Fusion of Tracks with Road Constraints

CHUN YANG
ERIK BLASCH

This paper is concerned with tracking of ground targets on roads and investigates possible ways to improve target state estimation via fusing a target's track with information about the road along which the target is traveling. A target track is estimated using a surveillance radar whereas a digital map provides the road network of the region under surveillance. When the information about roads is as accurate as (or even better than) radar measurements, it is desired naturally to incorporate such information (fusion) into target state estimation. In this paper, roads are modeled with analytic functions and their fusion with a target track is cast as linear or nonlinear state constraints in an optimization procedure. The constrained optimization is then solved with the Lagrangian multiplier, leading to a closed-form solution for linear constraints and an iterative solution for second-order nonlinear constraints. Geometric interpretations of the solutions are provided for special cases. Compared to other methods, the track-to-road fusion using the constrained optimization technique can be easily implemented as an add-on module without changes to an existing tracker. For curved roads with coarse waypoints, the nonlinear constrained solution outperforms the piecewise linearized constrained approach. Computer simulation results are presented to illustrate the algorithms.

Manuscript received October 4, 2006; revised August 3, 2007 and June 12, 2008; released for publication on June 25, 2008.

Refereeing of this contribution was handled by Benjamin J. Slocumb.

Authors' addresses: C. Yang, Sigtem Technology, Inc., 1343 Parrott Drive, San Mateo, CA 94402, E-mail: (chunyang@sigtem.com); E. Blasch, Air Force Research Lab/RYAA, 2241 Avionics Circle, WPAFB, OH 45433, E-mail: (erik.blasch@wpafb.af.mil).

1557-6418/08/$17.00

1. INTRODUCTION

With the rapid building up of geographic information system (GIS) including digital road maps (DRM) and digital terrain elevation data (DTED), information about roads becomes more accurate, up to date, and accessible. Looking for a map in the Internet is at fingertips with a least cost (i.e., distance or time) route plotted to a destination. Road and terrain information has been used in the past for navigation via terrain contour matching. Other examples include the increasingly popular use of digital maps for automobile navigation with a Global Positioning System (GPS) receiver and terrain-aided navigation for aircraft.

This paper is concerned with tracking of ground targets on roads and investigates possible ways to improve target state estimation via fusing a target's track with information about the road along which the target is traveling. A target track is estimated using a surveillance radar whereas a digital map provides the road network of the region under surveillance. Target tracking is not unfamiliar with road maps. For example, target tracks are represented by colorful dots and lines blinking along road networks on a big screen, often on top of a topographic or satellite image, in a situation room, in an air traffic control tower, and on a radar operator screen. In these applications, however, target tracks and road networks are merely displayed together with little or no interaction in the data processing level.

When the information about roads is as accurate as (or even better than) radar measurements, it is naturally desired to incorporate such information into target state estimation. When a vehicle travels off-road or on an unknown road, the state estimation problem is unconstrained. However, when the vehicle is traveling on a known road, be it straight or curved, the state estimation problem can be cast as constrained with the road network information available from digital road/terrain maps. In the past, such constraints are often ignored (or left for the users to perceive it as in the display example mentioned above). The resulting estimates, even obtained with the Kalman filter, cannot be optimal because they do not make full use of this additional information about state constraints.

To use such state constraints, previous attempts can be put into several groups. The first group is to incorporate road information into the state estimation process. One technique is to reduce the system model parameterization. Another technique is to translate the state constraints onto the state process and/or observation noise covariance matrix for the estimation filter [10]. The use of variable structure IMM (VS-IMM) methods also belongs to this group [7, 18, 19, 22]. Yet another technique is to project a dynamic system onto linear state constraints and then apply the Kalman filter to the projected systems [11]. Similarly, for nonlinear state constraints, there is the one-dimensional (1D) representation of a target motion along a curvilinear road [27].

The technique to model bounded random variables with truncated densities also belongs to this group, which is easily workable with such nonlinear filters as a particle filter [2, 4, 25, 26]. Road map information can also be integrated within Multi-Hypothesis Tracking (MHT) [12, 13].

The second group is to treat state constraints as pseudo measurements [8]. For a road segment, its analytic model not only constrains the target position but also the direction of the target's velocity. Indeed, the target velocity is closely aligned with the road orientation for a linear segment and with the tangent vector at the target position for a nonlinear segment. Furthermore, an estimate of centripetal acceleration can be obtained given the curvature and the target speed [27].

In the third group, an unconstrained Kalman filter solution is first obtained and then the unconstrained state estimate is projected onto the constrained surface [24]. This technique can also be viewed as post-processing (estimation or updating) correction [28] or track-to-road fusion as referred to in this paper. In conventional track fusion, two or more tracks are available, each consisting of an estimate of the underlying target trajectory with its estimation error covariance. The fused track is typically found that minimizes the sum of covariance-weighted state errors squared [5, 6]. In contrast to this conventional track fusion that operates on individual states, fusion of tracks with roads involves a state value (a point) and a subset of state values (an arc or interval). In this paper, roads are modeled with analytic functions and their fusion with a target track is therefore formulated as linear or nonlinear state constraints in an optimization procedure.

Although this paper presents a new technique for the third group, it is interesting to think of it relative to the first group in much the same way track fusion is compared with measurement fusion. In measurement fusion, measurements from all sensors are made available to a centralized tracker, which has the potential to fuse out the best estimate. However, measurement fusion may not be practical for distributed sensors wherein gathering all raw measurements is often limited by network transmission bandwidth and latency. Track fusion is frequently used as acceptable compromise between performance and cost.

Similarly, fusion of tracks with road constraints (in the third group) may not perform as well as an algorithm that incorporates road maps directly into the filtering process (in the first group). However, it has many merits of its own. First, it is simple and can be retrofitted into existing trackers as an add-on module without changes to the trackers. Since the tracks are obtained without constraints, it can easily switch between off-road and on-road operations when road information is available and the unconstrained tracks are deemed close to roads. Second, an up-to-date accurate road map may not be available to individual sensors but only at a fusion center. In this case, the algorithm of track-to-road fusion as presented in this paper can be applied directly whereas those in the first group cannot. Third, as noted in [18], the IMM methods based on road maps do not always perform better than those without road maps particularly when the updating interval is long. In contrast, constraining an on-road target's track onto a road (fusing) has no such a problem. Fourth, the track-to-road fusion algorithm goes beyond target tracking to navigation for instance where it can be used to loosely integrate GPS fixes and digital maps [16].

In this paper, we therefore focus on the third group and in particular present an optimization procedure for nonlinear state constraints which is shown to be superior to the linear approximation of nonlinear state constraints as suggested in [24].

There are a host of constrained nonlinear optimization techniques [15]. Primal methods search through the feasible region determined by the constraints. Penalty and barrier methods approximate constrained optimization problems by unconstrained problems through modifying the objective function (e.g., add a term for higher price if a constraint is violated). Instead of the original constrained problem, dual methods attempt to solve an alternate problem (the dual problem) whose unknowns are the Lagrangian multipliers of the first problem. Cutting plane algorithms work on a series of ever-improving approximating linear programs whose solutions converge to that of the original problem. Lagrangian relaxation methods are widely used in discrete constrained optimization problems.

In addition, moving horizon estimation reformulates the estimation problem as quadratic programming over a moving, fixed-size estimation window and has become an important approach to constrained nonlinear estimation [20]. Another approach to constrained linear estimation is to exploit the Lagrangian duality. Indeed, a constrained linear estimation problem is shown to be a particular nonlinear optimal control problem in [9]. Constrained state estimation has also been studied from a game-theoretical point of view (also called the minimax or H_∞ estimation) in [23].

In this paper, the constrained optimization is solved with the Lagrangian multiplier, leading to a closed-form solution for linear constraints and an iterative solution for nonlinear constraints. In the latter case, we present a method that allows for the use of second-order nonlinear state constraints exactly. The method can provide better approximation to higher order nonlinearities. The new method is based on a computational algorithm that iteratively finds the Lagrangian multiplier. The use of a second-order constraint versus linearization is a tradeoff between reducing approximation errors to higher-order nonlinearities and keeping the problem computationally tractable.

A nonlinear constraint can be approximated with linear constraints in a piecewise fashion. By judicious selection of the number of linear segments and their placement (i.e., the point around which to linearize), a reasonably good performance can be expected. In the

limit, a nonlinear function is represented by a piecewise function composed of an infinite number of linear segments. This naturally leads to the use of nonlinear constraints. As such, the proposed nonlinear constrained solution for curved roads is not only more accurate but also less complicated in implementation than a piecewise linearized constrained approach, to be shown later in simulation examples.

Although the main results are restricted to state equality constraints, it can be extended to inequality constraints. According to [24], the inequality constraints can be checked at each time step of filtering. If the inequality constraints are satisfied at a given time step, no action is taken since the inequality constrained problem is solved. If the inequality constraints are not satisfied at a given time step, then the constrained solution is applied to enforce the constraints.

The paper is organized as follows. Section 2 presents linearly constrained state estimation for fusion of tracks with linear road segments. Section 3 presents an iterative solution for fusion of tracks with nonlinear road segments. In both cases, geometric interpretations of the solutions are provided for special cases. In Section 4, computer simulation results are presented to illustrate the algorithms. Finally, Section 5 provides concluding remarks and suggestions for future work.

2. FUSION OF TRACKS WITH LINEAR ROAD SEGMENTS

When a road segment is straight, it can be modeled as a linear state constraint. In this section, we first summarize the results for linearly constrained state estimation [24] as an approach to fusion of tracks with linear road segments. We then show that this linearly constrained state estimation is equivalent to use of constraints as measurements in state update. Finally, we provide a simple geometric interpretation of the linearly constrained state estimation for track-to-road fusion.

2.1. Linearly Constrained State Estimation for Track-to-Road Fusion

Consider a linear time-invariant discrete-time dynamic system together with its measurement as

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \tag{1a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \tag{1b}$$

where the subscript k is the time index, $\mathbf{x}$ is the state vector, $\mathbf{u}$ is a known input, $\mathbf{y}$ is the measurement, and $\mathbf{w}$ and $\mathbf{v}$ are state and measurement noise processes, respectively. It is implied that all vectors and matrices have compatible dimensions, which are omitted for simplicity.

The goal is to find an estimate denoted by $\hat{\mathbf{x}}_k$ of $\mathbf{x}_k$ given the measurements up to time k denoted by $Y_k = \{\mathbf{y}_0, \ldots, \mathbf{y}_k\}$. Under the assumptions that the state and measurement noises are uncorrelated zero-mean white Gaussian with $\mathbf{w} \sim N\{0, \mathbf{Q}\}$ and $\mathbf{v} \sim N\{0, \mathbf{R}\}$ where $\mathbf{Q}$ and $\mathbf{R}$ are positive semi-definite covariance matrices, the Kalman filter provides an optimal estimator in the form of $\hat{\mathbf{x}}_k = E\{\mathbf{x}_k \mid Y_k\}$ [3]. Starting from an initial estimate $\hat{\mathbf{x}}_0 = E\{\mathbf{x}_0\}$ and its estimation error covariance matrix $\mathbf{P}_0 = E\{(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T\}$ where the superscript ^{T} stands for matrix transpose, the Kalman filter equations specify the propagation of $\hat{\mathbf{x}}_k$ and $\mathbf{P}_k$ over time and the update of $\hat{\mathbf{x}}_k$ and $\mathbf{P}_k$ by measurement $\mathbf{y}_k$ as

$$\bar{\mathbf{x}}_{k+1} = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_k \tag{2a}$$

$$\bar{\mathbf{P}}_{k+1} = \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{Q} \tag{2b}$$

$$\hat{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_{k+1} + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \mathbf{C}\bar{\mathbf{x}}_{k+1}) \tag{2c}$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C})\bar{\mathbf{P}}_{k+1} \tag{2d}$$

$$\mathbf{K}_{k+1} = \bar{\mathbf{P}}_{k+1}\mathbf{C}^T(\mathbf{C}\mathbf{P}_k\mathbf{C}^T + \mathbf{R})^{-1} \tag{2e}$$

where $\bar{\mathbf{x}}_{k+1}$ and $\bar{\mathbf{P}}_{k+1}$ are the predicted state and prediction error covariance, respectively.

Now in addition to the dynamic system of (1), we are given the linear state constraint equation

$$\mathbf{D}\mathbf{x}_k = \mathbf{d} \tag{3a}$$

where $\mathbf{D}$ is a known constant matrix of full rank, $\mathbf{d}$ is a known vector, and the number of rows in $\mathbf{D}$ is the number of constraints, which is assumed to be less than the dimension of states. If $\mathbf{D}$ is a square matrix, the state is fully constrained and can thus be solved by inverting (3a). Although no time index is given to $\mathbf{D}$ and $\mathbf{d}$ in (3a), it is implied that they can be time-dependent, leading to piecewise linear constraints.

The information about a target traveling along a linear road segment is well modeled by (3a) and illustrated in Fig. 1. As shown, the road is specified by the orientation θ defined as the angle of its normal vector $\mathbf{n}$ relative to the x-axis and the distance to the origin r. The unit vectors pointing along the road and perpendicular to the road are given by $\boldsymbol{\mu} = [-\sin\theta, \cos\theta]^T$ and $\mathbf{n} = [\cos\theta, \sin\theta]^T$, respectively. Clearly, a target at position $\mathbf{p} = [x, y]^T$ with velocity $\mathbf{v} = [\dot{x}, \dot{y}]^T$ satisfies the linear constraints $\mathbf{p}^T\mathbf{n} = r$ and $\mathbf{v}^T\mathbf{n} = 0$. These two equations can be easily put together into the format of (3a) with the corresponding $\mathbf{D}$ and $\mathbf{d}$ given below.

$$\mathbf{D} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & \cos\theta & 0 & \sin\theta \end{bmatrix}, \qquad \mathbf{d} = \begin{bmatrix} r \\ 0 \end{bmatrix} \tag{3b}$$

The constrained Kalman filter according to [24] is constructed by directly projecting the unconstrained state estimate $\hat{\mathbf{x}}_k$ onto the constrained surface $\mathcal{S} = \{\mathbf{x} : \mathbf{D}\mathbf{x} = \mathbf{d}\}$. It is formulated as the solution to the problem

$$\breve{\mathbf{x}} = \arg\min_{\mathbf{x} \in S} (\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{W} (\mathbf{x} - \hat{\mathbf{x}}) \tag{4}$$

where $\mathbf{W}$ is a symmetric positive definite weighting matrix. The time index subscript k is dropped from

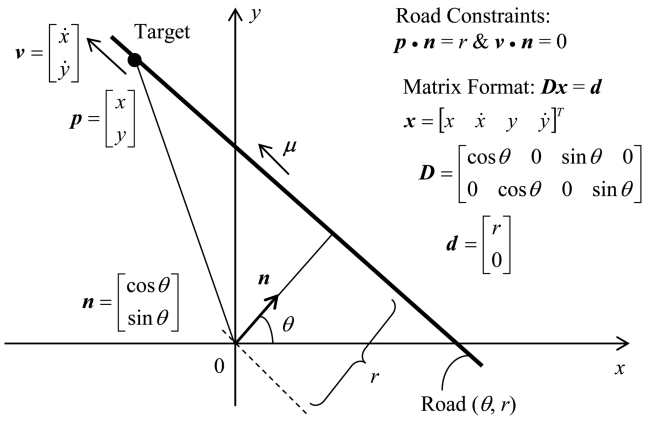


Fig. 1. Road models as linear constraints.

variables in (4) for simplicity. When $\mathbf{W} = \mathbf{I}$, the cost function of (4) is the standard least squares formulation. If $\mathbf{W}$ is chosen based on the estimation error covariance matrix $\mathbf{P}$, it becomes the weighted least squares solution.

Derived using the Lagrangian multiplier technique in Appendix A, the solution to the constrained optimization in (4) is given by [24]

$$\breve{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{W}^{-1}\mathbf{D}^T(\mathbf{D}\mathbf{W}^{-1}\mathbf{D}^T)^{-1}(\mathbf{D}\hat{\mathbf{x}} - \mathbf{d}). \tag{5}$$

Several interesting statistical properties of the constrained Kalman filter are presented in [24]. This includes the fact that the constrained state estimate as given by (5) is an unbiased state estimate for the system in (1) subject to the constraint in (3a) for a known symmetric positive definite weighting matrix $\mathbf{W}$. Furthermore when $\mathbf{W} = \mathbf{P}^{-1}$, the constrained state estimate has a smaller error covariance than that of the unconstrained state estimate, and it is actually the smallest for all constrained Kalman filters of this type.

2.2. Track-to-Road Fusion Architectures[1]

According to (4), the fusion of a target track (an unconstrained state estimate) $\hat{\mathbf{x}}_k$ with a road segment represented by a surface in the state space $\mathcal{S}$ is cast as a constrained least squares optimization problem, yielding the constrained solution $\breve{\mathbf{x}}$ and its estimation error covariance $P_{\breve{\mathbf{x}}}$. This leads to two possible implementation schemes. One is the open-loop architecture without feedback as shown in Fig. 2(a) and the other is the closed-loop architecture with feedback as shown in Fig. 2(b).

In the open-loop architecture of Fig. 2(a), the unconstrained solution can be used to help select the proper road constraints prior to fusion and the fused solution may be further used to refine road constraints for future target movements. However, the fused state is not fed back to the unconstrained tracker.

In contrast, the closed-loop architecture of Fig. 2(b) feeds back the fused state to the unconstrained Kalman tracker (i.e., to replace the state with the fused state). This has the advantage of keeping the one-step-ahead prediction closely aligned to the road estimates.

There are several issues to trade off when making the choice of one architecture versus the other. The closed-loop scheme needs to alter the unconstrained filter and its implementation therefore requires internal access. Further, a two-way data link may be necessary if the tracker and the fusion center are not co-located.

The open-loop architecture is simple and can be retrofitted into existing trackers as an add-on module without changes to the trackers. Since the tracks are obtained without constraints, it can easily switch between off-road and on-road operations. It is particularly useful for cases where no up-to-date accurate road map (e.g., the latest satellite imagery) is available to individual sensors but at a fusion center. In this paper, the open-loop scheme is implemented in the simulation examples presented in Section 4.

As shown in Fig. 2, an important step leading to the track-to-road fusion is the road constraint generation. It consists of two major parts, namely, creating an analytic

[1]This subsection is added based on the editor's comments.

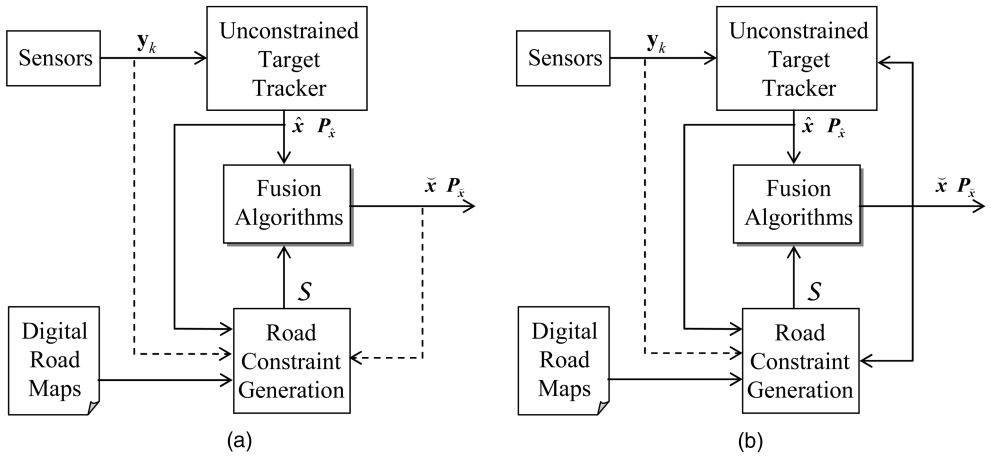


Fig. 2. Track-to-road fusion architectures.

representation for a given road segment and selecting the correct road segment(s) for fusion.

In a digital geographic database, road network is stored as a series of waypoints, a layer of the vector map (VMAP). The waypoints are typically extracted from survey data and aerial imagery among others and the density or spacing of waypoints is determined by the map resolution, which may be nonuniform. Although local survey data may contain the radius and turn center of a curved road segment, the waypoints themselves do not define the functions representing the road. To apply the road-constrained optimal fusion method, it is necessary to generate the constraint function based on the waypoints in the database. The most typical approach would use linear segments to connect the waypoints as evident from Google Map, MapQuest, or MSN Maps when zooming in. The waypoint connections defines a line representing road, which can be a simple line connecting two waypoints or a tangent passing through a waypoint. Alternately, a spline (a piecewise polynomial function) can be used to define the road in between the points, leading to a nonlinear function defining the road.

Ideally, the number of waypoints used to define the road is generated such that the maximum error between the actual road and the mathematical model for the road (linear segments, spline, etc.) is less than some allowable value. However, waypoints in most digital maps are pre-determined and fixed. Depending on the map resolution and sensor accuracy, when the error associated with the constraints becomes larger than the error in the sensor measurement, the benefits of using such constraints diminish. It is therefore desired to have a road modeling system that generates the waypoints to support "adaptive sampling" so that the error between the road and the road model is always less than some limit. The use of an analytic nonlinear representation, rather than fixed waypoints with linear segments, is a possible way toward adaptive sampling and resampling.

The second aspect of the road constraint generation shown in Fig. 2 is constraint selection, which identifies which road the target is on and the closest waypoints on the road and then produces the constraint function for those points of the road. Similar to the problem of target tracking with measurement origin uncertainties where data association is applied prior to measurement updating, the track-to-road fusion necessitates road constrained data association (RCDA) especially with closely spaced roads and around intersections. This association can be either measurement-based or predicted state-based and a data history may be needed to ascertain the winning hypothesis.

For an identified road, it then comes to select a piecewise constraint model. Without pre-determined analytic models available for the road segment, it is possible to perform on-line synthesis. For example, from two closest waypoints to a measurement, a line representation can be computed for those points. Or a spline representation of the road can be computed for the nearest three points of the digital map. A nonlinear representation (from the spline for instance) further allows for piecewise linearization with the point for the linearization chosen near the measurement or near the estimated track state. Iterative linearization can be used to refine the linearized constraints if necessary.

For lines between fixed waypoints or piecewise linearized segments from a nonlinear model, the linear constrained optimization method of this section can be applied. For curved roads, the nonlinear optimization method presented in Section 3 can be used advantageously when a nonlinear representation of a road is available.

The aspects of constraint modeling and selection are not further discussed in this paper. Another important issue that is not addressed either in this paper is possible errors in digital maps such as bias and misorientation for linear road segments and erroneous radius and turn center for curved segments. We leave it for future treatment but focus on fusion methodology in this paper.

2.3. Linear Road Constraint as Pseudo Measurement

As described above, the linear constrained estimator (5) can be obtained by different methods. It is shown in this section that it is also equivalent to the solution where the linear state constraints are considered as pseudo measurements.

For the linear time-invariant discrete-time dynamic system (1a) and its measurement (1b), consider the linear state constraint (3) as an additional measurement to the system, which can be used to perform the filter measurement update (2c) and (2d) right after (1b) without the filter time propagation (2a) and (2b) (i.e., stay the same). To apply (2), we identify the following equivalence:

$$\mathbf{C} = \mathbf{D}, \qquad \mathbf{R} = \mathbf{0}, \qquad \mathbf{y}_k = \mathbf{d}. \tag{6}$$

Consider $(\hat{\mathbf{x}}_k^{(i)}, \mathbf{P}_k^{(i)})$ as the constrained state and covariance after the ith iteration update with the constraints at time k. With this notation, $(\hat{\mathbf{x}}_k^{(i)}, \mathbf{P}_k^{(i)}) = (\hat{\mathbf{x}}_k, \mathbf{P}_k)$ estimate for $i = 0$ is the unconstrained state estimate and covariance at time k. The Kalman filter gain is given by

$$\mathbf{K}_k^{(i+1)} = \mathbf{P}_k^{(i)}\mathbf{D}^T(\mathbf{D}\mathbf{P}_k^{(i)}\mathbf{D}^T)^{-1}. \tag{7}$$

The updated state and error covariance becomes:

$$\hat{\mathbf{x}}_k^{(i+1)} = \hat{\mathbf{x}}_k^{(i)} + \mathbf{P}_k^{(i)}\mathbf{D}^T(\mathbf{D}\mathbf{P}_k^{(i)}\mathbf{D}^T)^{-1}(\mathbf{d} - \mathbf{D}\hat{\mathbf{x}}_k^{(i)}) \tag{8}$$

$$\mathbf{P}_k^{(i+1)} = \mathbf{P}_k^{(i)} - \mathbf{P}_k^{(i)}\mathbf{D}^T(\mathbf{D}\mathbf{P}_k^{(i)}\mathbf{D}^T)^{-1}\mathbf{D}\mathbf{P}_k^{(i)}. \tag{9}$$

If we choose $\mathbf{W} = (\mathbf{P}_k^{(i)})^{-1}$, (8) becomes

$$\hat{\mathbf{x}}_k^{(i+1)} = \hat{\mathbf{x}}_k^{(i)} + \mathbf{W}^{-1}\mathbf{D}^T(\mathbf{D}\mathbf{W}^{-1}\mathbf{D}^T)^{-1}(\mathbf{d} - \mathbf{D}\hat{\mathbf{x}}_k^{(i)}) \tag{10a}$$

$$= \hat{\mathbf{x}}_k^{(i)} - \mathbf{W}^{-1}\mathbf{D}^T(\mathbf{D}\mathbf{W}^{-1}\mathbf{D}^T)^{-1}(\mathbf{D}\hat{\mathbf{x}}_k^{(i)} - \mathbf{d}) \tag{10b}$$

$$= \hat{\mathbf{x}}_k - \mathbf{W}^{-1}\mathbf{D}^T(\mathbf{D}\mathbf{W}^{-1}\mathbf{D}^T)^{-1}(\mathbf{D}\hat{\mathbf{x}}_k - \mathbf{d}) \tag{10c}$$

which is exactly the same as the solution given by (5).

This equivalence affords a possible way to incorporate uncertainty in road modeling such as bias, width, and mis-orientation through pseudo measurement error covariance matrix $\mathbf{R}$. In the ideal case where roads are assumed to be known perfectly, this $\mathbf{R}$ is set to zero. Inequality constraint is another way to handle uncertainty if errors are within certain known bounds. Furthermore, when the track-to-road fusion is based on optimization with a least-squares criterion, it is possible to introduce weightings to account for directional errors given by covariance matrices of the track and/or the road.

2.4. Geometric Interpretation

Assume that the state dimension is n and the number of linear constraints is $m < n$. For $\mathbf{x} \in \mathbb{R}^n$, the constraint $\mathcal{S} = \{\mathbf{x} : \mathbf{D}\mathbf{x} = \mathbf{d}\}$ constitutes a surface in $\mathbb{R}^n$. It is shown in Appendix B that for the case where $\mathbf{W} = \mathbf{I}$, the linear constrained estimation (5) is the orthogonal projection of the unconstrained estimate onto the constraining surface. This offers a geometric interpretation and provides a theoretical justification of the intuitive practice of finding a point along the road that is of the shortest distance.

The theory still holds for $\mathbf{W} \neq \mathbf{I}$. The proof is given in Appendix C. The results presented in this and previous sections complement the work of [24], providing an interesting geometric interpretation to the linear constrained estimation by estimate projection.

3. FUSION OF TRACKS WITH NONLINEAR ROAD SEGMENTS

When a road segment is curved, it can be modeled as a nonlinear state constraint. In this section, we first analyze the linearizing approach and the associated constraint approximation error. We then present an iterative solution to a second order state constraint. Finally, we offer a geometric interpretation of the solution under a circular constraint and outline a simple approach to a more general second order state constraint problem of practical significance.

3.1. Approximation Errors in Constraint Linearization

To deal with nonlinearity, a simple approach is to project the unconstrained state estimate onto linearized state constraints. Once the constraints are linearized, the results presented in the previous section for linear cases can be applied. However, linearization introduces constraint approximation error, which is a function of the nonlinearity and, more importantly, of the point around which the linearization takes place. This may lead to an undesired divergence problem as analyzed below.

Consider the nonlinear state constraint of the form

$$\mathbf{g}(\mathbf{x}) = \mathbf{d}. \tag{11}$$

We can expand the nonlinear state constraints about a constrained state estimate $\breve{\mathbf{x}}$ and for the ith row of (11), we have

$$g_i(\mathbf{x}) - d_i = g_i(\breve{\mathbf{x}}) + \mathbf{g}_i'(\breve{\mathbf{x}})^T(\mathbf{x} - \breve{\mathbf{x}}) + \frac{1}{2!}(\mathbf{x} - \breve{\mathbf{x}})^T\mathbf{g}_i''(\breve{\mathbf{x}}) + (\mathbf{x} - \breve{\mathbf{x}}) + \cdots - d_i = 0 \tag{12}$$

where the superscripts $'$ and $''$ denote the first and second partial derivatives.

Keeping only the first-order terms as suggested in [24], some rearrangement leads to

$$\mathbf{g}'(\breve{\mathbf{x}})^T\mathbf{x} \approx \mathbf{d} - \mathbf{g}(\breve{\mathbf{x}}) + \mathbf{g}'(\breve{\mathbf{x}})^T\breve{\mathbf{x}} \tag{13}$$

where $\mathbf{g}(\mathbf{x}) = [\ldots g_i(\mathbf{x}) \ldots]^T$, $\mathbf{d} = [\ldots d_i \ldots]^T$, and $\mathbf{g}'(\mathbf{x}) = [\ldots \mathbf{g}_i'(\mathbf{x}) \ldots]$. An approximate linear constraint is therefore formed by replacing $\mathbf{D}$ and $\mathbf{d}$ in (3) with $\mathbf{g}'(\mathbf{x})^T$ and $\mathbf{d} - \mathbf{g}(\breve{\mathbf{x}}) + \mathbf{g}'(\breve{\mathbf{x}})^T\breve{\mathbf{x}}$, respectively.

Fig. 3 illustrates this linearization process and identifies possible errors associated with linear approximation of a nonlinear state constraint. As shown, a previous constrained state estimate $\breve{\mathbf{x}}^-$ lies somewhere on the constrained surface but is away from the true state $\mathbf{x}$. The projection of the unconstrained state estimate $\hat{\mathbf{x}}$ onto the approximate linear state constraint produces the current constrained state estimate $\breve{\mathbf{x}}^+$, which is however subject to the constraint approximation error. Clearly, the further away is $\breve{\mathbf{x}}^-$ from $\mathbf{x}$, the larger is the approximation-introduced error. More critically, such an approximately linear constrained estimate may not satisfy the original nonlinear constraint specified in (11). It is therefore desired to reduce this approximation-introduced error by including higher-order terms while keeping the problem computationally tractable. One possible approach is presented in the next section.

As discussed in Section 2.2, when waypoints of a digital map are used to construct linear constraints directly, their modeling error is related to a large extent to the coarseness of waypoints, which is determined in turn by the map resolution.

Working with an analytic road model, simply curve-fitted from fixed waypoints for instance, provides the opportunity for possible "iterated linearization" or "adaptive sampling" so as to maintain small uniform linearization errors. As shown in Fig. 3, when the linearization point is far away from the true state, the lineariza-

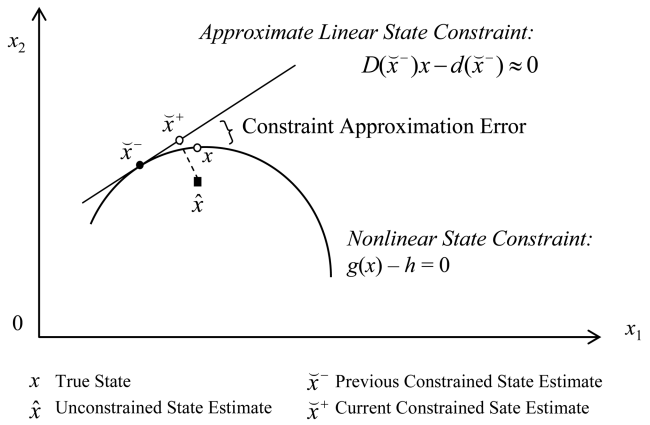


Fig. 3. Errors in linear approximation of nonlinear state constraints.

tion is poor. In this particular case, the linearization point is the predicted state $\breve{\mathbf{x}}^-$, which happens to be on road. However, due to target motion, this predicted state is offset from the true state $\mathbf{x}$. The linear constrained state estimate $\breve{\mathbf{x}}^+$ is now "closer" to the true state than the predicted one and can be used to re-linearize the function as done in an iterated extended Kalman filter. This iterated linearization may reduce linearization error in a sense but cannot guarantee a smaller state estimation error because the linear constrained state estimate and its iterations may not always fall onto the road.

At a first glance, a curved road can be well approximated with a sufficient number of waypoints where the linearization points are critically placed (i.e., the waypoint sample rate is sufficient to keep the error between the road and the road model small). In the limit, a piecewise linear approximation converges to a continuous function; and the direct use of a nonlinear constraint itself, rather than its approximation, becomes natural.

In practical cases, however, only a limited number of waypoints are available. For sharp turns, linear approximation errors dominate. As shown later in Section 4.1, the selection of a linear segment and in particular the transition from one segment to another is a rather involved process. On the other hand, the track-to-road fusion is considerably simplified with nonlinear constrained optimization as described below.

3.2. Iterative Solution to Second-Order Constraints

Naturally formed roads tend to have more bends and turns of irregular shapes (high nonlinearity). Even highways have to follow terrain contours when crossing mountains. Locally, however, it suffices to represent a curved road segment by a second-order state constraint function as

$$f(\mathbf{x}) = [\mathbf{x}^T \;\; 1] \begin{bmatrix} \mathbf{M} & \mathbf{m} \\ \mathbf{m}^T & m_0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

$$= \mathbf{x}^T\mathbf{M}\mathbf{x} + \mathbf{m}^T\mathbf{x} + \mathbf{x}^T\mathbf{m} + m_0 = 0 \qquad (14)$$

which can be viewed as a second-order approximation to an arbitrary nonlinearity in a digital terrain map.

Similar to (4), we can formulate the projection of an unconstrained state estimation onto a nonlinear constraint surface as the constrained least-squares optimization problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} (\mathbf{z} - \mathbf{H}\mathbf{x})^T(\mathbf{z} - \mathbf{H}\mathbf{x}) \qquad (15a)$$

$$\text{subject to} \quad f(\mathbf{x}) = 0. \qquad (15b)$$

If we let $\mathbf{W} = \mathbf{H}^T\mathbf{H}$ and $\mathbf{z} = \mathbf{H}\hat{\mathbf{x}}$, the formulation in (15) becomes the same as in (4). In a sense, (15) is a more general formulation because it can also be interpreted as a nonlinear constrained measurement update or a projection in the predicted measurement domain.

The solution to the constrained optimization (15) can be obtained again using the Lagrangian multiplier technique, which is detailed in Appendix D, as

$$\hat{\mathbf{x}} = \mathbf{G}^{-1}\mathbf{V}(\mathbf{I} + \lambda\mathbf{\Sigma}^T\mathbf{\Sigma})^{-1}\mathbf{e}(\lambda) \qquad (16a)$$

$$q(\lambda) = \sum_i \frac{e_i^2(\lambda)\sigma_i^2}{(1+\lambda\sigma_i^2)^2} + 2\sum_i \frac{e_i(\lambda)t_j}{1+\lambda\sigma_i^2} + m_0 = 0 \qquad (16b)$$

where $\mathbf{G}$ is an upper right diagonal matrix resulting from the Cholesky factorization of $\mathbf{W} = \mathbf{H}^T\mathbf{H}$ as

$$\mathbf{W} = \mathbf{G}^T\mathbf{G} \qquad (16c)$$

$\mathbf{V}$, an orthonormal matrix, and $\mathbf{\Sigma}$, a diagonal matrix with its diagonal elements denoted by σ_i, are obtained from the singular value decomposition (SVD) of the matrix $\mathbf{L}G^{-1}$ as

$$\mathbf{L}G^{-1} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \qquad (16d)$$

where $\mathbf{U}$ is the other orthonormal matrix of the SVD and $\mathbf{L}$ results from the factorization $\mathbf{M} = \mathbf{L}^T\mathbf{L}$, and

$$\mathbf{e}(\lambda) = [\ldots e_i(\lambda), \ldots]^T = \mathbf{V}^T(\mathbf{G}^T)^{-1}(\mathbf{H}^T\mathbf{z} - \lambda\mathbf{m}) \qquad (16e)$$

$$\mathbf{t} = [\ldots t_i \ldots]^T = \mathbf{V}^T(\mathbf{G}^T)^{-1}\mathbf{m}. \qquad (16f)$$

As a nonlinear equation in λ, it is difficult to find a closed-form solution in general for the nonlinear equation $q(\lambda) = 0$ in (16b). Numerical root-finding algorithms may be used instead. For example, the Newton's method is used below. Denote the derivative of $q(\lambda)$ with respect to λ as

$$\dot{q}(\lambda) = 2\sum_i \frac{e_i(\lambda)\dot{e}_i(1+\lambda\sigma_i^2)\sigma_i^2 - e_i^2(\lambda)\sigma_i^4}{(1+\lambda\sigma_i^2)^3} + 2\sum_i \frac{\dot{e}_i t_i(1+\lambda\sigma_i^2) - e_i(\lambda)t_i\sigma_i^2}{(1+\lambda\sigma_i^2)^2} \qquad (17a)$$

where

$$\dot{\mathbf{e}} = [\ldots \dot{e}_i \ldots]^T = -\mathbf{V}^T(\mathbf{G}^T)^{-1}\mathbf{m}. \qquad (17b)$$

Then the iterative solution for λ is given by

$$\lambda_{k+1} = \lambda_k - \frac{q(\lambda_k)}{\dot{q}(\lambda_k)} \qquad \text{starting with} \quad \lambda_0 = 0. \qquad (18)$$

The iteration stops when $|\lambda_{k+1} - \lambda_k| < \tau$, a given small value or the number of iterations reaches a prespecified number. Then bringing the converged Lagrangian multiplier λ back to (16a) provides the constrained optimal solution.

Now consider the special case where $\mathbf{W} = \mathbf{H}^T\mathbf{H}$, $\mathbf{z} = \mathbf{H}\hat{\mathbf{x}}$, and $\mathbf{m} = \mathbf{0}$, that is, a quadratic constraint on the state. Under these conditions, $\mathbf{t} = 0$ and $\mathbf{e}$ is no longer a function of λ so its derivative relative to λ vanishes, $\dot{\mathbf{e}} = 0$. The quadratic constrained solution is then given by

$$\breve{\mathbf{x}} = (\mathbf{W} + \lambda\mathbf{M})^{-1}\mathbf{W}\hat{\mathbf{x}} \tag{19a}$$

where the Lagrangian multiplier λ is obtained iteratively as in (18) with the corresponding $q(\lambda)$ and $\dot{q}(\lambda)$ given by

$$q(\lambda) = \sum_i \frac{e_i^2\sigma_i^2}{(1+\lambda\sigma_i^2)^2} + m_0 = 0 \tag{19b}$$

$$\dot{q}(\lambda) = -2\sum_i \frac{e_i^2\sigma_i^4}{(1+\lambda\sigma_i^2)^3}. \tag{19c}$$

The solution of (19) is also called the constrained least squares [17: pp 765–766], which was previously applied for the joint estimation of angles of arrival and calibration of channel biases of a linear array [29]. Similar techniques have been used for the design of filters for radar applications [1] and in robust minimum variance beamforming [14]. When $\mathbf{M} = 0$, the constraint in (14) degenerates to a linear one. The constrained solution is still valid. However, the iterative solution for finding λ is no longer applicable but a closed-form solution is available instead as given in (5).

3.3. Geometric Interpolation for Simple Cases

Consider a simple example where a target travels along a circle. For this case, in fact, a closed-form solution can be derived. Assume that $\mathbf{W} = \mathbf{I}$, $\mathbf{M} = \mathbf{I}$, $\mathbf{m} = \mathbf{0}$, and $m_0 = -r^2$. Let $\mathbf{p}$ be the position components of the state $\mathbf{x}$, to which the constraint is applied. The nonlinear constraint can be equivalently written as:

$$\mathbf{p}^T\mathbf{p} = r^2. \tag{20}$$

The quadratic constrained estimate given in (19a) becomes:

$$\breve{\mathbf{p}} = (\mathbf{W} + \lambda\mathbf{M})^{-1}\mathbf{W}\hat{\mathbf{p}} = (1+\lambda)^{-1}\hat{\mathbf{p}} \tag{21}$$

where λ is the Lagrangian multiplier.

Bringing (21) back to (20) gives:

$$\breve{\mathbf{p}}^T\breve{\mathbf{p}} = \left(\frac{\hat{\mathbf{p}}}{1+\lambda}\right)^T \frac{\hat{\mathbf{p}}}{1+\lambda} = r^2. \tag{22}$$

One solution for λ is:

$$\lambda = \frac{\sqrt{\hat{\mathbf{p}}^T\hat{\mathbf{p}}}}{r} - 1 = \frac{\|\hat{\mathbf{p}}\|_2}{r} - 1 \tag{23}$$

where $\|\cdot\|_2$ stands for the L_2-norm or length for the vector.

Bringing the solution for λ in (23) back to (21) gives:

$$\breve{\mathbf{p}} = r\frac{\hat{\mathbf{p}}}{\|\hat{\mathbf{p}}\|_2}. \tag{24}$$

This indicates that for this particular case with a circular constraint, the constraining results in normalization.

This further suggests a simple solution for some practical applications. When a target is traveling along a circular path (or approximately so), one can first find the equivalent center of the circle around which to establish a new coordinate system. Then express the unconstrained solution in the new coordinate and normalize it as the constrained solution. Finally convert it back to the original coordinates. For non-circular but second-order paths, eigenvalue-based scaling may be effected following coordinate translation and rotation in order to apply this circular normalization. Reverse operations are in order to transform back to the original coordinates. For applications of high dimensionality, the scalar iterative solution of (17) may be more efficient.

4. SIMULATION RESULTS

In this section, two simulation examples are presented in the context of on-road ground vehicle tracking. The first example compares linearized and nonlinear constraining schemes for a simple tracker and the second example compares unconstrained and constrained IMM trackers.

4.1. Linearized versus Nonlinear Constraints for a Simple Tracker

In this simulation example, a ground vehicle is assumed to travel along a circular road segment as shown in Fig. 3. The turn center is chosen as the origin of the x-y coordinates and the turn radius is $r = 100$ m. The target maintains a constant turn rate of 5.7296 deg/s with an equivalent linear speed of 10 m/s. The initial state is

$$\mathbf{x}_{k=0} = [x\ \ \dot{x}\ \ y\ \ \dot{y}]^T = [100\text{ m}\ \ 0\text{ m/s}\ \ 0\text{ m}\ \ 10\text{ m/s}]^T. \tag{25}$$

The vehicle is tracked by a radar sensor with a sampling interval of $T = 1$ s. The sensor provides position measurements of the vehicle as

$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\mathbf{x}_k + \mathbf{v}_k \tag{26}$$

where the measurement error $\mathbf{v} \sim N(0, \mathbf{R})$ is a zero-mean Gaussian noise, independent in the x- and y-axis. The covariance matrix $\mathbf{R} = \text{diag}([\sigma_x^2\ \ \sigma_y^2])$ uses the particular values of $\sigma_x = \sigma_y = 7$ m in the simulation. To use the position measurement model (26), it is assumed that the radar-produced measurements in a polar frame are converted to the Cartesian frame and the errors associated with the conversion are ignored.

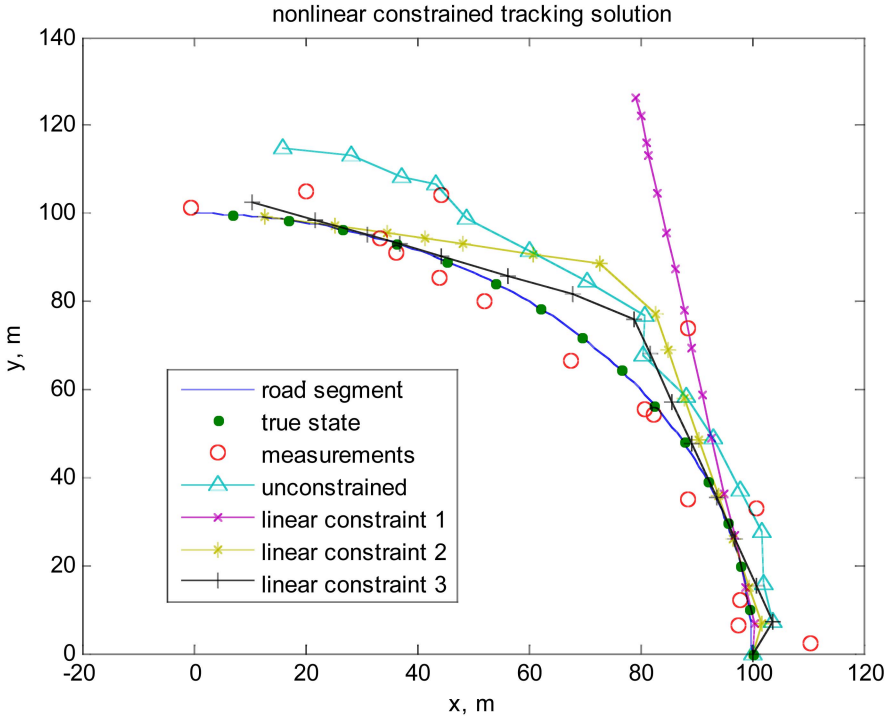


Fig. 4. Sample trajectories for linear constrained Kalman filter.

The radar implements a simple tracker based on the following discrete-time second-order kinematic model (nearly constant velocity)

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \frac{1}{2}T^2 & 0 \\ T & 0 \\ 0 & \frac{1}{2}T^2 \\ 0 & T \end{bmatrix} \mathbf{w}_k \tag{27}$$

where the process noise $\mathbf{w} \sim N(0, \mathbf{Q})$ is also a zero-mean Gaussian noise, independent of the measurement noise $\mathbf{v}$. The covariance matrix $\mathbf{Q} = \text{diag}([\sigma_{\ddot{x}}^2 \ \sigma_{\ddot{y}}^2])$ uses the particular values of $\sigma_{\ddot{x}} = \sigma_{\ddot{y}} = 0.32$ m/s^2 in the simulation.

When represented in a Cartesian coordinate system, a target traveling along a curved road is certainly subject to acceleration in both the x- and y-axis. However, no effect is made in this simulation to optimize the tracker for maneuver but merely to select $\mathbf{Q}$ and the initial conditions so as to focus on constraining the estimates. The use of an IMM filter [5] with "coordinated turn" models will be presented next in Section 4.2. The initial state is selected to be the same as the true state, i.e., $\hat{\mathbf{x}}_0 = \mathbf{x}_0$ for this example, again to focus on the aspect of track-to-road fusion, not on that of tracker design. The initial estimation error covariance is selected to be

$$\mathbf{P}_0 = \text{diag}([5^2 \text{ m}^2 \ \ 1^2 \text{ (m/s)}^2 \ \ 5^2 \text{ m}^2 \ \ 1^2 \text{ (m/s)}^2]). \tag{28}$$

Fig. 4 shows sample trajectories of the linear constrained Kalman filter. There are 5 curves and 2 series of data points in the figure. The true state is represented by a series of dots (·) at consecutive sampling instants, which is plotted on the solid line being the road segment. The corresponding measurements are a series of circles (o).

The estimates of the unconstrained Kalman filter are shown as the connected triangles (△) whereas those of linearly constrained Kalman filters are shown as the connected crosses (×), stars (*), and pluses (+) for three linear approximations of the nonlinear constraint of the curved road, respectively.

In the first approximation (the line with cross × labeled "linear constraint 1"), a single linearizing point at $\theta_1 = 10°$ is chosen to cover the entire curved road, where θ is the angle made relative to the x-axis, positive in the counter-clock direction. The linearized state constraint at θ_1 can be written as

$$\begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ 0 & \cos\theta_1 & 0 & \sin\theta_1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} r \\ 0 \end{bmatrix}. \tag{29}$$

Although all estimates are faithfully projected by the constrained filter onto this linear constraint, tangential to the curve at the linearizing point, it runs away from the true trajectory and the resulting errors continue to grow. The apparent divergence is caused by the choice of linearization.

In the second approximation (the line with star * labeled "linear constraint 2"), two linearizing points at $\theta_1 = 15°$ and $\theta_2 = 80°$ are chosen to cover the curved road with two linear segments. The switching point from one linear segment to the other in this case is at $\theta = 45°$. As shown, the estimates are projected onto one of the two linear segments. Except near the corner

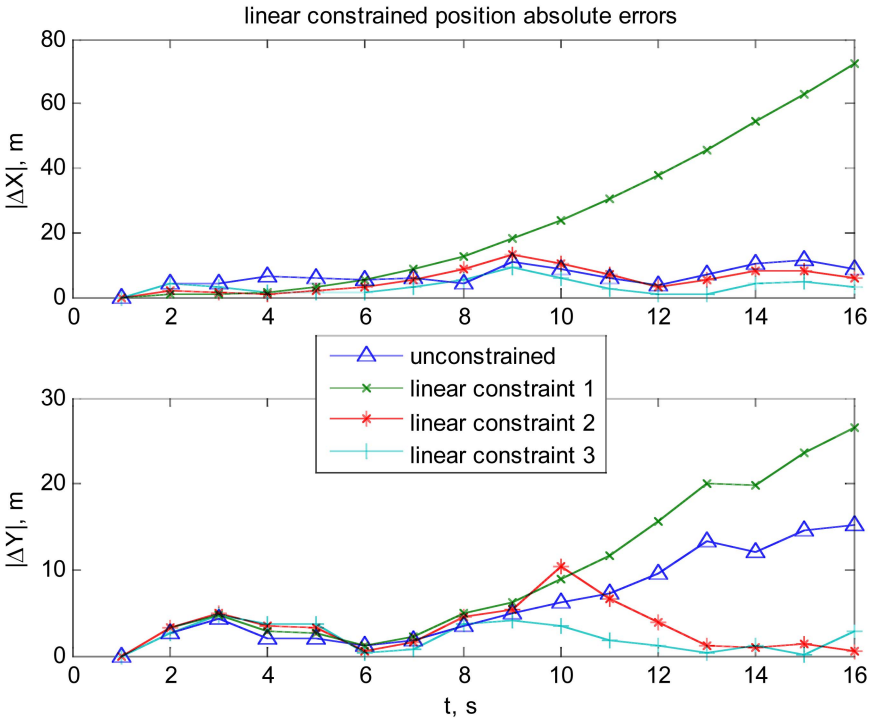


Fig. 5. Linear constrained position errors versus time.

where the two linear approximations intersect (which is far away from both linearizing points), the linear constrained estimates typically outperform the unconstrained estimates (closer to the true state). This is better illustrated in Fig. 5 where the upper plot is for the absolute position error in x while the lower plot is for the absolute position error in y, both plotted as a function of time.

Still with two linearizing points and the same switching point at $\theta = 45°$, the third approximation (the line with plus+labeled "linear constraint 3") adjusts linearizing points to $\theta_1 = 20°$ and $\theta_2 = 70°$. A better overall performance is achieved as shown in Fig. 5.

It is clear from Fig. 4 that a nonlinear constraint can be approximated with linear constraints in a piecewise fashion. By judicious selection of the number of linear segments and their placement (i.e., the point around which to linearize), a reasonably good performance can be expected. In the limit, a nonlinear function is represented by a piecewise function composed of an infinite number of linear segments. This naturally leads to the use of nonlinear constraints.

Fig. 6 shows sample trajectories of the nonlinear constrained Kalman filter. There are 2 curves and 4 series of data points in the figure. The true state is still represented by a series of dots (·) at the sampling instants, which is plotted on the solid line of road segment. The corresponding measurements are again a series of circles (o). The unconstrained Kalman filter is shown as the connected crosses (×) whereas the estimates of nonlinearly constrained Kalman filters are shown as a series of pluses (+) and stars (*) for two implementations, respectively.

The first implementation (the series of pluses +) only applies the nonlinear constraint to the position estimate whereas the second implementation (the series of stars *) applies constraints to both the position and velocity estimates. In fact, we encounter a hybrid (mixed) linear and nonlinear state constraint situation. The constrained position estimate is given by (19) for the quadratic case (equivalent to (24) for a circular road). Since the velocity direction is along the tangent of the road curve, the constrained velocity estimate is obtained by the following projection

$$\hat{\mathbf{v}}_{\text{constrained}} = (\hat{\mathbf{v}}^T_{\text{unconstrained}}\boldsymbol{\mu})\boldsymbol{\mu} \tag{30}$$

where $\hat{\mathbf{v}} = [\hat{\dot{x}}\ \ \hat{\dot{y}}]^T$ is the estimated velocity vector and $\boldsymbol{\mu} = [-\sin\theta\ \ \cos\theta]^T$ is the constrained unit direction vector associated with the constrained position at $\theta = \tan^{-1}(\hat{y}/\hat{x})$.

In the present simulation, the open-loop architecture without feedback is used. In this implementation, the unconstrained estimation error covariance is not modified after the constrained estimate is obtained using the projection algorithms (19). The implementation is therefore pessimistic (suboptimal) in the sense that it does not take into account the reduction in the estimation error covariance brought in by constraining. One consequence of this simplification is more volatile state estimates. To quantify this effect, one approach is to project the unconstrained probability density function (i.e., a normal distribution with support on the whole state space) onto the nonlinear constraint. Statistics can then be calculated from the constrained probability density function with the constraint as its support. Again,

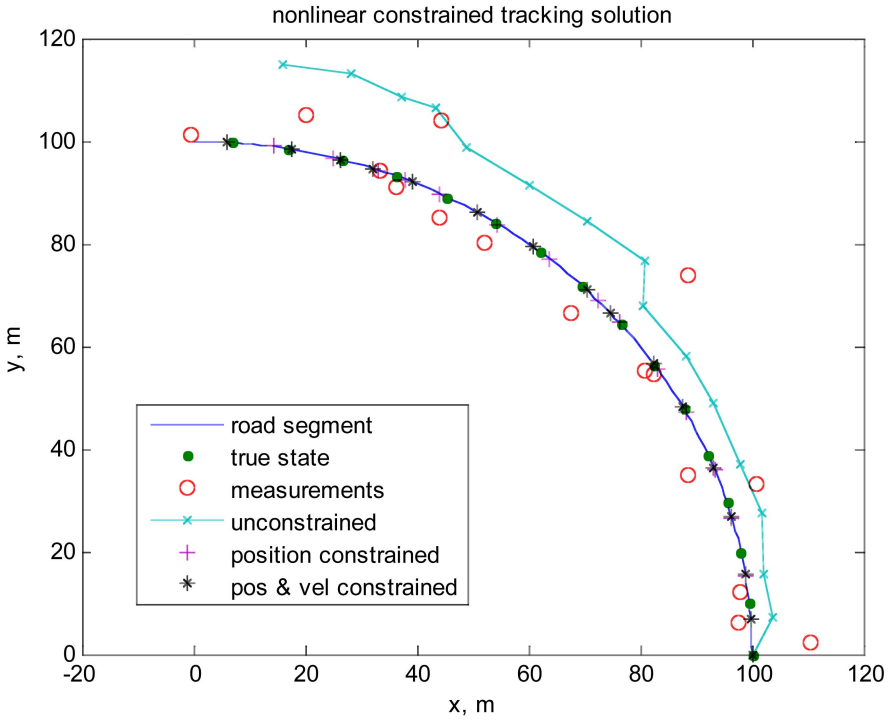


Fig. 6. Sample trajectories for nonlinear constrained Kalman filter.

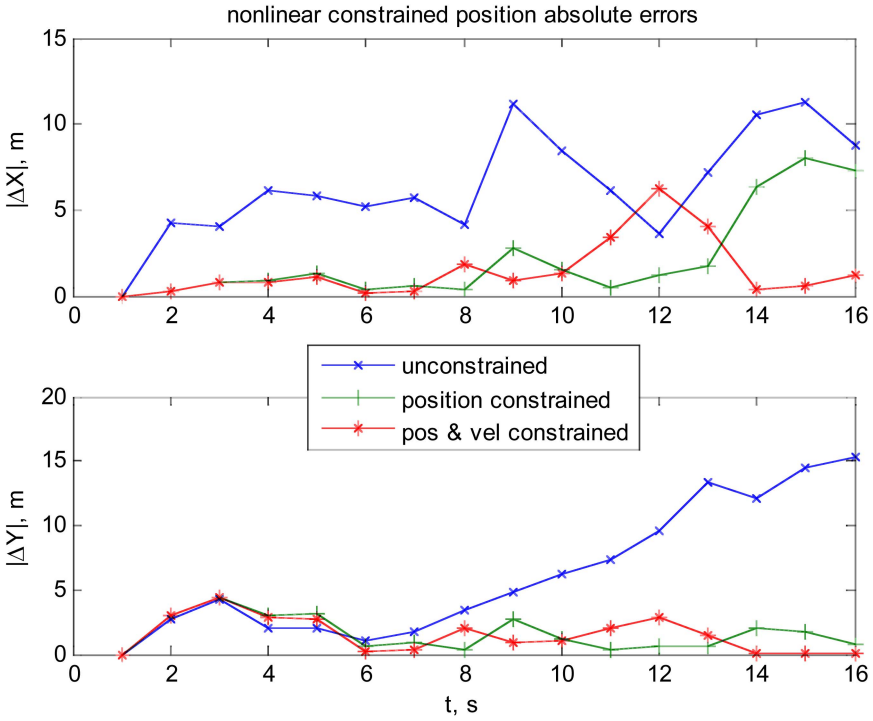


Fig. 7. Nonlinear constrained position errors versus time.

the resulting error ellipse represented by the covariance matrix is only an approximation to the second order. As explained in Section 2.2, the open-loop architecture without feedback has many merits of its own and it here provides a reference point for fusion architecture study.

As shown in Fig. 6, both the nonlinear constrained estimates fall onto the road as expected. Overall the position and velocity constrained estimates are better (closer to the true state) than the position-only constrained estimates. This is illustrated in Fig. 7 where the upper plot is for the absolute position error in x while the lower plot is for the absolute position error in y.

A Monte Carlo simulation is used to generate the root mean square (RMS) errors of state estimation. The results are based on a total of 100 runs across 16

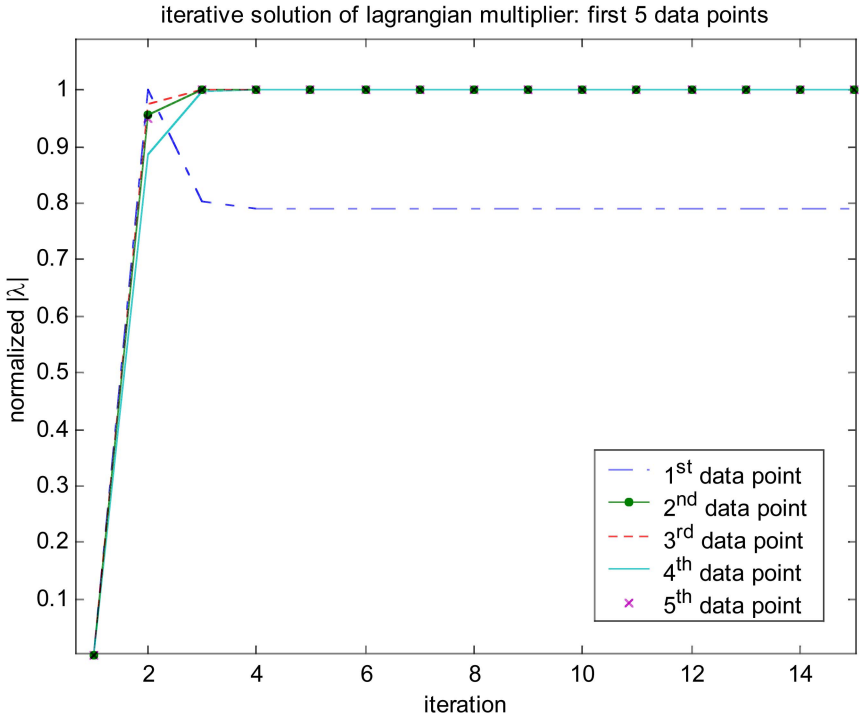


Fig. 8. Convergence in iterative Lagrangian multiplier.

TABLE I
RMS Estimation Errors

Estimators	RMS Estimation Error	
	Position (m)	Velocity (m/s)
Unconstrained	8.4	4.3
Best Linear Constrained	5.5	2.5
Nonlinear Constrained	1.8	0.4

updates and summarized in Table I. The performance improvement of the nonlinear constrained filter over the linearized constrained filter is demonstrated.

Finally for this simulation example, we use Fig. 8 to show an example of the Lagrangian multiplier as it is calculated iteratively using (19). The runs for five unconstrained state estimates are plotted in the same figure and to make it fit, the normalized absolute values of λ are taken. As shown, starting from zero, it typically takes 4 iterations for the algorithm to converge in the example presented.

4.2. Unconstrained Versus Constrained IMM Trackers

In the previous example, a nearly constant velocity model (27) was used in the filter. Obviously, a tracker that uses a maneuvering model can do better in tracking a turning target. However, it may still not be able to produce a track that falls on road all the time. The track-to-road fusion algorithm described in this paper can be applied in conjunction with a maneuvering target tracker to further improve target state estimation as illustrated in the following simulation example.

An IMM filter is constructed based on the "coordinated turn" models. For a ground vehicle, its wide turning maneuver is reasonably well modeled by a coordinated turn, i.e., at a constant turn rate with a constant speed. For the state vector $\mathbf{x}_k$ defined in (25), the coordinated turn model is given by

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \frac{\sin\omega T}{\omega} & 0 & -\frac{1-\cos\omega T}{\omega} \\ 0 & \cos\omega T & 0 & -\sin\omega T \\ 0 & \frac{1-\cos\omega T}{\omega} & 1 & \frac{\sin\omega T}{\omega} \\ 0 & \sin\omega T & 0 & \cos\omega T \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \frac{1}{2}T^2 & 0 \\ T & 0 \\ 0 & \frac{1}{2}T^2 \\ 0 & T \end{bmatrix} \mathbf{w}_k \tag{31}$$

where ω is the turn rate considered to be a known modeling parameter and $\mathbf{w}_k$ is defined as for (27).

For the IMM filter, three models are specified by choosing different values for ω. In the first model, setting $\omega = 0$ in (31) leads to the nearly constant velocity or non-maneuver model (27). In the second model, $\omega = 5.7$ deg/s represents a left turn maneuver while in the third model, $\omega = -5.7$ deg/s represents a right turn maneuver. The three models have an equal initial model probability of 1/3 and the model transition probability matrix is taken as

$$\mathbf{\Pi} = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}. \tag{32}$$

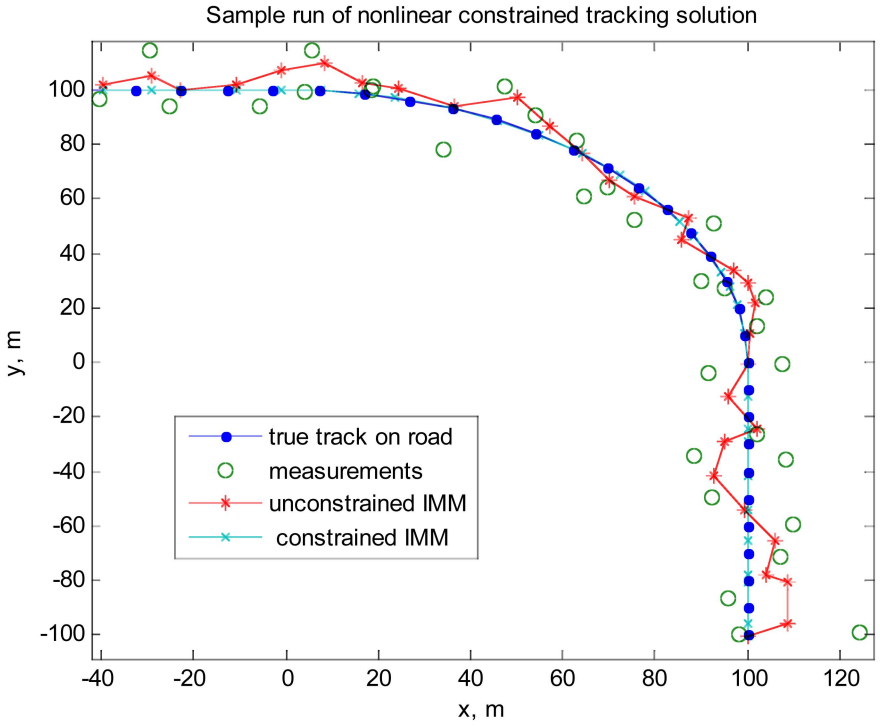


Fig. 9. Sample trajectories for unconstrained versus constrained IMM.

The three interacting filters inside the IMM tracker are identically initialized as $\hat{\mathbf{x}}_i \sim N(\mathbf{x}_0, \mathbf{P}_0)$ for $i = 1, 2, 3$. The same sensor model as (26) is used for generating radar measurements in this simulation. Hybrid constraints are applied with the nonlinear constraint (14) for position estimates and the linear constraint (30) for velocity estimates. In the Monte Carlo simulation, the truth track of the target remains the same but the initial estimate is drawn from the distribution for each run as described above. So is the measurement noise at each sampling instant for each run.

Fig. 9 shows sample trajectories wherein the target starts from an initial position at $(x_0, y_0) = (100 \text{ m}, -100 \text{ m})$ heading due north along a straight road with $\dot{y} = 10$ m/s. At $k = 10$ s, it follows the curve and makes a left turn at a rate of 5.7 deg/s for 16 s. At $k = 17$ s, it comes to another straight road heading due west with $\dot{x} = -10$ m/s for 5 s.

The true state is represented by a series of dots (·) plotted on the solid line of road segment. The corresponding measurements are a series of circles (o). The unconstrained IMM filter is shown as a series of connected stars (*) whereas the constrained IMM filter is shown as a series of connected crosses (×). When the target is on linear road segments, the linear constrained solution (5) is applied to the combined state of the IMM filter while on the curved road segment, a hybrid constrained solution is used (nonlinear for position and linear for velocity). From Fig. 9, the typical behavior of an unconstrained IMM filter can be seen. It converges rather quickly from the initialization of large errors, develops an overshoot right after the maneuver but corrects itself towards the true trajectory, and converges again after the maneuver terminated. However, these unconstrained IMM estimates (*) are off road while the target is on road.

In contrast, the constrained IMM estimates (×) are always on road even though they do not fall exactly on top of the true positions (·). As a result, the constrained position errors are smaller than the unconstrained ones as shown in Fig. 10, which are obtained by a Monte Carlo simulation with 100 runs. In particular, the velocity errors of the unconstrained IMM solution grow during the maneuver period whereas those of the constrained solution appear to be uniform.

The RMS errors averaged over the entire trajectory are summarized in Table II. The values in Table II are bigger than those in Table I because of larger initialization errors and longer simulation run. It shows an improvement of approximately 3 folds in position and in velocity.

5. CONCLUSIONS

In this paper, we presented an approach to incorporating road information into target tracking via track-to-road fusion. In this approach, road segments were modeled with analytic functions and their fusion with a target track was cast as a linearly or nonlinearly state constrained optimization procedure. With the Lagrangian multiplier, a closed-form solution was found for linear constraints and an iterative solution for nonlinear constraints. Geometric interpretations of the solutions were provided for simple cases. Computer simulation results demonstrate the performance of the algorithms.

Future work includes both algorithms development and practical applications. It is of interest to extend

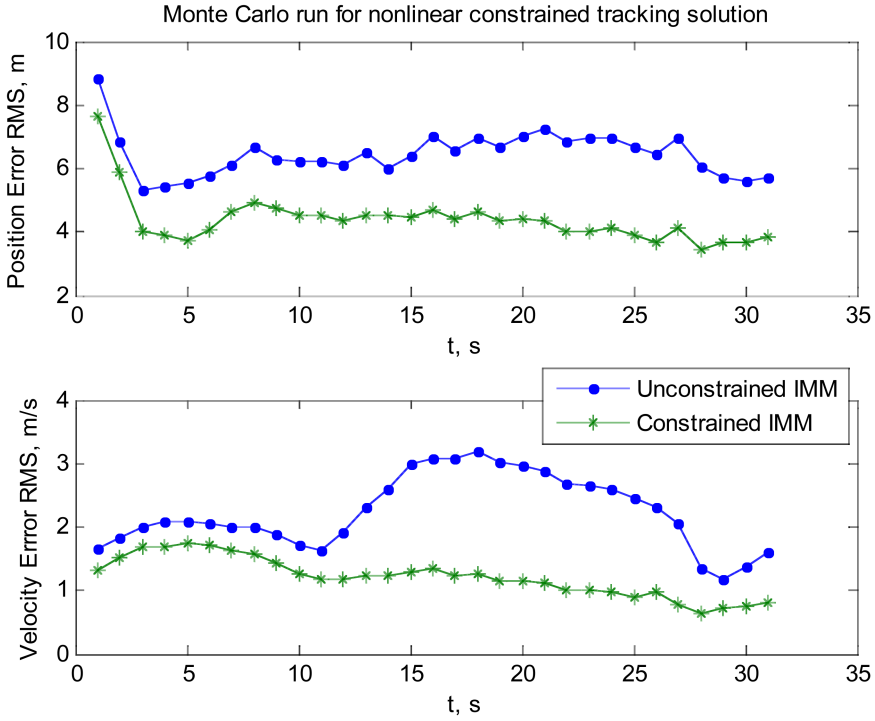


Fig. 10. Position and velocity error RMS versus time (100 Monte Carlo runs).

the iterative method presented in the paper for second-order nonlinear state constraints to other types of nonlinear constraints of practical significance and to search for more efficient root-finding algorithms to solve for the Lagrangian multiplier. Similarly, the simple fusion of a single track to a single road as presented in this paper is being extended to multiple targets moving along closely-spaced road networks with intersections and by-passes. In this case, the fusion (or constraining) can take place in the measurement level as well as in the track level, involving road constrained data association (RCDA). Results will be reported in future papers.

ACKNOWLEDGMENTS

Research supported in part under Contract No. FA8650-05-C-1808, which is gratefully acknowledged. Thanks also go to the reviewers and the editor for their careful reading, insightful comments, and constructive suggestions, which were incorporated into the text.

APPENDIX A

To solve the constrained optimization problem in (4), we form the cost function including the Lagrangian multiplier

$$J(\mathbf{x},\lambda) = (\mathbf{x}-\hat{\mathbf{x}})^T\mathbf{W}(\mathbf{x}-\hat{\mathbf{x}}) + 2\lambda^T(\mathbf{D}\mathbf{x}-\mathbf{d}). \quad \text{(A1)}$$

The first order conditions necessary for a minimum are given by

$$\frac{\partial J}{\partial \mathbf{x}} = 0 \Rightarrow \mathbf{W}(\mathbf{x}-\hat{\mathbf{x}}) + \mathbf{D}^T\lambda = 0 \quad \text{(A2a)}$$

TABLE II
RMS Estimation Errors

Estimators	RMS Estimation Error	
	Position (m)	Velocity (m/s)
Unconstrained IMM	9.0	6.2
Nonlinear Constrained IMM	3.2	1.7

$$\frac{\partial J}{\partial \lambda} = 0 \Rightarrow \mathbf{D}\mathbf{x}-\mathbf{d} = 0. \quad \text{(A2b)}$$

The solution for the optimal Lagrangian multiplier λ can be found first as

$$\lambda = (\mathbf{D}\mathbf{W}^{-1}\mathbf{D}^T)^{-1}(\mathbf{D}\hat{\mathbf{x}}-\mathbf{d}). \quad \text{(A3)}$$

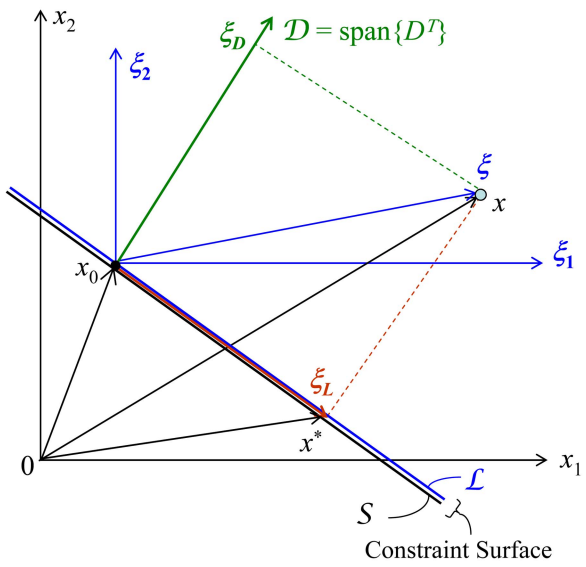
Bringing this solution back to (A1) leads to the constrained solution of the state in (5).

Note that the above derivation does not depend on the conditional Gaussian nature of the unconstrained estimate $\hat{\mathbf{x}}$. It was shown in [24] that when $\mathbf{W} = \mathbf{I}$, the solution in (5) is the same as what is obtained by the mean square method, which attempts to minimize the conditional mean square error subject to the state constraints, that is,

$$\min_{\mathbf{x}} E\{\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2 \mid Y\} \quad \text{such that} \quad \mathbf{D}\mathbf{x} = \mathbf{d}. \quad \text{(A4)}$$

Furthermore, when $\mathbf{W} = \mathbf{P}^{-1}$, i.e., the inverse of the unconstrained state estimation error covariance, the solution in (5) reduces to the result given by the maximum conditional probability method

$$\max_{\mathbf{x}} \ln \text{Prob}\{\mathbf{x} \mid Y\} \quad \text{such that} \quad \mathbf{D}\mathbf{x} = \mathbf{d}. \quad \text{(A5)}$$



$\{x_1, x_2\}$ ~ Basis for Original Space
$\{\xi_1, \xi_2\}$ ~ Basis for Translated Space
$S = \{x: Dx = d\}$ ~ Constraint Surface in Original Space
$\mathcal{L} = \{\xi: D\xi = 0\}$ ~ Constraint Surface in Translated Surface (Subspace)
$\mathcal{D} = \text{span}\{D^T\}$ ~ Subspace Orthogonal to Constraint Subspace
$\mathcal{D} \oplus \mathcal{L}$ = Translated Space
x_0 ~ Arbitrary Point on Constraint Surface in Original Space
x ~ Unconstrained State in Original Space
ξ ~ Unconstrained State in Translated Space
ξ_L ~ Constrained State in Translated Space
ξ_D ~ Projection onto $\mathcal{D}$
x^* ~ Constrained State in Original Space

Fig. B1. Geometrical interpretation of linear constrained solution.

More results and proofs can be found in [24].

APPENDIX B

For $\mathbf{x} \in \mathbb{R}^n$, the constraint surface $\mathcal{S} = \{\mathbf{x} : \mathbf{D}\mathbf{x} = \mathbf{d}\}$ with the number of linear constraints $m < n$ is not a subspace simply because for $\mathbf{d} \neq \mathbf{0}$, the null vector is not inside $\mathcal{S}$. To construct a subspace, first find an arbitrary point $\mathbf{x}_0 \in \mathcal{S}$ and then define $\boldsymbol{\xi} = \mathbf{x} - \mathbf{x}_0$. This is equivalent to shifting the origin of the coordinates to $\mathbf{x}_0$, thus performing an affine transformation, denoted by $\mathbf{T}$. For all $\mathbf{x} \in \mathcal{S}$, the corresponding shifted vector $\boldsymbol{\xi}$ has the following property:

$$\mathbf{D}\boldsymbol{\xi} = \mathbf{D}(\mathbf{x} - \mathbf{x}_0) = \mathbf{D}\mathbf{x} - \mathbf{D}\mathbf{x}_0 = \mathbf{d} - \mathbf{d} = 0. \quad \text{(B1)}$$

In other words, the constraint surface after the affine transformation $\mathbf{T}$ becomes a subspace, denoted by $\mathcal{L} = \mathbf{T}\mathcal{S} = \{\boldsymbol{\xi} : \mathbf{D}\boldsymbol{\xi} = \mathbf{0}\}$, which has a dimension $n - m$. The affine transformation is illustrated in Fig. B1.

We are now to express $\mathcal{L}$. But first, the row vectors of $\mathbf{D}$ can be expressed as:

$$\mathbf{D}^T = [\mathbf{d}_1 \ \ \mathbf{d}_2 \cdots \mathbf{d}_m]. \quad \text{(B2)}$$

Since $\mathbf{D}$ is of full rank by assumption, the row vectors of $\mathbf{D}$ can be used as the non-orthogonal basis for a subspace denoted by $\mathcal{D} = \text{span}\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m\}$. In light of (B1) and by definition of $\mathcal{L}$, it is easy to see that $\mathcal{D}$ is an orthogonal complement of $\mathcal{L}$, that is, $\mathcal{D} \perp \mathcal{L}$ and $\mathcal{D} \oplus \mathcal{L} = \mathbb{R}^n$ where $\oplus$ stands for direct sum between two orthogonal subspaces.

For $\boldsymbol{\delta} \in \mathcal{D}$, it can be written as:

$$\boldsymbol{\delta} = \sum_{i=1}^{m} c_i \mathbf{d}_i = [\mathbf{d}_1 \ \ \mathbf{d}_2 \cdots \mathbf{d}_m] \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \mathbf{D}^T\mathbf{c}. \quad \text{(B3)}$$

Then for $\boldsymbol{\xi} \in \mathcal{L}$, we have

$$\langle \boldsymbol{\delta}, \boldsymbol{\xi} \rangle = \langle \mathbf{D}^T\mathbf{c}, \boldsymbol{\xi} \rangle = \boldsymbol{\delta}^T\boldsymbol{\xi} = \mathbf{c}^T\mathbf{D}\boldsymbol{\xi} = 0 \quad \text{(B4)}$$

where $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T\mathbf{b}$ is the inner product defined on $\mathbb{R}^n$.

By the principle of orthogonality, an arbitrary vector $\boldsymbol{\xi}$ can be decomposed into its projections onto the orthogonal complements $\mathcal{D}$ and $\mathcal{L}$, denoted by $\boldsymbol{\xi}_D$ and $\boldsymbol{\xi}_L$, respectively, as

$$\boldsymbol{\xi} = \boldsymbol{\xi}_D + \boldsymbol{\xi}_L. \quad \text{(B5)}$$

Adding $\mathbf{x}_0$ to both sides of (B5), we can express the vectors in the original coordinates as:

$$\mathbf{x} = \boldsymbol{\xi} + \mathbf{x}_0 = \boldsymbol{\xi}_D + \boldsymbol{\xi}_L + \mathbf{x}_0 = \boldsymbol{\xi}_D + \mathbf{x}^*. \quad \text{(B6)}$$

The projection of the arbitrary vector on the constraint subspace $\mathcal{L}$ and the constraint surface $\mathcal{S}$ can be obtained, respectively, as:

$$\boldsymbol{\xi}_L = \boldsymbol{\xi} - \boldsymbol{\xi}_D \quad \text{(B7a)}$$

$$\mathbf{x}^* = \mathbf{x} - \boldsymbol{\xi}_D. \quad \text{(B7b)}$$

To obtain $\boldsymbol{\xi}_D$, express it as a linear combination of the non-orthogonal bases of $\mathbf{D}^T$ with the coefficient vector $\mathbf{c}$ as:

$$\boldsymbol{\xi}_D = \sum_{i=1}^{m} c_i \mathbf{d}_i = [\mathbf{d}_1 \ \ \mathbf{d}_2 \cdots \mathbf{d}_m] \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \mathbf{D}^T\mathbf{c}. \quad \text{(B8)}$$

Again, by the principle of orthogonality, the projection error vector $\boldsymbol{\xi} - \boldsymbol{\xi}_D$ is orthogonal to $\mathcal{D}$, i.e., each and every basis of it:

$$\langle \boldsymbol{\xi} - \boldsymbol{\xi}_D, \mathbf{d}_i \rangle = \langle \boldsymbol{\xi} - \mathbf{D}^T\mathbf{c}, \mathbf{d}_i \rangle = \mathbf{d}_i^T(\boldsymbol{\xi} - \mathbf{D}^T\mathbf{c}) = 0,$$
$$i = 1, \ldots, m. \quad \text{(B9)}$$

Stacking these orthogonality conditions, we obtain

$$\begin{bmatrix} \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \vdots \\ \mathbf{d}_m^T \end{bmatrix} (\boldsymbol{\xi} - \mathbf{D}^T\mathbf{c}) = 0 \qquad \text{or} \qquad \mathbf{D}(\boldsymbol{\xi} - \mathbf{D}^T\mathbf{c}) = 0. \tag{B10}$$

Since $\mathbf{D}\mathbf{D}^T$ is an $m \times m$ matrix and invertible, the coefficient vector can be obtained as:

$$\mathbf{c} = (\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D}\boldsymbol{\xi}. \tag{B11}$$

Bringing (B11) back to (B8) gives the projection vector as:

$$\boldsymbol{\xi}_D = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D}\boldsymbol{\xi} = \mathbf{P}\boldsymbol{\xi} \tag{B12}$$

where $\mathbf{P} = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D}$ is usually referred to as the projection matrix onto $\mathcal{D}$ and $(\mathbf{I} - \mathbf{P})$ is the projection matrix onto $\mathcal{L}$.

Bringing (B12) back to (B7) gives

$$\boldsymbol{\xi}_L = \boldsymbol{\xi} - \mathbf{P}\boldsymbol{\xi} = (\mathbf{I} - \mathbf{P})\boldsymbol{\xi} \tag{B13a}$$

$$\mathbf{x}^* = \mathbf{x} - \mathbf{P}\boldsymbol{\xi} = \mathbf{x} - \mathbf{P}(\mathbf{x} - \mathbf{x}_0). \tag{B13b}$$

Bringing the expression for $\mathbf{P}$ into (B13b) gives

$$\begin{aligned} \mathbf{x}^* &= \mathbf{x} - \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D}(\mathbf{x} - \mathbf{x}_0) \\ &= \mathbf{x} - \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}(\mathbf{D}\mathbf{x} - \mathbf{D}\mathbf{x}_0) \\ &= \mathbf{x} - \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}(\mathbf{D}\mathbf{x} - \mathbf{d}) \end{aligned} \tag{B14}$$

where $\mathbf{D}\mathbf{x}_0 = \mathbf{d}$ is used to arrive at the last equation because of $\mathbf{x}_0 \in \mathcal{S}$.

Clearly, (B14) is exactly the same as (5) when $\mathbf{W} = \mathbf{I}$. This offers a geometric interpretation that the linear constrained estimation is the orthogonal projection of the unconstrained estimate onto the constrained surface. It provides a theoretical justification of the intuitive practice of finding a point along the road that is of the shortest distance.

APPENDIX C

When $\mathbf{W} \neq \mathbf{I}$, we can rewrite the weighted square error formulation as

$$\begin{aligned} \breve{\mathbf{x}} &= \arg\min_{\mathbf{x}\in S}(\mathbf{x} - \hat{\mathbf{x}})^T\mathbf{W}(\mathbf{x} - \hat{\mathbf{x}}) \\ &= \arg\min_{\mathbf{x}\in S}[\mathbf{W}^{1/2}(\mathbf{x} - \hat{\mathbf{x}})]^T\mathbf{W}^{1/2}(\mathbf{x} - \hat{\mathbf{x}}) = \arg\min_{\mathbf{z}\in\bar{S}}\mathbf{z}^T\mathbf{z} \end{aligned} \tag{C1}$$

where $\mathbf{W} = \mathbf{W}^{1/2}\mathbf{W}^{1/2}$ is a symmetric positive definite weighting matrix. This can be understood as if we perform an equivalent un-weighted optimization on the transformed state:

$$\mathbf{z} = \mathbf{W}^{1/2}\mathbf{x}. \tag{C2}$$

The constraint can be written as:

$$\mathbf{D}\mathbf{x} = \mathbf{D}\mathbf{W}^{-1/2}\mathbf{W}^{1/2}\mathbf{x} = \mathbf{M}\mathbf{z} = \mathbf{d} \tag{C3}$$

where $\mathbf{M} = \mathbf{D}\mathbf{W}^{-1/2}$ by definition. The constrained surface $\bar{S} = \{\mathbf{z} : \mathbf{M}\mathbf{z} = \mathbf{d}\}$ is used in the last equality of (C1).

Since the constrained solution in (B14) holds for $\mathbf{z}$ with $\mathbf{M}$ and $\mathbf{d}$, we have

$$\mathbf{z}^* = \mathbf{z} - \mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^{-1}(\mathbf{M}\mathbf{z} - \mathbf{d}). \tag{C4}$$

Putting (C2) and (C3) into (C4) gives

$$\begin{aligned} \mathbf{W}^{1/2}\mathbf{x}^* = \mathbf{W}^{1/2}\mathbf{x} &- \mathbf{W}^{-1/2}\mathbf{D}^T(\mathbf{D}\mathbf{W}^{-1/2}\mathbf{W}^{-1/2}\mathbf{D}^T)^{-1} \\ &\times (\mathbf{D}\mathbf{W}^{-1/2}\mathbf{W}^{1/2}\mathbf{x} - \mathbf{d}). \end{aligned} \tag{C5}$$

Multiplying both sides by $\mathbf{W}^{-1/2}$ gives the weighted constrained solution as:

$$\mathbf{x}^* = \mathbf{x} - \mathbf{W}^{-1}\mathbf{D}^T(\mathbf{D}\mathbf{W}^{-1}\mathbf{D}^T)^{-1}(\mathbf{D}\mathbf{x} - \mathbf{d}) \tag{C6}$$

which is exactly the same as (5).

It is interesting to note that the use of $\mathbf{W} = \mathbf{P}^{-1}$ has the effect of pre-whitening in the sense that

$$\mathrm{E}\{\mathbf{z}\mathbf{z}^T\} = \mathbf{P}^{-1/2}\mathrm{E}\{\mathbf{x}\mathbf{x}^T\}\mathbf{P}^{-1/2} = \mathbf{P}^{-1/2}\mathbf{P}\mathbf{P}^{-1/2} = \mathbf{I}. \tag{C7}$$

APPENDIX D

Construct the Lagrangian with the Lagrangian multiplier λ as

$$J(\mathbf{x}, \lambda) = (\mathbf{z} - \mathbf{H}\mathbf{x})^T(\mathbf{z} - \mathbf{H}\mathbf{x}) + \lambda f(\mathbf{x}). \tag{D1}$$

Taking the partial derivatives of $J(\mathbf{x}, \lambda)$ with respect to $\mathbf{x}$ and λ, respectively, setting them to zero leads to the necessary conditions

$$-\mathbf{H}^T\mathbf{z} + \lambda\mathbf{m} + (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{M})\mathbf{x} = \mathbf{0} \tag{D2a}$$

$$\mathbf{x}^T\mathbf{M}\mathbf{x} + \mathbf{m}^T\mathbf{x} + \mathbf{x}^T\mathbf{m} + m_0 = 0. \tag{D2b}$$

Assume that the inverse matrix of $\mathbf{H}^T\mathbf{H} + \lambda\mathbf{M}$ exists. Then, $\mathbf{x}$ can be solved from (D2a), giving the constrained solution in terms of the unknown λ as

$$\mathbf{x} = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{M})^{-1}(\mathbf{H}^T\mathbf{z} - \lambda\mathbf{m}) \tag{D3}$$

which reduces to the unconstrained least-squares solution when $\lambda = 0$.

Assume that the matrix $\mathbf{M}$ admits the factorization $\mathbf{M} = \mathbf{L}^T\mathbf{L}$ and apply the Cholesky factorization to $\mathbf{W} = \mathbf{H}^T\mathbf{H}$ as

$$\mathbf{W} = \mathbf{G}^T\mathbf{G} \tag{D4}$$

where $\mathbf{G}$ is an upper right diagonal matrix. We then perform the SVD [17] of the matrix $\mathbf{L}\mathbf{G}^{-1}$ as

$$\mathbf{L}\mathbf{G}^{-1} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \tag{D5}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices and $\boldsymbol{\Sigma}$ is a diagonal matrix with its diagonal elements denoted by σ_i. For a square matrix, this becomes the eigenvalue decomposition.

Introduce two new vectors

$$\mathbf{e}(\lambda) = [\ldots e_i(\lambda), \ldots]^T = \mathbf{V}^T(\mathbf{G}^T)^{-1}(\mathbf{H}^T\mathbf{z} - \lambda\mathbf{m}) \tag{D6a}$$

$$\mathbf{t} = [\ldots t_i \ldots]^T = \mathbf{V}^T(\mathbf{G}^T)^{-1}\mathbf{m}. \tag{D6b}$$

With these factorizations and new matrix and vector notations, the constrained solution in (D3) can be simplified into (16a), which is repeated below for easy reference as

$$\mathbf{x} = \mathbf{G}^{-1}\mathbf{V}(\mathbf{I} + \lambda\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^{-1}\mathbf{e}(\lambda). \tag{D7}$$

The first and second order terms of $\mathbf{x}$ in (D2b) can be expressed in λ as:

$$\mathbf{x}^T\mathbf{M}\mathbf{x} = \mathbf{e}(\lambda)^T(\mathbf{I} + \lambda\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^{-T}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}(\mathbf{I} + \lambda\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^{-1}\mathbf{e}(\lambda) = \sum_i \frac{e_i^2(\lambda)\sigma_i^2}{(1 + \lambda\sigma_i^2)^2} \tag{D8a}$$

$$\mathbf{m}^T\mathbf{x} = \mathbf{t}^T(\mathbf{I} + \lambda\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^{-1}\mathbf{e}(\lambda) = \sum_i \frac{e_i(\lambda)t_i}{1 + \lambda\sigma_i^2} \tag{D8b}$$

$$\mathbf{x}^T\mathbf{m} = \mathbf{e}(\lambda)^T(\mathbf{I} + \lambda\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^{-1}\mathbf{t} = \sum_i \frac{e_i(\lambda)t_i}{1 + \lambda\sigma_i^2}. \tag{D8c}$$

Bringing these terms into the constrained equation in (D2b) gives rise to the constraint equation, now expressed in terms of the unknown Lagrangian multiplier λ, as

$$\begin{aligned} q(\lambda) &= (\mathbf{z}^T\mathbf{H} - \lambda\mathbf{m}^T)(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{M})^{-2}(\mathbf{H}^T\mathbf{z} - \lambda\mathbf{m}) \\ &\quad + \mathbf{m}^T(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{M})^{-1}(\mathbf{H}^T\mathbf{z} - \lambda\mathbf{m}) \\ &\quad + (\mathbf{z}^T\mathbf{H} - \lambda\mathbf{m}^T)(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{M})^{-1}\mathbf{m} + m_0 \\ &= \mathbf{e}(\lambda)^T(\mathbf{I} + \lambda\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}(\mathbf{I} + \lambda\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^{-1}\mathbf{e}(\lambda) \\ &\quad + \mathbf{t}^T(\mathbf{I} + \lambda\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^{-1}\mathbf{e}(\lambda) + \mathbf{e}(\lambda)^T(\mathbf{I} + \lambda\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})^{-1}\mathbf{t} + m_0 \\ &= \sum_i \frac{e_i^2(\lambda)\sigma_i^2}{(1 + \lambda\sigma_i^2)^2} + 2\sum_i \frac{e_i(\lambda)t_j}{1 + \lambda\sigma_i^2} + m_0 = 0. \end{aligned} \tag{D9}$$

which is (16b) given in Section 3.

REFERENCES

[1] Y. I. Abromovich and M. B. Sverdlik
Synthesis of a filter response which maximizes the signal to noise ratio under additional quadratic constraints.
Radio Eng. and Electron. Phys., **5** (Nov. 1970), 1977–1984.

[2] C. Agate and K. J. Sullivan
Road-constraint target tracking and identification using a particle filter.
In O. E. Drummond (Ed.), *SPIE Proceedings of Signal and Data Processing of Small Targets*, vol. 5204, 2003, 532–543.

[3] B. Anderson and J. Moore
Optimal Filtering.
Englewood Cliffs, NJ: Prentice-Hall, 1979.

[4] S. Arulampalam, N. Gordon, M. Orton and B. Ristic
A variable structure multiple model particle filter for GMTI tracking.
In *Proceedings of the 5th International Conference on Information Fusion*, Annapolis, MD, July 2002, 927–934.

[5] Y. Bar-Shalom and X. R. Li
Multitarget-Multisensor Tracking: Principles and Techniques.
Storrs, CT: YBS Publishing, 1995.

[6] S. Blackman and R. Popoli
Design and Analysis of Modern Tracking Systems.
Boston, MA: Artrch House, 1999.

[7] T. Connare and E. Blasch
Group IMM tracking utilizing track and identification fusion.
In *Proceedings of the Workshop on Estimation, Tracking, and Fusion; A Tribute to Yaakov Bar Shalom*, Monterey, CA, May 2001, 205–220.

[8] A. Farina, L. Ferranti and G. Golino
Constrained tracking filters for A-SMGCS.
In *Proceedings of the 6th International Conference on Information Fusion*, Cairns, Queensland, Australia, July 2003, 414–421.

[9] G. C. Goodwin, J. A. De Dona, M. M. Seron and X. W. Zhuo
Lagrangian Duality between constrained estimation and control.
Automatica, **41** (June 2005), 935–944.

[10] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati and I. Kadar
Ground target tracking with variable structure IMM estimator.
IEEE Transactions on Aerospace and Electronic Systems, **36** (Jan. 2002), 26–46.

[11] S. Ko and R. Bitmead
State estimation for linear systems with state equality constraints.
Automatica, **43**, 8 (Aug. 2007), 1363–1368.

[12] K. Koch
Ground target tracking with STAP-Radar: Selected tracking aspects.
In R. Klemm (Ed.), *Applications of Space-Time Adaptive Processing*, London: Institute of Electrical Engineers, 2004.

[13] M. Ulmke and W. Koch
Multi hypothesis track extraction and maintenance of GMTI sensor data.
In *Proceedings of the 8th International Conference on Information Fusion*, Philadelphia, PA, July 2005.

[14] R. G Lorenz and S. P. Boyd
Robust minimum variance beamforming.
In J. Li and P. Stoica (Eds.), *Robust Adaptive Beamforming*, Hoboken, NJ: Wiley-Interscience, 2006.

[15] D. G. Luenberger
Linear and Nonlinear Programming (2nd ed.).
Reading, MA: Addison-Wesley, 1989.

[16] M. Miller, E. Blasch, T. Nguyen and C. Yang
Constrained estimation for GPS/digital map integration.
In *Proceedings of the National Technical Meeting of the Institute of Navigation*, San Diego, CA, Jan. 2007, 1119–1127.

[17] T. K. Moon and W. C. Stirling
Mathematical Methods and Algorithms for Signal Processing.
Upper Saddle River, NJ: Prentice-Hall, 2000.

[18] B. J. Noe and N. Collins
Variable structure interacting multiple model filter (VS-IMM) for tracking targets with transportation network constraints.
In O. E. Drummond (Ed.), *SPIE Proceedings of Signal and Data Processing of Small Targets*, vol. 4048, 2000, 247–257.

[19] B. Pannetier, K. Benameur, V. Nimier and M. Rombau
VS-IMM using road map information for a ground target tracking.
In *Proceedings of the 8th International Conference on Information Fusion*, Philadelphia, PA, July 2005, A1-2 (1–8).

[20] C. V. Rao, J. B. Rawlings and D. Q. Mayne
Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations.
IEEE Transactions on Automatic Control, **48** (Feb. 2003), 246–258.

[21] B. Ristic, S. Arullampalam and N. Gordon
Beyond the Kalman Filter—Particle Filters for Tracking Applications.
Boston, MA: Artech House, 2004.

[22] P. J. Shea, T. Zadra, D. Klamer, E. Frangione and R. Brouillard
Improved state estimation through use of roads in ground tracking.
In O. E. Drummond (Ed.), *SPIE Proceedings of Signal and Data Processing of Small Targets*, vol. 4048, 2000, 321–332.

[23] D. Simon
A game theory approach to constrained minimax state estimation.
IEEE Transactions on Signal Processing, **54** (Feb. 2006), 405–412.

[24] D. Simon and T. L. Chia
Kalman filtering with state equality constraints.
IEEE Transactions on Aerospace and Electronic Systems, **38** (Jan. 2002), 128–136.

[25] M. Ulmke
Improved GMTI-tracking using road maps and topographic information.
In O. E. Drummond (Ed.), *SPIE Proceedings of Signal and Data Processing of Small Targets*, vol. 5204, 2003.

[26] M. Ulmke and W. Koch
Road-map assisted ground moving target tracking.
IEEE Transactions on Aerospace and Electronic Systems, **42**, 4 (Oct. 2006), 1264–1274.

[27] C. Yang, M. Bakich and E. Blasch
Nonlinear constrained tracking of targets on roads.
In *Proceedings of the 8th International Conf. on Information Fusion*, Philadelphia, PA, July 2005, A8-3 (1–8).

[28] C. Yang and E. Blasch
Kalman filtering with nonlinear state constraints.
In *Proceedings of the 9th International Conference on Information Fusion*, Florence, Italy, July 2006, 8–15.

[29] C. Yang and D. M. Lin
Constrained optimization for joint estimation of channel biases and angles of arrival for small GPS antenna arrays.
In *Proceedings of the 60th Annual Meeting of the Institute of Navigation*, Dayton, OH, June 2004, 548–559.

Chun Yang received his Bachelor of Engineering from Northeastern University, Shenyang, China, in 1984, majored in industrial automation. In 1986, he received his Diplôme d'Études Approfondies from Institut National Polytechnique de Grenoble, Grenoble, France, majored in automatique et traitement du signal (LAG/ENSIEG). In 1989, he received his title of Docteur en Science from Université de Paris, Orsay, France, in sciences physiques (LSS/CNRS-ESE).

After two years of postdoctoral research at University of Connecticut, Storrs, CT, he moved on with his industrial R&D career. Since 1994, he has been with Sigtem Technology, Inc. and has been working on adaptive array and baseband signal processing for GNSS receivers and radar systems as well as on nonlinear state estimation with applications in target tracking, integrated inertial navigation, and information fusion. He served as tutorial chair for the ISIF/IEEE Fusion Conferences in Annapolis, MD (2000) and in Philadelphia, PA (2005), respectively. Dr. Yang is also an Adjunct Professor of Electrical and Computer Engineering at Miami University. He is the co-inventor of seven issued and pending U.S. patents, the co-author of an ION-GNSS best presentation paper (2007), the coauthor of an IEEE PLANS/ION AM best paper (2006), and the co-recipient of the ION Samuel M. Burka Award (2007).

Erik Blasch received his B.S. in mechanical engineering from the Massachusetts Institute of Technology and Masters Degrees in mechanical, health science, and industrial engineering from Georgia Tech. He attended the University of Wisconsin for an M.D./Ph.D. in mechanical engineering/neurosciences until being called to active duty in the United States Air Force. He completed an M.B.A., M.S.E.E., M.S. econ, M.S./Ph.D. psychology (ABD), and a Ph.D. in electrical engineering from Wright State University.

He is an information fusion evaluation tech lead for the Air Force Research Laboratory—COMprehensive Performance Assessment of Sensor Exploitation (COMPASE) Center (AFRL/RYAA), adjunct EE and BME professor at Wright State University (WSU) and Air Force Institute of Technology (AFIT), and a reserve Major with the Air Force Office of Scientific Research (AFRL/AFOSR). He was a founding member of the International Society of Information Fusion (ISIF) in 1998 and the 2007 ISIF president.

Dr. Blasch has many military and civilian career awards; but engineering highlights include team member of the winning 1991 American Tour del Sol solar car competition, 1994 AIAA mobile robotics contest, and the 1993 Aerial Unmanned Vehicle competition where they were first in the world to automatically control a helicopter. Since that time, Dr. Blasch has focused on automatic target recognition, targeting tracking, and information fusion research compiling 200+ scientific papers and book chapters. He is active in IEEE (AES and SMC) and SPIE including regional activities, conference boards, journal reviews, and scholarship committees; and participates in the Data Fusion Information Group (DFIG), the program committee for the DOD Automatic Target Recognition Working Group (ATRWG), and the SENSIAC National Symposium on Sensor and Data Fusion (NSSDF). He is a Fellow of SPIE.