized in Table I. We can see both from the plots in Figure 5 and the results in Table I that the sum-of-norms estimator significantly outperforms the two other estimators. The benefit improves as the percentage of outliers grows, but it is already very noticeable for 5% of outliers measurements. Aside from reducing the RMS estimation error, we can see in Figure 5 that the sum-of-norms estimator is especially effective at reducing the

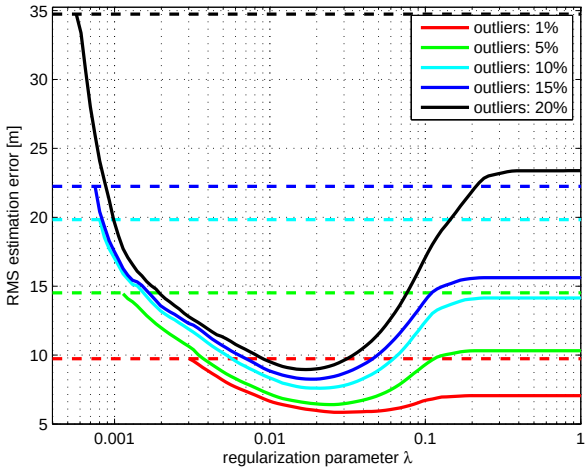


Fig. 4. RMS estimation error obtained using the sum-of-norms estimator versus the parameter λ that characterizes the relative weights of the Gaussian and Laplace (impulsive) components of the noise, for different percentages of outlier measurements. The regularization parameter λ providing the best results for 1%, 5%, 10%, 15%, and 20% of outlier measurements are $\lambda = 0.0276$, $\lambda = 0.0240$, $\lambda = 0.0209$, $\lambda = 0.0182$, $\lambda = 0.0170$, respectively. The dashed lines represent the raw RMS sensor measurement error averaged over time.

TABLE I

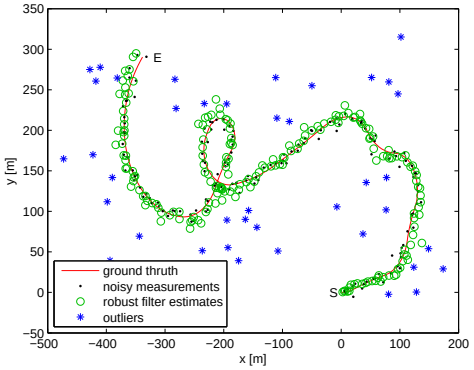
COMPARISON OF THE RME ERROR FOR A KALMAN FILTER, A BATCHED LEAST SQUARE ESTIMATOR, AND THE PROPOSED SUM-OF-NORMS ESTIMATOR ALGORITHM UNDER DIFFERENT CHOICES OF λ. EACH COLUMN HAS ONE ENTRY HIGHLIGHTED IN BOLDFACE, WHICH REPRESENTS THE RMS ESTIMATION ERROR OBTAINED FOR THE ROBUST FILTER WITH THE BEST VALUE FOR THE PARAMETER λ.

	1% **outliers**	5% **outliers**	10% **outliers**	15% **outliers**	20% **outliers**
RMS measur. error	9.7330m	15.4845m	20.5564m	22.9942m	35.2351m
Kalman RMS estim. error	6.8656m	10.6555m	14.3670m	15.9173m	23.8675m
BLS RMS estim. error	7.0527m	10.6061m	14.3661m	15.8792m	23.5845m
sum-of-norms RMS estim. error $\lambda = 0.0276$	**5.8470 m**	6.5350m	7.8242m	8.7256m	9.6274m
sum-of-norms RMS estim. error $\lambda = 0.0240$	5.8999m	**6.4056 m**	7.6994m	8.5111m	9.3299m
sum-of-norms RMS estim. error $\lambda = 0.0209$	5.9540m	6.4782m	**7.5963 m**	8.3933m	9.1502m
sum-of-norms RMS estim. error $\lambda = 0.0182$	6.0405m	6.5231m	7.6376m	**8.2533 m**	9.0623m
sum-of-norms RMS estim. error $\lambda = 0.0170$	6.0766m	6.5535m	7.6434m	8.3368m	**8.9530 m**

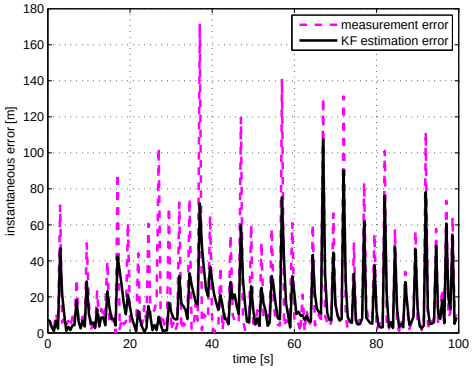
peak estimation error from around 100m for the Kalman filter and the BLS estimator to about 26.6m for the sum-of-norms estimator.

B. Experimental Validation

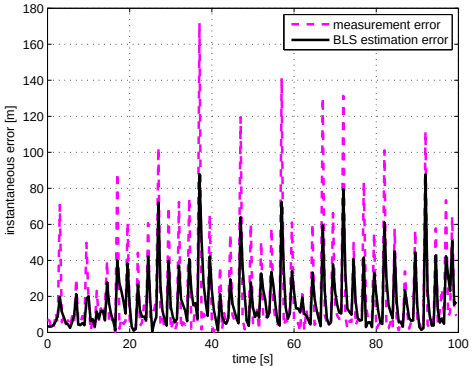
Real-time experimental results were obtained using a team of two UAVs, each equipped with an onboard video camera, inertial sensors, and GPS. The UAVs used video measurements to cooperatively detect, geolocate, and track the ground-moving target (see Figure 6).



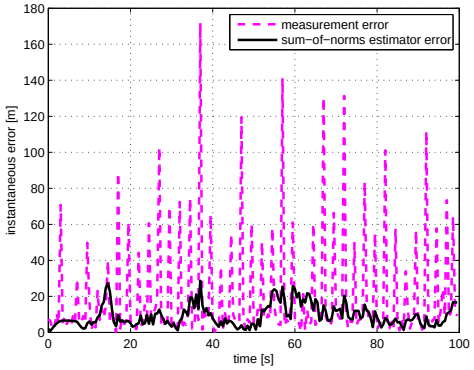
(a) Target sample path with 80% small-noise measurements (solid dots), 20% of outliers ("*" symbols), and the resulting sum-of-norms estimates with $\lambda = .0170$ (green circles).



(b) Instantaneous estimation error obtained using a Kalman filter.



(c) Instantaneous estimation error obtained using a BLS estimator.



(d) Instantaneous estimation error obtained using the sum-of-norms estimator with $\lambda = .0170$.

Fig. 5. Performance comparison of a Kalman filter, a Batched Least Squares (BLS) estimator, and a sum-of-norms estimator based on simulated data. The sum-of-norms produces estimation errors that are significantly smaller than those obtained by the other methods.

1) UAV Ground Control Software and Vision Processing: Two small Unicorn UAVs (named "Yellow" and "Blue") were used in the hardware experiments, controlled by the experimental system shown in Figure 7. Toyon's

TABLE II

UAV AND VISION SENSOR SPECIFICATIONS.

Fixed Wing Unicorn UAV	
Endurance	$\sim$ 60 min
Wingspan	1.5m
Speed	50–65 km/hr
Weight	$\sim$ 2kg incl. batts & payload

Unicorn Payload	
Sensor	1/3" global shutter CMOS
Resolution	752×480
FOV	Hor: 34° $\times$ Ver: 22° (adj.)
FOR	170° Az. $\times$ 90° El.
Weight	45 grams

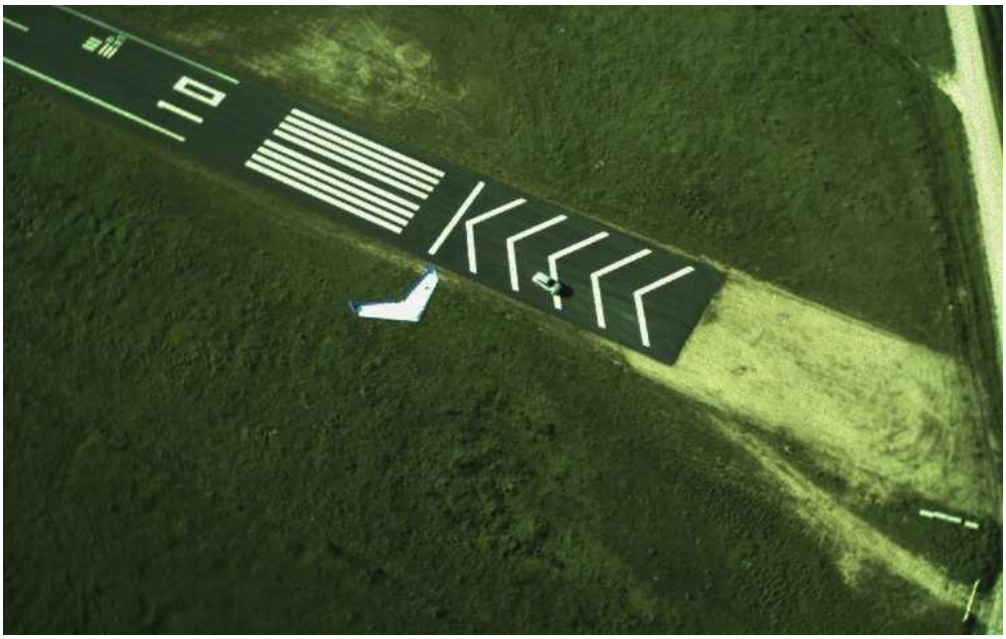


Fig. 6. One UAV tracking the target, as imaged by the other UAV flying at a higher altitude.

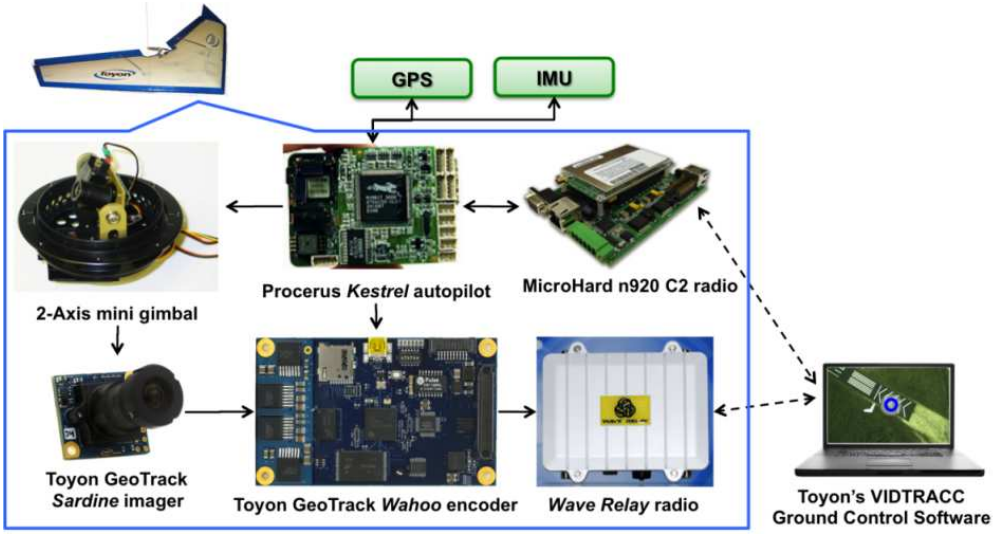


Fig. 7. Toyon's experimental UAV system is based on the Unicorn electric flying wing. It employs a 900 MHz C2 link that relays controls and telemetry to and from the autopilot. The autopilot accepts waypoint or roll commands from the Ground Control Station (GCS) and actuates the aircraft controls to execute these commands. The autopilot also commands a 2-axis gimbal to maintain the target in the sensor FOV. Toyon's micro digital video system provides MJPEG or H.264 video to the ground-based vision processing software via a long-range 2.4 GHz radio.

Video Interface Detection, Tracking, Routing, And Control Center (VIDTRACC) software runs on the Ground Control Station (GCS) computer. This software receives the video streams from both UAVs and employs proprietary computer vision algorithms to automatically detect and track moving targets in video, as shown in Figure 8. Images

from Toyon's micro digital video system are time-stamped using the GPS clock with millisecond resolution and synchronized with the UAV state telemetry. VIDTRACC then uses terrain data and telemetry from the UAV and sensor to geolocate each image-plane detection on the earth. The georegistered target detections were fed to the sum-of-norms estimator, which fused detections to generate a target state estimate in real time.

Fig. 8. Toyon's VIDTRACC ground control software performs real-time automatic target detection and tracking on the video streams from both UAVs. The image on the left shows only one detection (a red polygon) on the target. The image on the right shows false detections resulting from the increased detection sensitivity parameters.

The small, lightweight GPS and IMU sensors that must be used on micro UAVs are not very accurate. Errors in the telemetry data produced by these sensors are primarily responsible for generating Gaussian noise in the georegistered target detections, as shown in Figure 9. Enabled by the use of the sum-of-norms estimator, the sensitivity of Toyon's vision processing software was increased beyond normal thresholds to make the software extremely sensitive to detecting targets. This improved the software's ability to detect very small, dim, or slow-moving targets, at the expense of generating more false detections, as shown in Figure 8. These false detections were the primary source of impulsive disturbances in the data. This configuration stressed the sum-of-norms estimator's ability to handle significant levels of outlier measurements.

2) UAV-to-target measurement model: Measurements of the by each UAV are first obtained in the image plane in pixel coordinates. The pixel coordinates of each detection are then georegistered to the surface of the Earth using the camera intrinsic parameters, the UAV and camera orientation, and terrain data. This process is depicted in Figure 10 and described in more detail in [36]. The error associated with the UAV-to-target relative position measurements are dominated by errors in the UAV orientation [37]. Additional errors are introduced by false detections in the image plane, of trees, UAV wings or landing gear, etc., resulting in impulsive noise.

3) Experimental Results: A set of experiments were conducted in November and December 2012, at the Center for Interdisciplinary Remotely-Piloted Aircraft Studies (CIRPAS) in Camp Roberts, California. Toyon Research Corporation operated the UAV systems and ground control software described in Section IV-B1. The Yellow and Blue Unicorn UAVs were launched and tasked to cooperatively follow, image, and track a vehicle as it maneuvered along the roads of Camp Roberts. The UAV paths were computed by Toyon's Sensor Guided Flight (SGF) algorithm

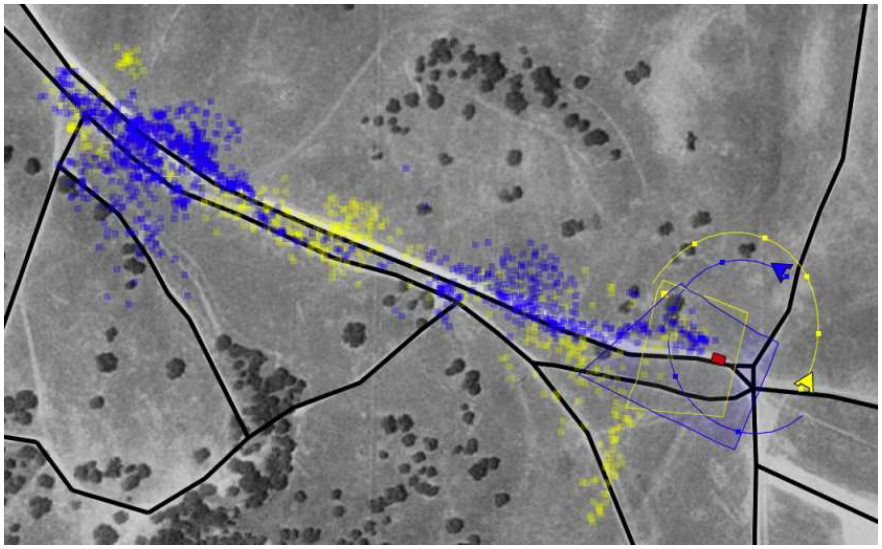


Fig. 9. Target detections are georegistered on the Earth surface and shown on a map view as blue or yellow dots, depending on which UAV collected the image on which the detection was made.

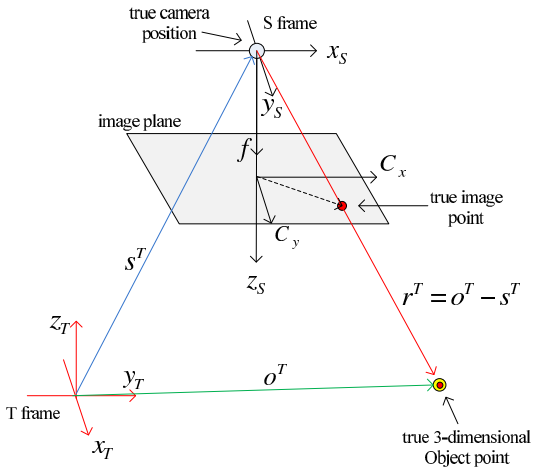


Fig. 10. UAV-to-target measurement model, where S denotes the camera frame on-board the UAV and T an earth fixed coordinate frame. The mapping from the target position to image coordinates is given by the perspective transformation.

[38] that uses receding horizon control to continuously optimize the sensor-to-target viewing geometry. The UAVs were flown at a nearly constant airspeed that was about twice that of the ground vehicle's speed, forcing them to periodically loop around the vehicle to stay close to it. A GPS unit onboard the target vehicle provided ground truth; this data was used to route the UAVs and to compute the error metrics shown in Figure 11.

The sum-of-norms estimator was connected to Toyon's VIDTRACC software over a TCP/IP connection to receive all georegistered measurement data in real time. Our algorithm estimated the target position using the sum-of-norms estimator (10) with the parameter λ set to 0.0209, which consistently provided good performance in our simulations

(see Table I).

Figure 11 shows data obtained from one of our hardware experiments. Even for a single UAV the sum-of-norms estimation errors are typically much smaller than the errors present in the raw sensor data. Occasionally, the estimation errors increase when the raw measurements contain many outliers and/or the UAV-to-target viewing geometry amplifies the errors (e.g., when the UAV is far from the target). When data from both UAVs is fused to produce the target estimates, one consistently obtains very small estimation errors because it becomes highly unlikely that both UAVs have bad viewing geometries and produce outliers at the same time. These results also illustrate the benefits of using multiple UAVs for target tracking.

V. Conclusions and Future Work

We have proposed new algorithms for the state estimation of multi-agent systems under impulsive noise and disturbances. These algorithms correspond to a maximum likelihood estimator with noise and disturbances modeled by a mixture of Gaussian and Laplacian terms. The estimates are computed by solving online optimizations that can be encoded as graphs. The effectiveness of the algorithms was demonstrated — both in simulations and in hardware experiments — in scenarios where a team of UAVs tracked a ground moving target with vision and navigation sensors.

A research item that was not addressed here is the development of distributed versions of the algorithms proposed. For the target tracking scenario considered here, the need for a distributed algorithm did not arise because all UAV data was being processed by a central GCS. However, extending this scenario to larger teams of UAVs tracking multiple targets could overload the wireless bandwidth and computational resources of a single GCS, forcing the need for decentralized solutions. The development of decentralized algorithms was one of the original motivations to consider graph-based representations of estimation problems and will likely also prove very useful for multi-agent systems with impulsive noise/disturbances. Another direction for future work includes the development of adaptive methods to adjust the regularization parameter λ, as the number of outliers in the measurements varies.

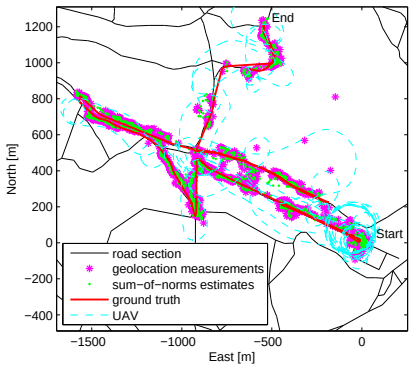
References

[1] R.E. Kalman, *A New Approach to Linear Filtering and Prediction Problems*, Transactions of the ASME–Journal of Basic Engineering, vol. 82, issue D, pp. 35-45, 1960. (cited in p. 1)

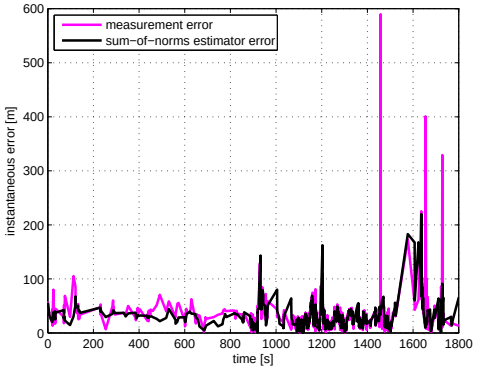
[2] A.J. Pope, *The statistics of residuals and the detection of outliers*, NOAA Technical Report NOS 65 NGS 1, Geodetic Survey, Rockville Md., May 1976. (cited in p. 1)

[3] S.A. van De Geer, *Least Squares Estimation*, vol. 2, pp. 1041-1045, Encyclopedia of Statistics in Behavioral Science, John Wiley & Sons, Ltd, Chichester, 2005. (cited in p. 2)

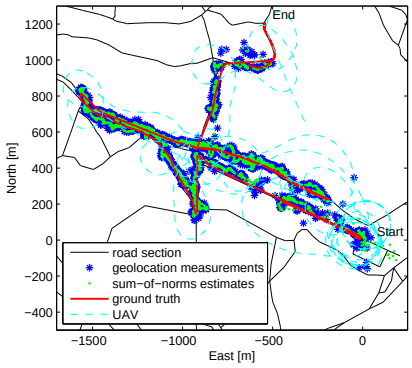
[4] K.J. Sullivan, C.S. Agate, and D.E. Beckman, *Feature-aided Tracking of Ground Targets using a Class-Independent Approach*, Proceedings of the SPIE Conference on Signal Processing, Sensor Fusion, and Target Recognition, Orlando, FL, April 2004. (cited in p. 2)



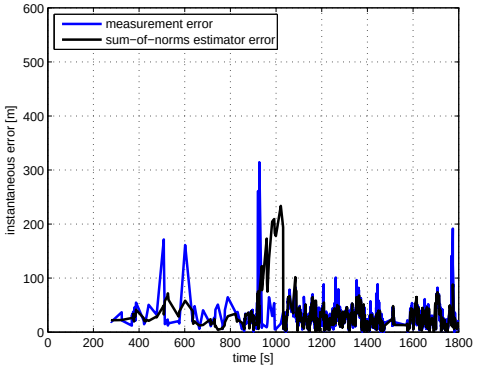
(a) Raw measurements (purple dots) obtained from the UAV "Yellow" and the corresponding estimated target positions (green dots). Several outliers seen around $(-250m, 600m)$ do not significantly affect the target position estimates.



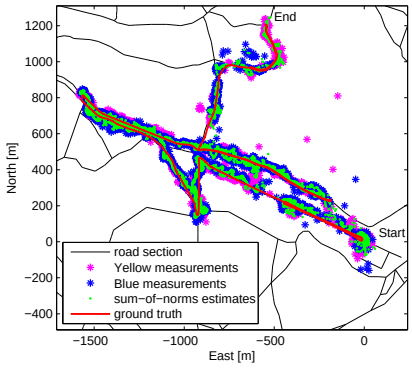
(b) Measurement and estimation errors for the data collected by the UAV "Yellow." Several outliers can be seen between 1400s and 1800s. The period around 1600s, where the estimation error grew to about 200m, corresponds to a poor target-UAV geometry that amplified the target estimation errors.



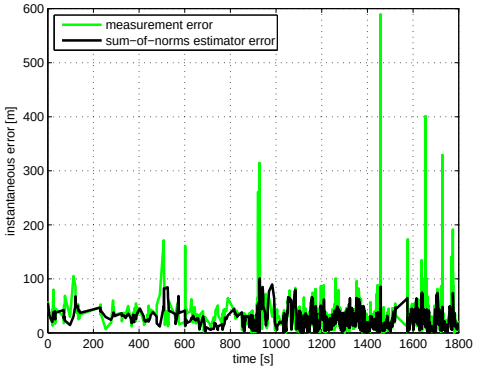
(c) Raw measurements (blue dots) obtained from the UAV "Blue," which was deployed 4 minutes after UAV "Yellow," and the corresponding estimated target positions (green dots).



(d) Measurement and estimation errors for the data collected by the UAV "Blue." Several outliers can be seen around 1000s, where the estimation error grew to about 200m.



(e) Raw measurements (purple and blue dots) obtained from both UAVs and the corresponding estimated target positions (green dots).



(f) Measurement and estimation errors for the data collected by both UAVs. By fusing the data from both UAVs one can consistently maintain a small estimation error.

Fig. 11. Experimental results collected at the Center for Interdisciplinary Remotely-Piloted Aircraft Studies (CIRPAS), Camp Roberts, California.

[5] A.P. Brown, K.J. Sullivan, and D.J. Miller, *Feature-aided Multiple Target Tracking in the Image Plane*, Proceedings of the SPIE Defense and Security Symposium, Orlando, FL, April 2006. (cited in p. 2)

[6] K. Ezal and C. Agate, *Tracking and Interception of Ground-based RF Sources Using Autonomous Guided Munitions with Passive Bearings-only Sensors and Tracking Algorithms*, Proceedings of the SPIE, Defense and Security Symposium, Orlando, FL, April 2004. (cited in p. 2)

[7] A. Kitanov, V. Tubin, and I. Petrovic, *Extending functionality of RF Ultrasound positioning system with dead-reckoning to accurately determine mobile robot's orientation*, IEEE Multiconference on Systems and Control, Saint Petersburg, Russia, July 2009. (cited in p. 2)

[8] M. Ekman, K. Davstad, and L. Sjoberg, *Ground target tracking using acoustic sensors*, Information, Decision and Control, IDC, pp.182-187, February 2007. (cited in p. 2)

[9] D. Klein, J. Isaacs, S. Venkateswaran, J. Burman, T. Pham, J. Hespanha, and U. Madhow. *Source Localization in a Sparse Acoustic Sensor Network using UAVs as Information Seeking Data Mules* IEEE Transactions on Sensor Networks, vol. 9, issue 4, November 2013. (cited in p. 2)

[10] J. Mattingley and S. Boyd, *CVXGEN: a code generator for embedded convex optimization*, Optimization and Engineering, vol. 13, issue 1, pp. 1-27, 2012. (cited in p. 2, 10)

[11] G. Bradski and A. Kaehler, *Learning OpenCV*, 1st Edition, O'Reilly Media Inc. Sebastopol, CA, USA, 2008. (cited in p. 10)

[12] A. Domahidi, A. Zgraggen, M.N. Zeilinger, M. Morari, and C.N. Jones, *Efficient Interior Point Methods for Multistage Problems Arising in Receding Horizon Control*, IEEE Conference on Decision and Control, Maui, USA, December 2012. (cited in p. 2, 10)

[13] R. Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of Royal Statistical Society B (Methodological), vol. 58, issue 1, pp. 267-288, 1996. (cited in p. 3)

[14] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2004. (cited in p. 4, 9, 12)

[15] S. Chen and D. Donoho. *Basis pursuit*, In Proceedings of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 41-44, 1994. (cited in p. 4)

[16] J. Starck, M. Elad, and D. Donoho, *Image decomposition via the combination of sparse representations and a variational approach*, IEEE Transactions on Image Processing, vol. 14 issue 10, pp. 1570-1582, 2005. (cited in p. 4)

[17] E. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Transactions on Information Theory, vol. 52, issue 2, pp. 489-509, 2006. (cited in p. 4)

[18] J. Tropp. *Just relax: Convex programming methods for identifying sparse signals in noise*, IEEE Transactions on Information Theory, vol. 53, issue 3, pp. 1030-1051, 2006. (cited in p. 4)

[19] S. Joshi and S. Boyd, *Sensor selection via convex optimization*, IEEE Transactions on Signal Processing, vol. 57, issue 2, pp. 451-462, February 2009. (cited in p. 4)

[20] A. Zymnis, S. Boyd, and D. Gorinevsky. *Relaxed maximum a posteriori fault identifcation*, Signal Processing,

vol. 89, issue 6, pp. 989-999, June 2009. (cited in p. 4)

[21] Q. Ke and T. Kanade, *Robust subspace computation using L1 norm*, Technical Report CMUCS-03-172, Carnegie Mellon University, Pittsburgh, PA, 2003. (cited in p. 4)

[22] Y. Sharon, J. Wright and Y. Ma, *Minimum Sum of Distances Estimator: Robustness and Stability*, Proceedings of the IEEE American Control Conference, pp. 524-530, St. Louis, Missouri, USA, 2009. (cited in p. 4)

[23] H. Ohlsson, F. Gustafsson, L. Ljung and S. Boyd, *Smoothed state estimates under abrupt changes using sum-of-norms regularization*, Automatica, vol. 48, issue 4, April 2012, pp. 595-605. (cited in p. 4, 9)

[24] P. Barooah and J.P. Hespanha. *Estimation from relative measurements: Algorithms and scaling laws*. IEEE Control Systems Magazine, vol. 27, issue 4, pp. 57-74, August 2007. (cited in p. 4)

[25] P. Zhang, and M. Martonosi, *LOCALE: collaborative localization estimation for sparse mobile sensor networks*. International Conference on Information Processing in Sensor Networks (IPSN), pp. 195–206, April 2008. (cited in p. 4)

[26] M. Kamgarpour and C. Tomlin, *Convergence Properties of a Decentralized Kalman Filter*, IEEE Conference on Decision and Control, pp. 3205–3210, Cancun, Mexico, December 2008. (cited in p. 4)

[27] P. Barooah, W. Russell and J. Hespanha, *Approximate distributed Kalman filtering for cooperative multi-agent localization*, Distributed Computing in Sensor Systems, Lect. Notes in Comput. Science, pp. 102-115, June 2010. (cited in p. 4)

[28] W. Russell, D. Klein and J. Hespanha, *Optimal Estimation on the Graph Cycle Space*, IEEE Transactions on Signal Processing, vol. 59, issue 6, pp. 2834-2846, June 2011. (cited in p. 4)

[29] P. Oguz-Ekim, J.P. Gomes, J. Xavier, and P. Oliveira, *Robust Localization of Nodes and Time-Recursive Tracking in Sensor Networks Using Noisy Range Measurements*, IEEE Transactions on Signal Processing, vol. 59, issue 8, pp. 3930–3942, Agust 2011. (cited in p. 4)

[30] X. Rong Li and V.P. Jilkov, *Survey of maneuvering target tracking. Part I. Dynamic models*, IEEE Transactions on Aerospace and Electronic Systems, vol. 39, no. 4, pp. 1333-1364, October 2003. (cited in p. 5)

[31] J. Löfberg, *YALMIP: a toolbox for modeling and optimization in MATLAB*, IEEE International Symposium on Computer Aided Control Systems Design, Taipei, Taiwan, September 24, 2004. (cited in p. 10)

[32] M. Grant and S. Boyd, *Graph implementations for nonsmooth convex programs*, In V. Blondel, S. Boyd, and H. Kimura, editors, Lecture Notes in Control and Information Sciences, pp. 95-110, Springer, 2008. (cited in p. 10)

[33] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming*, Version 1.22 (May 2012, Build 835), http://cvxr.com/cvx (cited in p. 10)

[34] J. Mattingley, Y. Wang and S. Boyd, *Receding Horizon Control: Automatic Generation of High-Speed Solvers*, IEEE Control Systems Magazine, vol. 31, issue 3, pp. 52-65, June 2011. (cited in p. 10)

[35] J. Laumond and J. Risler, *Nonholonomic systems: controllability and complexity*, Theoretical Computer Science, vol. 157, issue 1, pp. 101-114, April 1996. (cited in p. 11)

[36] M. Mallick, *Geolocation using Video Sensor Measurements*, Proceedings of the 10^{th} International Conference

on Information Fusion, Quebec, Canada, July 2007. (cited in p. 16)

[37] S.A.P. Quintero, F. Papi, D.J. Klein, L. Chisci and J. Hespanha. *Optimal UAV Coordination for Target Tracking using Dynamic Programming*. IEE Conference on Decision and Control, pp. 4541-4546, Atlanta, Georgia, USA, December 2010. (cited in p. 16)

[38] G.E. Collins, C.R. Stankevitz, and J. Liese, *Implementation of a Sensor Guided Flight Algorithm for Target Tracking by Small UAS*, Proceedings of the SPIE Conference on Ground/Air Multi-Sensor Interoperability, Integration, and Networking for Persistent ISR II, vol. 8047, Orlando, FL, April 2011. (cited in p. 17)