Huber–Based Divided Difference Filtering

Christopher D. Karlgaard and Hanspeter Schaub

Virginia Tech, Blacksburg, VA 24061-0203

Simulated Reprint from

Journal of Guidance, Navigation and Control

Volume 30, Number 3, May–June, 2007, Pages 885–891

A publication of the
American Institute of Aeronautics and Astronautics, Inc.
1801 Alexander Bell Drive, Suite 500
Reston, VA 22091

Huber–Based Divided Difference Filtering

Christopher D. Karlgaard* and Hanspeter Schaub†
Virginia Tech, Blacksburg, VA 24061-0203

I. Introduction

This Note describes a robust modification of the Divided Difference Filtering technique. The robust technique relies on Huber's generalized maximum likelihood approach to estimation.[1] Specifically, Huber's method is a combined minimum ℓ_1 and ℓ_2 norm estimation technique, which exhibits robustness with respect to deviations from the commonly assumed Gaussian error probability density functions, for which the least–squares or minimum ℓ_2 norm technique exhibits a severe degradation in estimation accuracy.[2] The Huber–based estimates are robust in the sense that they minimize the maximum asymptotic estimation variance when applied to contaminated Gaussian densities. The Huber technique was originally developed as a generalization of maximum likelihood estimation, applied first to estimating the center of a probability distribution in Ref. 1 and generalized to multiple linear regression in Refs. 2, 3, 4.

The Kalman filter is a recursive minimum ℓ_2 norm technique and therefore exhibits sensitivity to deviations in the true underlying error probability distributions.[5] For this reason, the Huber technique has been further extended to dynamic estimation problems. Boncelet and Dickinson[6] first proposed to solve the robust filtering problem by means of the Huber technique at each measurement, by expressing the discrete-time Kalman filter as a sequence of linear regression problems. The authors do not provide any simulation results to validate the proposed technique. Kovacevic, *et al*[7] follow the work of Ref. 6 and develop a robust Kalman filter using the Huber technique applied to a linear regression problem at each measurement update. Refs. 8, 9, 10 express the dynamic filtering problem as a sequential linear regression to be solved by the Huber technique, and apply the filter to underwater vehicle tracking, power system state estimation, and spacecraft rendezvous navigation, respectively. The increase in computation due to the use of the Huber technique was found in Ref. 10 to be small. It should be noted that Refs. 6, 7, 8, 9, 10 apply the Huber methodology to *linearized* filters.

The Divided Difference Filter is one of several new estimation techniques that are collectively known as Sigma–Point Kalman Filters (SPKF). The First–Order (DD1) and Second–Order (DD2) Divided Difference Filters[11, 12] are generalizations of the filter introduced by Schei,[13] and are two examples of SPKF–class estimators; other examples can be found in Refs. 16, 14, 15. Like the basic Kalman filter, the SPKFs seek to determine a state estimate that minimizes the ℓ_2–norm of the residuals. The SPKF technique differs from the standard Kalman filter in the sense that the SPKFs do not linearize the dynamic system for the propagation, but instead propagate a cluster of points centered around the current estimate in order to form improved approximations of the conditional mean and covariance. Specifically, the divided difference filters make use of multidimensional interpolation formulas to approximate the nonlinear transformations. As a result of this approach, the filter does not require knowledge or existence of the partial derivatives of the system dynamics and measurement equations. SPKFs have the additional advantage over the basic Kalman filter in that they can easily be extended to determine second–order solutions to the minimum ℓ_2 norm filtering problem, which increases the estimation accuracy when the system and measurement equations are nonlinear. It is important to note that the SPKFs use a minimum ℓ_2–norm measurement update and are therefore subject to the same sensitivity to non-Gaussian measurement errors as the Kalman filter. Therefore, the purpose of this Note is to modify the DD1 and DD2 measurement update equations by making use of the Huber technique to provide robustness again deviations from Gaussianity without a large increase in computation.

This Note first provides a short review of the DD1 and DD2 filters, and then shows how the measurement update can be expressed in terms of a standard regression problem, which can be solved using the robust Huber technique. The filtering techniques are then applied to a benchmark problem involving estimating the trajectory of an entry body from discrete–time range data measured by a radar tracking station. The simulation is conducted using Monte-Carlo techniques for both Gaussian and non-Gaussian cases. The computational cost associated with each filter is provided.

II. Robust Divided Difference Filtering

The filter summary closely follows that given in Refs. 11 and 12. The filter equations rely upon a discrete representation of the system dynamics, as follows

$$\mathbf{x}_{k+1} = \mathbf{F}\left(\mathbf{x}_k, \mathbf{v}_k, t_k\right) \tag{1}$$

where $\mathbf{x}_k$ is the state vector and $\mathbf{v}_k$ are random inputs to the system at time t_k. The mean value of $\mathbf{v}_k$ is $\bar{\mathbf{v}}_k = \mathbf{0}$ and its covariance is $\mathbf{Q}_k$. Observations of the state are of the form

$$\mathbf{y}_k = \mathbf{G}\left(\mathbf{x}_k, \mathbf{w}_k, t_k\right) \tag{2}$$

where $\mathbf{y}_k$ is the measurement at time t_k, and $\mathbf{w}_k$ is the measurement noise at time t_k. The mean value of $\mathbf{w}_k$ is $\bar{\mathbf{w}}_k = \mathbf{0}$ and its covariance is $\mathbf{R}_k$.

The following square-root decompositions of the predicted state error covariance, $\bar{\mathbf{P}}_k$, corrected state error covariance, $\hat{\mathbf{P}}_k$, process noise covariance, $\mathbf{Q}_k$, and measurement noise covariance, $\mathbf{R}_k$, are introduced as

$$\bar{\mathbf{P}}_k = \bar{\mathbf{S}}_{x_k}\bar{\mathbf{S}}_{x_k}^T \tag{3}$$
$$\hat{\mathbf{P}}_k = \hat{\mathbf{S}}_{x_k}\hat{\mathbf{S}}_{x_k}^T \tag{4}$$
$$\mathbf{Q}_k = \mathbf{S}_{v_k}\mathbf{S}_{v_k}^T \tag{5}$$
$$\mathbf{R}_k = \mathbf{S}_{w_k}\mathbf{S}_{w_k}^T \tag{6}$$

Also, the jth column of $\bar{\mathbf{S}}_{x_k}$ shall be referred to as $\bar{\mathbf{s}}_{x_{k_j}}$ and likewise for the other matrices.

A. Overview of the DD1 Filter

The DD1 filter makes use of first–order divided differences to approximate the system and measurement dynamics rather than the first–order Taylor series expansions used in the EKF. The following matrices of first–order divided differences are defined as

$$\mathbf{S}'_{x\hat{x}_{k_{i,j}}} = \frac{1}{2c}\Big[\mathbf{F}_i\left(\hat{\mathbf{x}} + c\hat{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{v}}_k, t_k\right) - \mathbf{F}_i\left(\hat{\mathbf{x}}_k - c\hat{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{v}}_k, t_k\right)\Big] \tag{7}$$

$$\mathbf{S}'_{xv_{k_{i,j}}} = \frac{1}{2c}\Big[\mathbf{F}_i\left(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k + c\mathbf{s}_{v_{k_j}}, t_k\right) - \mathbf{F}_i\left(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k - c\mathbf{s}_{v_{k_j}}, t_k\right)\Big] \tag{8}$$

$$\mathbf{S}'_{y\bar{x}_{k_{i,j}}} = \frac{1}{2c}\Big[\mathbf{G}_i\left(\bar{\mathbf{x}}_k + c\bar{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{w}}_k, t_k\right) - \mathbf{G}_i\left(\bar{\mathbf{x}}_k - c\bar{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{w}}_k, t_k\right)\Big] \tag{9}$$

$$\mathbf{S}'_{yw_{k_{i,j}}} = \frac{1}{2c}\Big[\mathbf{G}_i\left(\bar{\mathbf{x}}_k, \bar{\mathbf{w}}_k + c\mathbf{s}_{w_{k_j}}, t_k\right) - \mathbf{G}_i\left(\bar{\mathbf{x}}_k, \bar{\mathbf{w}}_k - c\mathbf{s}_{w_{k_j}}, t_k\right)\Big] \tag{10}$$

*Senior Project Engineer, Analytical Mechanics Associates, Inc., 303 Butler Farm Road, Suite 104A, Hampton VA, 23666. karlgaard@ama-inc.com. Senior Member AIAA.
†Assistant Professor, Department of Aerospace and Ocean Engineering, Virginia Polytechnic Institute and State University, 228 Randolph Hall, Blacksburg, VA, 24061-0203. schaub@vt.edu. Associate Fellow AIAA.
Presented as Paper 06–3792 at the AIAA Guidance, Navigation and Control Conference, San Diego, CA, July 29–31, 1996.

where c the divided–difference perturbing parameter.

The state prediction, $\bar{\mathbf{x}}_{k+1}$ state root–covariance prediction, $\bar{\mathbf{S}}_{x_{k+1}}$, measurement prediction, $\bar{\mathbf{y}}_k$, and measurement root-covariance prediction, $\mathbf{S}_{y_k}$, are given by

$$\bar{\mathbf{x}}_{k+1} = \mathbf{F}\left(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k, t_k\right) \quad (11)$$
$$\bar{\mathbf{S}}_{x_{k+1}} = \mathcal{H}\left(\begin{bmatrix} \mathbf{S}'_{x\hat{x}_k} & \mathbf{S}'_{xv_k} \end{bmatrix}\right) \quad (12)$$
$$\bar{\mathbf{y}}_k = \mathbf{G}\left(\bar{\mathbf{x}}_k, \bar{\mathbf{w}}_k, t_k\right) \quad (13)$$
$$\mathbf{S}_{y_k} = \mathcal{H}\left(\begin{bmatrix} \mathbf{S}'_{y\bar{x}_k} & \mathbf{S}'_{yw_k} \end{bmatrix}\right) \quad (14)$$

where $\mathcal{H}(\cdot)$ represents a Householder transformation of the argument matrix.[11, 12]

The state, $\hat{\mathbf{x}}_k$, and state root-covariance, $\hat{\mathbf{S}}_{x_k}$, measurement update equations are given by

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k\left(\mathbf{y}_k - \bar{\mathbf{y}}_k\right) \quad (15)$$
$$\hat{\mathbf{S}}_{x_k} = \mathcal{H}\left(\begin{bmatrix} \bar{\mathbf{S}}_{x_k} - \mathbf{K}_k\mathbf{S}'_{yx_k} & \mathbf{K}_k\mathbf{S}'_{yw_k} \end{bmatrix}\right) \quad (16)$$

where $\mathbf{K}_k = \bar{\mathbf{S}}_{x_k}\mathbf{S}'^T_{y\bar{x}_k}\left(\mathbf{S}_{y_k}\mathbf{S}^T_{y_k}\right)^{-1}$ is the Kalman gain matrix.

B. Overview of the DD2 Filter

The DD2 filter makes use of second–order divided differences to approximate nonlinear transformation of the state and covariance. The matrices of second–order divided differences are defined as

$$\begin{aligned}\mathbf{S}''_{x\hat{x}_{k_{i,j}}} = & \frac{\sqrt{c^2-1}}{2c^2}\Big[\mathbf{F}_i\left(\hat{\mathbf{x}}_k + c\hat{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{v}}_k, t_k\right) \\ & + \mathbf{F}_i\left(\hat{\mathbf{x}}_k - c\hat{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{v}}_k, t_k\right) \\ & - 2\mathbf{F}_i\left(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k, t_k\right)\Big]\end{aligned} \quad (17)$$

$$\begin{aligned}\mathbf{S}''_{xv_{k_{i,j}}} = & \frac{\sqrt{c^2-1}}{2c^2}\Big[\mathbf{F}_i\left(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k + c\mathbf{s}_{v_{k_j}}, t_k\right) \\ & + \mathbf{F}_i\left(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k - c\mathbf{s}_{v_{k_j}}, t_k\right) \\ & - 2\mathbf{F}_i\left(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k, t_k\right)\Big]\end{aligned} \quad (18)$$

$$\begin{aligned}\mathbf{S}''_{y\bar{x}_{k_{i,j}}} = & \frac{\sqrt{c^2-1}}{2c^2}\Big[\mathbf{G}_i\left(\bar{\mathbf{x}} + c\bar{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{w}}_k, t_k\right) \\ & + \mathbf{G}_i\left(\bar{\mathbf{x}} - c\bar{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{w}}_k, t_k\right) \\ & - 2\mathbf{G}_i\left(\bar{\mathbf{x}}_k, \bar{\mathbf{w}}_k, t_k\right)\Big]\end{aligned} \quad (19)$$

$$\begin{aligned}\mathbf{S}''_{yw_{k_{i,j}}} = & \frac{\sqrt{c^2-1}}{2c^2}\Big[\mathbf{G}_i\left(\bar{\mathbf{x}}_k, \bar{\mathbf{w}}_k + c\mathbf{s}_{w_{k_j}}, t_k\right) \\ & + \mathbf{G}_i\left(\bar{\mathbf{x}}_k\bar{\mathbf{w}}_k - c\mathbf{s}_{w_{k_j}}, t_k\right) \\ & - 2\mathbf{G}_i\left(\bar{\mathbf{x}}_k, \bar{\mathbf{w}}_k, t_k\right)\Big]\end{aligned} \quad (20)$$

The state, state root–covariance, measurement, and measurement covariance predictions are given by

$$\begin{aligned}\bar{\mathbf{x}}_{k+1} = & \left(\frac{c^2 - n_x - n_v}{c^2}\right)\mathbf{F}\left(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k, t_k\right) \\ & + \frac{1}{2c^2}\sum_{j=1}^{n_x}\Big[\mathbf{F}\left(\hat{\mathbf{x}}_k + c\hat{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{v}}_k, t_k\right) \\ & \qquad + \mathbf{F}\left(\hat{\mathbf{x}}_k - c\hat{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{v}}_k, t_k\right)\Big] \\ & + \frac{1}{2c^2}\sum_{j=1}^{n_v}\Big[\mathbf{F}\left(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k + c\mathbf{s}_{v_{k_j}}, t_k\right) \\ & \qquad + \mathbf{F}\left(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k - c\mathbf{s}_{v_{k_j}}, t_k\right)\Big]\end{aligned} \quad (21)$$

$$\bar{\mathbf{S}}_{x_{k+1}} = \mathcal{H}\left(\begin{bmatrix} \mathbf{S}'_{x\hat{x}_k} & \mathbf{S}'_{xv_k} & \mathbf{S}''_{x\hat{x}_k} & \mathbf{S}''_{xv_k} \end{bmatrix}\right) \quad (22)$$

$$\begin{aligned}\bar{\mathbf{y}}_k = & \left(\frac{c^2 - n_x - n_w}{c^2}\right)\mathbf{G}\left(\bar{\mathbf{x}}_k, \bar{\mathbf{w}}_k, t_k\right) \\ & + \frac{1}{2c^2}\sum_{j=1}^{n_x}\Big[\mathbf{G}\left(\bar{\mathbf{x}}_k + c\bar{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{w}}_k, t_k\right) \\ & \qquad + \mathbf{G}\left(\bar{\mathbf{x}}_k - c\bar{\mathbf{s}}_{x_{k_j}}, \bar{\mathbf{w}}_k, t_k\right)\Big] \\ & + \frac{1}{2c^2}\sum_{j=1}^{n_w}\Big[\mathbf{G}\left(\bar{\mathbf{x}}_k, \bar{\mathbf{w}}_k + c\mathbf{s}_{w_{k_j}}, t_k\right) \\ & \qquad + \mathbf{G}\left(\bar{\mathbf{x}}_k, \bar{\mathbf{w}}_k - c\mathbf{s}_{w_{k_j}}, t_k\right)\Big]\end{aligned} \quad (23)$$

$$\mathbf{S}_{y_k} = \mathcal{H}\left(\begin{bmatrix} \mathbf{S}'_{y\bar{x}_k} & \mathbf{S}'_{yw_k} & \mathbf{S}''_{y\bar{x}_k} & \mathbf{S}''_{yw_k} \end{bmatrix}\right) \quad (24)$$

where n_x is the size of the state dimension, n_v is the size of the process noise dimension, and n_w is the size of the measurement noise dimension.

Lastly, the state and root-covariance update equations are given by

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k\left(\mathbf{y}_k - \bar{\mathbf{y}}_k\right) \quad (25)$$

$$\hat{\mathbf{S}}_{x_k} = \mathcal{H}\left(\begin{bmatrix} \bar{\mathbf{S}}_{x_k} - \mathbf{K}_k\mathbf{S}'_{yx_k} & \mathbf{K}_k\mathbf{S}'_{yw_k} & \cdots \\ \cdots & \mathbf{K}_k\mathbf{S}'_{yx_k} & \mathbf{K}_k\mathbf{S}'_{yw_k} \end{bmatrix}\right) \quad (26)$$

where $\mathbf{K}_k = \bar{\mathbf{S}}_{x_k}\mathbf{S}'^T_{y\bar{x}_k}\left(\mathbf{S}_{y_k}\mathbf{S}^T_{y_k}\right)^{-1}$ is the Kalman gain matrix.

Note that many of the same state and noise perturbations used to calculate the first–order divided differences are again used to compute the second–order divided differences. This point has important implications with regard to the computational costs, suggesting that the DD2 filter may not require a great deal more computing time than the DD1 filter.

C. Modification of Measurement Update Using Huber's Technique

This section discusses how the measurement update equations of the DD1 and DD2 filters can be solved using the Huber method. To apply this method, it is first required to recast the measurement update as a regression problem between the observed quantity and the state prediction. If the true value of the state is written as $\mathbf{x}_k$ and the state prediction error is written as $\boldsymbol{\delta}_k = \mathbf{x}_k - \bar{\mathbf{x}}_k$, then the state prediction can be expressed as $\bar{\mathbf{x}}_k = \mathbf{x}_k - \boldsymbol{\delta}_k$. By defining the cross-covariance matrix,[11] $\mathbf{P}_{xy_k} = \bar{\mathbf{S}}_{x_k}\left(\mathbf{S}'_{yx_k}\right)^T$, and the matrix $\mathbf{H}_k = \mathbf{P}^T_{xy_k}\bar{\mathbf{P}}^{-1}_k = \mathbf{S}'_{yx_k}\left(\bar{\mathbf{S}}^T_{x_k}\right)^{-1}$, then the measurement equation can be approximated by $\mathbf{y}_k \approx \bar{\mathbf{y}}_k + \mathbf{H}_k\left(\mathbf{x}_k - \bar{\mathbf{x}}_k\right)$. The measurement update can then be written as the solution to the linear regression problem[10, 17]

$$\mathbf{z}_k = \mathbf{M}_k\mathbf{x}_k + \boldsymbol{\xi}_k \quad (27)$$

where

$$\mathbf{z}_k = \Lambda_k^{-1}\left\{\begin{array}{c}\mathbf{y}_k - \bar{\mathbf{y}}_k + \mathbf{H}_k\bar{\mathbf{x}}_k \\ \bar{\mathbf{x}}_k\end{array}\right\} \quad (28)$$

$$\mathbf{M}_k = \Lambda_k^{-1}\left[\begin{array}{c}\mathbf{H}_k \\ \mathbf{I}\end{array}\right] \quad (29)$$

$$\boldsymbol{\xi}_k = \Lambda_k^{-1}\left\{\begin{array}{c}\mathbf{w}_k \\ -\boldsymbol{\delta}_k\end{array}\right\} \quad (30)$$

$$\Lambda_k = \left[\begin{array}{cc}\mathbf{S}_{w_k} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{S}}_{x_k}\end{array}\right] \quad (31)$$

This regression problem can be solved using Huber's generalized maximum likelihood technique, in which the solution is found by minimizing the cost function

$$J(\mathbf{x}) = \sum_{i=1}^{m} \rho(\zeta_i) \quad (32)$$

Here, ζ_i is the ith component of the residual vector, $\boldsymbol{\zeta} = \mathbf{M}_k\hat{\mathbf{x}}_k - \mathbf{z}_k$, m is the dimension of $\boldsymbol{\zeta}$, and ρ is a positive symmetric function with $\rho(0) = 0$, but otherwise arbitrary. Note that this formulation reduces to the standard maximum likelihood technique when $\rho(\zeta_i) = -\ln[f(\zeta_i)]$, where $f(\zeta_i)$ is the assumed probability density function. For instance, Huber[1] introduces a ρ function of the form

$$\rho(\zeta_i) = \begin{cases} \frac{1}{2}\zeta_i^2 & \text{for } |\zeta_i| < \gamma \\ \gamma|\zeta_i| - \frac{1}{2}\gamma^2 & \text{for } |\zeta_i| \geq \gamma \end{cases} \quad (33)$$

where γ is a tuning parameter. This ρ function is a blend of the ℓ_1 and ℓ_2 norm functions, and estimates derived from the use of this ρ function have desirable robustness properties. Specifically, the estimates minimize the maximum asymptotic estimation variance when applied to contaminated Gaussian densities. See Ref. 17 for more information on the Huber technique, including guidelines on how to choose the value of the tuning parameter γ.

If the ρ function is differentiable, which is the case for Eq. (33), then the solution to the generalized maximum likelihood regression problem can be found from the implicit equation

$$\sum_{i=1}^{m} \phi(\zeta_i)\frac{\partial \zeta_i}{\partial \mathbf{x}} = \mathbf{0} \quad (34)$$

where $\phi(\zeta_i) = \rho'(\zeta_i)$. By defining the function $\psi(\zeta_i) = \phi(\zeta_i)/\zeta_i$, and the matrix $\boldsymbol{\Psi} = \text{diag}[\psi(\zeta_i)]$, the implicit equation can be written in matrix form as

$$\mathbf{M}_k^T\boldsymbol{\Psi}(\mathbf{M}_k\mathbf{x}_k - \mathbf{z}_k) = \mathbf{0} \quad (35)$$

Equation (35) can be expanded to yield $\mathbf{M}^T\boldsymbol{\Psi}\mathbf{M}\mathbf{x}_k = \mathbf{M}^T\boldsymbol{\Psi}\mathbf{z}_k$, which can be solved for $\mathbf{x}_k$ to give $\mathbf{x}_k = \left(\mathbf{M}^T\boldsymbol{\Psi}\mathbf{H}\right)^{-1}\mathbf{M}_k^T\boldsymbol{\Psi}\mathbf{z}_k$. Since the matrix $\boldsymbol{\Psi}$ depends on the residuals ζ_i, and hence on $\mathbf{x}_k$, an iterative solution to Eq. (35) is expressed as

$$\mathbf{x}_k^{(j+1)} = \left(\mathbf{M}_k^T\boldsymbol{\Psi}^{(j)}\mathbf{M}_k\right)^{-1}\mathbf{M}_k^T\boldsymbol{\Psi}^{(j)}\mathbf{z}_k \quad (36)$$

where the superscript (j) refers to the iteration index. The method can be initialized by using the least–squares solution $\mathbf{x}_k^{(0)} = \left(\mathbf{M}_k^T\mathbf{M}_k\right)^{-1}\mathbf{M}_k^T\mathbf{z}_k$. The converged value from the iterative procedure is taken as the state estimate, $\hat{\mathbf{x}}_k$. This technique is known as iteratively reweighted least squares,[18] and is generally attributed to Beaton and Tukey.[19] This iteration will converge if the ψ function is non-increasing[20] (for $\zeta_i > 0$), which is the case when using the ρ function in Eq. (33). The algorithm can be iterated until convergence or can be carried out through only one fixed iteration step, as discussed by Bickel[21] and Rousseeuw and Leroy,[22] an approach that captures the robustness properties and also saves on the computational costs associated with the iterative solution.

Due to the particular structure of the matrix $\mathbf{M}_k$, the discrete time dynamic state estimation technique can be simplified considerably from the static state estimation technique by application of the matrix inversion lemma.[23] By first decomposing the $\boldsymbol{\Psi}$ matrix into two portions $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ corresponding to the state prediction and measurement prediction residuals so that

$$\boldsymbol{\Psi} = \left[\begin{array}{cc}\boldsymbol{\Psi}_y & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_x\end{array}\right] \quad (37)$$

then the measurement update can be expressed in the same recursive form as the DD1 and DD2 filter updates. The results for each filter are as follows.

If the measurement update given in Eq. (15) is taken as the initial guess for the state then a one-step Huber update for the DD1 filter can be written as

$$\mathbf{S}_{y_k}^{(1)} = \mathcal{H}\left(\left[\begin{array}{cc}\mathbf{S}'_{y\bar{x}_k}\boldsymbol{\Psi}_x^{-1/2} & \mathbf{S}'_{yw_k}\boldsymbol{\Psi}_y^{-1/2}\end{array}\right]\right) \quad (38)$$

$$\mathbf{K}_k^{(1)} = \bar{\mathbf{S}}_{x_k}\boldsymbol{\Psi}_x^{-1}\mathbf{S}'^{T}_{y\bar{x}_k}\left(\mathbf{S}_{y_k}^{(1)}\mathbf{S}_{y_k}^{(1)T}\right)^{-1} \quad (39)$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k^{(1)}(\mathbf{y}_k - \bar{\mathbf{y}}_k) \quad (40)$$

$$\hat{\mathbf{S}}_{x_k} = \mathcal{H}\Big(\Big[\begin{array}{ccc}\bar{\mathbf{S}}_{x_k}\boldsymbol{\Psi}_x^{-1/2} - \mathbf{K}_k^{(1)}\mathbf{S}'_{yx_k}\boldsymbol{\Psi}_x^{-1/2} & \cdots \\ \cdots & \mathbf{K}_k^{(1)}\mathbf{S}'_{yw_k}\boldsymbol{\Psi}_y^{-1/2}\end{array}\Big]\Big) \quad (41)$$

Similarly for the DD2 filter, if the measurement update given in Eq. (25) is taken as the initial guess for the state, then a one-step Huber update can be written as

$$\mathbf{S}_{y_k}^{(1)} = \mathcal{H}\Big(\Big[\begin{array}{ccc}\mathbf{S}'_{y\bar{x}_k}\boldsymbol{\Psi}_x^{-1/2} & \mathbf{S}'_{yw_k}\boldsymbol{\Psi}_y^{-1/2} & \cdots \\ \cdots & \mathbf{S}''_{y\bar{x}_k}\boldsymbol{\Psi}_x^{-1/2} & \mathbf{S}''_{yw_k}\boldsymbol{\Psi}_y^{-1/2}\end{array}\Big]\Big) \quad (42)$$

$$\mathbf{K}_k^{(1)} = \bar{\mathbf{S}}_{x_k}\boldsymbol{\Psi}_x^{-1}\mathbf{S}'^{T}_{y\bar{x}_k}\left(\mathbf{S}_{y_k}^{(1)}\mathbf{S}_{y_k}^{(1)T}\right)^{-1} \quad (43)$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k^{(1)}(\mathbf{y}_k - \bar{\mathbf{y}}_k) \quad (44)$$

$$\hat{\mathbf{S}}_{x_k} = \mathcal{H}\Big(\Big[\begin{array}{ccc}\bar{\mathbf{S}}_{x_k}\boldsymbol{\Psi}_x^{-1/2} - \mathbf{K}_k^{(1)}\mathbf{S}'_{yx_k}\boldsymbol{\Psi}_x^{-1/2} & \cdots \\ \cdots & \mathbf{K}_k^{(1)}\mathbf{S}'_{yw_k}\boldsymbol{\Psi}_y^{-1/2} & \mathbf{K}_k^{(1)}\mathbf{S}''_{yx_k}\boldsymbol{\Psi}_x^{-1/2} & \cdots \\ \cdots & \mathbf{K}_k^{(1)}\mathbf{S}''_{yw_k}\boldsymbol{\Psi}_y^{-1/2}\end{array}\Big]\Big) \quad (45)$$

In each case, $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ are diagonal matrices computed from the Huber ψ function, with residuals that take the form

$$\boldsymbol{\zeta} = \Lambda_k^{-1}\left\{\begin{array}{c}\mathbf{y}_k - \bar{\mathbf{y}}_k \\ \hat{\mathbf{x}}_k^{(0)} - \bar{\mathbf{x}}_k\end{array}\right\} \quad (46)$$

where the superscript (0) refers to the initial state estimate computed from the standard DD1 or DD2 update.

Note that as $\gamma \to \infty$, the modified DD1 and DD2 measurement updates using the Huber technique reduce to the original form presented in Sec. II.A. and Sec. II.B.. Specifically, when $\gamma \to \infty$, the matrix $\boldsymbol{\Psi} \to \mathbf{I}$, and Eq. (35) can be solved exactly in one iteration step and is equal to the standard DD1 and DD2 measurement updates.

III. Application to a Benchmark Nonlinear Filtering Problem

This section discusses the application of the robust filters to the problem of estimating the trajectory of a target using range measurements recorded from a radar tracking station. The example problem is to estimate the trajectory of a mass falling through an exponential atmosphere with a constant, yet unknown, drag coefficient. The gravitational acceleration acting on the body is neglected in the dynamic model. This truncated model is valid for high initial

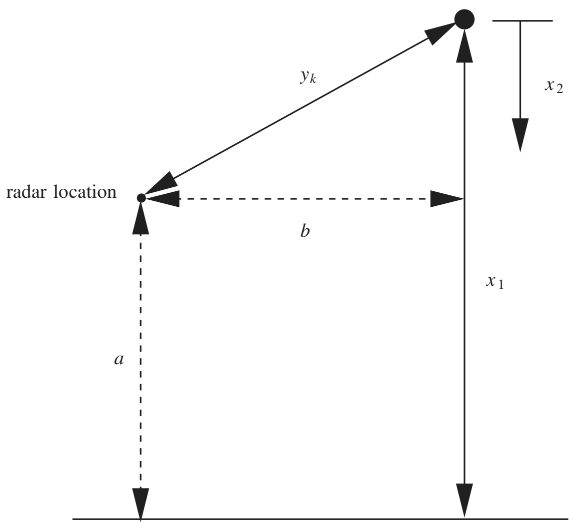


Figure 1: Geometry of Example Problem

velocities that cause the aerodynamic acceleration to dominate over the gravitational acceleration. This benchmark nonlinear filtering problem was initially studied in Ref. 24 and has been repeated numerous times in the literature. Fig. 1 shows the geometry of the problem.

A. Dynamic Model and Measurement Equations

The dynamic model for this problem is

$$\dot{x}_1 = -x_2 \tag{47}$$
$$\dot{x}_2 = -x_3^2 x_2^2 \mathrm{e}^{-\eta x_1} \tag{48}$$
$$\dot{x}_3 = 0 \tag{49}$$

where x_1 represents the altitude of the mass, x_2 its downward velocity, x_3 is a constant ballistic parameter, and η is the known constant inverse atmospheric density scale height.

The radar range measurement equation is

$$y_k = \sqrt{b^2 + \left[x_1\left(t_k\right) - a\right]^2} + w_k \tag{50}$$

where w_k represents zero-mean random error, with probability density function $f(w_k)$. Random measurement errors are drawn from the mixture of zero–mean Gaussian probability distributions, defined by the probability density function

$$f\left(w_k\right) = \left(\frac{1-\epsilon}{\sigma_1\sqrt{2\pi}}\right)\exp\left[-\left(\frac{w_k^2}{2\sigma_1^2}\right)\right] + \left(\frac{\epsilon}{\sigma_2\sqrt{2\pi}}\right)\exp\left[-\left(\frac{w_k^2}{2\sigma_2^2}\right)\right] \tag{51}$$

where σ_1 and σ_2 are the standard deviations of the individual Gaussian distributions, and ϵ is a perturbing parameter that represents error model contamination. The standard deviations σ_1 are chosen according to Table 1 and σ_2 is chosen as $\sigma_2 = 5\sigma_1$. The measurements are assumed to occur at a frequency of 1 Hz. The model parameters and initial conditions for the problem are summarized in Table 1 and Table 2.

B. Results of Example Problem

This section discusses the results of applying several filters to the benchmark tracking problem. These filters include the EKF, DD1, DD2 and the robust versions of the EKF (discussed in Ref. 10) and DD1 and DD2 (discussed in Sec. II.C.), each using the one–step Huber update. The results of a Monte–Carlo simulation are shown in the following figures. In this simulation, 2000 trial cases have been conducted, each case terminating after an elapsed time of 60 s.

Table 1: Simulation Parameters

Parameter	Value
a, km	30.5
b, km	30.5
η, m^{-1}	$1.64 \cdot 10^{-4}$
σ_1, m	30.5
γ	1.345
c^2	3.0

Table 2: Initial Conditions

Initial State	True Value	Estimated Value	Standard Deviation
$x_1(0)$, km	91.5	91.5	0.31
$x_2(0)$, km/s	6.1	6.1	0.06
$x_3(0)$, $1/\sqrt{\mathrm{m}}$	0.06	0.01	0.02

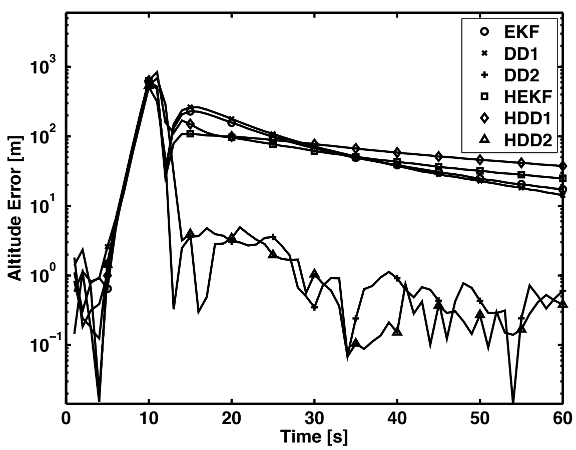
Table 3: EKF-Relative Computation Ratios

Filter	Computation Ratio
H-EKF	1.08
DD1	2.95
H-DD1	3.14
DD2	3.02
H-DD2	3.19

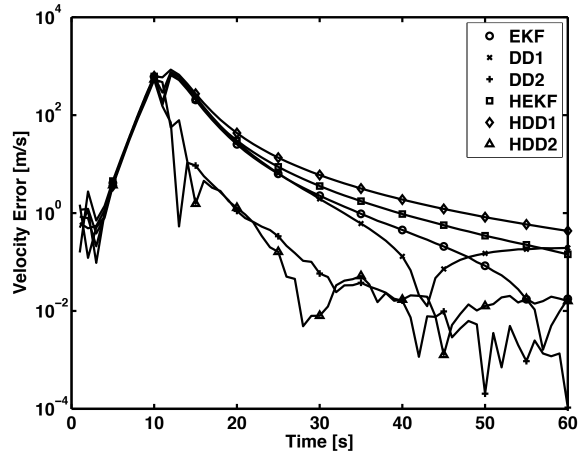
The absolute value of the median errors are shown for the case of $\epsilon = 0$ in Figs. 2(a)–(c). In this case the DD2 filter gives a smaller estimation error than the other robust and non-robust filters. The DD1 filter exhibits slightly larger errors than the EKF. These results are not surprising based on the results given in Ref. 11 and 12. The robust filters do not perform as well as their non-robust counterparts in this case, because the Huber update does not minimize the ℓ_2 norm during the measurement update. The increase in the estimation error for the robust filters is to be expected in a perfectly Gaussian simulation since the minimum ℓ_2 norm is the maximum likelihood estimator in this case.

The absolute value of the median errors are shown in Figs. 2(d)–(f) for the case $\epsilon = 0.5$, for the case where the measurement errors are highly non–Gaussian. In this case, the Huber–EKF and DD2 filters give comparable results to each other for the position and velocity errors, but the Huber–EKF gives a smaller error in the estimate of the ballistic parameter. Both Huber–EKF and DD2 are superior to the EKF and DD1 filter in this case. The Huber–DD2 filter exhibits the smallest errors since it captures both nonlinearity and non-Gaussianity. The DD1 and Huber–DD1 filters do not perform as well as the EKF and the Huber–EKF, respectively, in the non–Gaussian case, which follows the behavior from the Gaussian case. In the non-Gaussian case, six cases of the EKF and two cases of the DD1 methods diverged completely, while the Huber-EKF and Huber-DD1 did not exhibit any divergence. The DD2 filter, being a second–order filter and therefore not as sensitive to initialization, did not diverge in any of the Monte-Carlo cases, but clearly the median error is reduced by making use of the Huber update method.

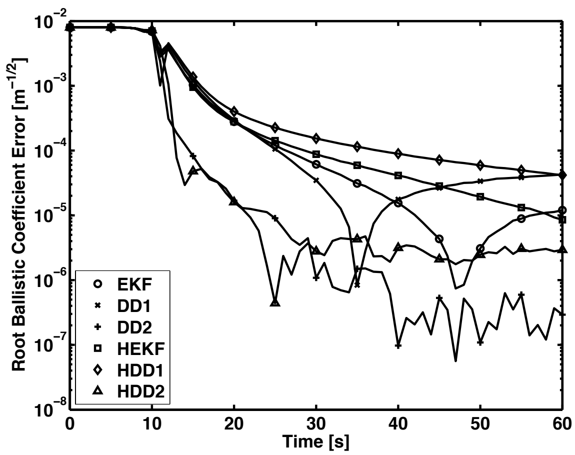
The computational cost associated with implementing the each filter is summarized in Table 3. In this table, the computational costs are divided by the EKF processing time to provide a ratio of the cost relative to that associated with the EKF. These ratios are based on the median computation time for each filter during the Monte-Carlo simulation. The results show that the Huber–EKF filter requires the smallest relative computational cost whereas the DD2 and Huber–DD2 filters require more, at 3.12 and 3.15 times



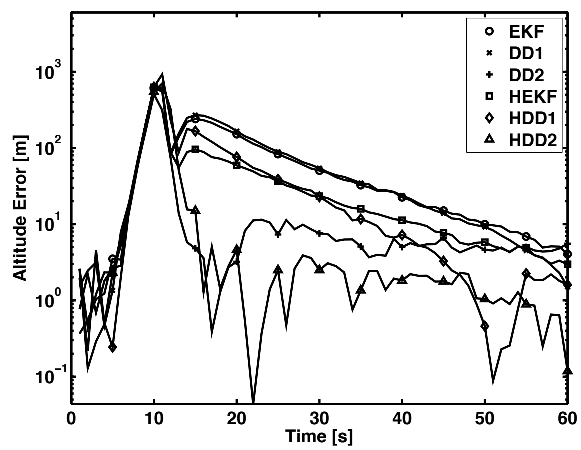
a) Estimated Position Errors for $\epsilon = 0.0$



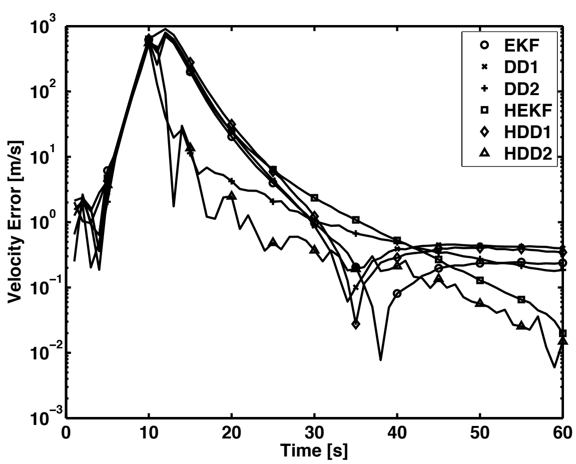
b) Estimated Velocity Errors for $\epsilon = 0.0$



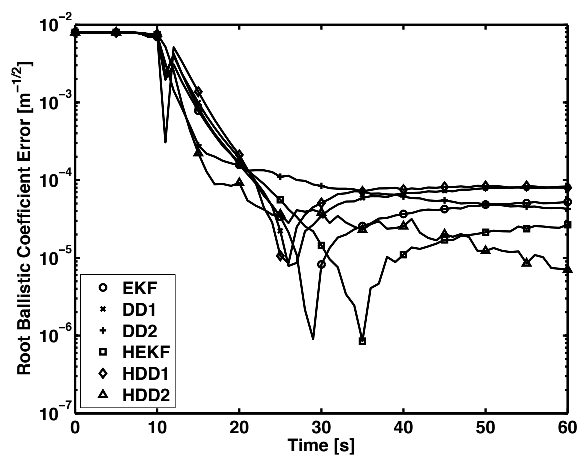
c) Estimated Ballistic Coefficient Errors for $\epsilon = 0.0$



d) Estimated Position Errors for $\epsilon = 0.5$



e) Estimated Velocity Errors for $\epsilon = 0.5$



f) Estimated Ballistic Coefficient Errors for $\epsilon = 0.5$

Figure 2: Estimated Trajectory Errors

the total computation, respectively. It is not surprising that the Huber–DD2 filter has the largest cost since it has the smallest errors for the non-Gaussian case, but it is interesting to note that the similar levels of accuracy in the non-Gaussian case can be found by use of the Huber–EKF over that of DD2 filter, for only a fraction of the computation time.

IV. Conclusions

This paper has discussed robust divided difference filtering techniques based on Huber's generalized maximum likelihood estimation theory, which provides robustness against deviations from the Gaussian error distribution. The standard divided difference filtering problem was recast in the form of a sequence of linear regression problems, to be solved at each measurement point using the robust Huber technique. The robust filters were applied to a benchmark tracking problem involving the estimation of the trajectory of an entry body from discrete–time, noisy range measurement data provided by a radar tracking system. The simulation includes the standard EKF and an EKF with Huber update, in addition to the divided difference filtering techniques discussed in this paper. The simulation was conducted using Monte–Carlo techniques with both Gaussian and non–Gaussian error distributions in order to asses the performance of the filtering techniques.

The results show that for perfectly Gaussian error distributions the standard DD2 filter exhibits the lowest estimation error time history, which was the expected outcome based on previously published results. The DD1 filter exhibited slightly larger estimation errors than that of the EKF. The filters with the Huber update technique produced larger errors than the standard update, since the standard form of the update is a the maximum likelihood estimate for the perfectly Gaussian case. However, for non–Gaussian error distributions, the modified filters with the Huber update outperformed the standard filters. The modified DD2 filter with the Huber update equation exhibited the smallest errors in the non-Gaussian numerical simulations conducted. The Huber-EKF and the Huber-DD1 filters were able to mitigate divergence problems in their non-robust counterparts.

Comparisons of the computational costs associated with each filter show that the Huber–EKF filter is able to process data at a rate approximately three times faster than the standard DD2 filter, and produces similar accuracy levels in the non–Gaussian case. Therefore, for non-Gaussian cases, the Huber–EKF filter is superior to the standard DD2 filter. If computation costs are not a concern for the particular application, then the Huber–DD2 filter exhibits the best performance.

References

[1] Huber, P. J., "Robust Estimation of a Location Parameter," *Annals of Mathematical Statistics*, Vol. 35, No. 2, 1964, pp. 73–101.

[2] Huber, P. J., "Robust Statistics: A Review," *Annals of Mathematical Statistics*, Vol. 43, No. 4, 1972, pp. 1041–1067.

[3] Huber, P. J., "Robust Regression: Asymptotics, Conjectures and Monte Carlo," *Annals of Statistics*, Vol. 1, No. 5, 1973, pp. 799–821.

[4] Huber, P. J., *Robust Statistics*, Wiley, New York, 1981, pp. 43–106, 153–198.

[5] Schick, I. C. and Mitter, S. K., "Robust Recursive Estimation in the Presence of Heavy-Tailed Observation Noise," *Annals of Statistics*, Vol. 22, No. 2, 1994, pp. 1045–1080.

[6] Boncelet, C. G. and Dickinson, B. W., "An Approach to Robust Kalman Filtering," *22nd IEEE Conference on Decision and Control*, Institute of Electrical and Electronics Engineers, New York, NY, 1983, pp. 304–305.

[7] Kovacevic, B., Durovic, Z., and Glavaski, S., "On Robust Kalman Filtering," *International Journal of Control*, Vol. 56, No. 3, 1992, pp. 547–562.

[8] El-Hawary, F. and Jing, Y., "Robust Regression-Based EKF for Tracking Underwater Targets," *IEEE Journal of Oceanic Engineering*, Vol. 20, No. 1, 1995, pp. 31–41.

[9] Durgaprasad, G. and Thakur, S. S., "Robust Dynamic State Estimation of Power Systems Based on M-Estimation and Realistic Modeling of System Dynamics," *IEEE Transactions on Power Systems*, Vol. 13, No. 4, 1998, pp. 1331–1336.

[10] Karlgaard, C. D., "Robust Rendezvous Navigation in Elliptical Orbit," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 2, 2006, pp. 495–499.

[11] Nørgaard, M., Poulsen, N. K. and Ravn, O., "Advances in Derivative-Free State Estimation for Nonlinear Systems," Technical Report IMM-REP-1998-15, Department of Mathematical Modelling, Technical University of Denmark, revised April 2000.

[12] Nørgaard, M., Poulsen, N. K. and Ravn, O., "New Developments in State Estimation for Nonlinear Systems," *Automatica,* Vol. 36, No. 11, 2000, pp. 1627–1638.

[13] Schei, T. S., "A Finite–Difference Method for Linearization in Nonlinear Estimation Algorithms," *Automatica*, Vol. 33, No. 11, 1997, pp. 2053–2058.

[14] Julier, S., Uhlmann, J., and Durrant–Whyte, H. F., "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, 2000, pp. 477–482.

[15] Julier, S. J. and Uhlmann, J. K., "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, 2004, pp. 401–422.

[16] Ito, K. and Xiong, K., "Gaussian Filters for Nonlinear Filtering Problems," *IEEE Transactions on Automatic Control,* Vol. 45, No. 5, 2000, pp. 910–927.

[17] Karlgaard, C. D. and Schaub, H., "Comparison of Several Nonlinear Filters for a Benchmark Tracking Problem," American Institute of Aeronautics and Astronautics, AIAA Paper 2006-6243, August 2006.

[18] Holland, P. W. and Welsch, R. E., "Robust Regression Using Iteratively Reweighted Least Squares," *Communications in Statistics: Theory and Methods*, Vol. 6, No. 9, 1977, pp. 813–827.

[19] Beaton, A. E. and Tukey, J. W., "The Fitting of Power Series, Meaning Polynomials, Illustrated on Band–Spectroscopic Data," *Technometrics*, Vol. 16, No. 2, 1974, pp. 147–185.

[20] Maronna, R. A., Martin, R. D., and Yohai, V. J., *Robust Statistics: Theory and Methods*, Wiley, Chichester, 2006, pp. 104–105.

[21] Bickel, P. J., "One-Step Huber Estimates in the Linear Model," *Journal of the American Statistical Association*, Vol. 70, No. 350, 1975, pp. 428-434.

[22] Rousseeuw, P. J. and Leroy, A. M., *Robust Regression and Outlier Detection*, Wiley, New York, 1987, pp. 158–174.

[23] Burl, J. B., *Linear Optimal Control*, Addison–Wesley, Reading, MA, 1999, pp. 437.

[24] Athans, M., Wishner, R. P. and Bertolini, A., "Suboptimal State Estimation for Continuous-Time Nonlinear Systems from Discrete Noisy Measurements," *IEEE Transactions on Automatic Control,* Vol. 13, No. 5, 1968, pp. 504–514.