Target Track Initiation in Difficult Scenarios Using Probability-1 Homotopy Methods and Cubature Integration

David Frederic Crouse
Naval Research Laboratory
4555 Overlook Ave., SW
Washington DC 20375
E-Mail: david.crouse@nrl.navy.mil

Abstract—Probability-1 homotopy algorithms are seldom taught in standard numerical analysis classes, and, with a few exceptions, their utility in target track initiation has been largely overlooked. This paper demonstrates how such methods can be used for unbiased measurement conversion and initial target-state estimation in difficult scenarios involving non-simultaneous measurements when combined with cubature integration techniques. Scenarios considered are range-only and time-delay-of-arrival-only measurements as well as surface-wave time-delay-of-arrival-only measurements.

TABLE OF CONTENTS
1. INTRODUCTION ........................................... 1
2. CUBATURE INTEGRATION ............................... 2
3. PROBABILITY-1 HOMOTOPY METHODS ............ 3
4. BISTATIC RANGE-ONLY TRACK INITIATION ... 8
5. BISTATIC TIME-DELAY-OF-ARRIVAL-ONLY TRACK INITIATION ........................................ 13
6. BISTATIC TDOA-ONLY SURFACE-WAVE TARGET LOCALIZATION .......................... 14
7. RECOGNIZING MORE DIFFICULT PROBLEMS .... 15
8. CONCLUSIONS ............................................ 15
APPENDICES .................................................. 17
A. FORMULAE FOR THE CUBATURE POINTS ...... 17
B. SIMPLE NUMERICAL DIFFERENTIATION ....... 18
REFERENCES ................................................ 18

1. INTRODUCTION
The majority of work on target-tracking algorithms utilizes multivariate Gaussian approximations for the target state and focuses on how measurements can be assigned to targets after a track has been started on each target. However, to start a track on a target, one must determine that a set of measurements over time belongs to a common target, and one must transform the measurements into an initial target-state estimate, which usually consists of a mean and a covariance matrix. This paper considers how a set of measurements can be transformed into complete Cartesian locations, one-point or two-point differencing can be used to get an initial target-state estimate as described in [2, Ch. 5.5] and [50]. “One-point differencing” means that one sets the position components of the target state to the value of the measurement, the covariance associated with those components is the covariance of the measurement, and the velocity components of the state are set to zero with the covariance of the state set to some bounds associated with the maximum allowable velocity of the target. “Two-point differencing” means that the two measurements are used to algebraically solve for the velocity under a linear dynamic model.

Probability-1 homotopy methods have most likely been overlooked in the tracking literature due to the complexity of implementing even the simplest one. As many readers might also not want to go to the trouble of implementing the probability-1 homotopy method described in this paper, an implementation has been made available for download as part of the Tracker Component Library, which is currently at https://github.com/DavidFCrouse/Tracker-Component-Library. The relevant function is called homotopySolver, is in the "Mathematical Functions" folder, and requires the Runge-Kutta routines that are also part of the library.

Typically, if a target state consists of position and velocity components and measurements can be transformed into complete Cartesian locations, one-point or two-point differencing can be used to get an initial target-state estimate as described in [2, Ch. 5.5] and [50]. “One-point differencing” means that one sets the position components of the target state to the value of the measurement, the covariance associated with those components is the covariance of the measurement, and the velocity components of the state are set to zero with the covariance of the state set to some bounds associated with the maximum allowable velocity of the target. “Two-point differencing” means that the two measurements are used to algebraically solve for the velocity under a linear dynamic model.

More sophisticated methods of starting a target track are discussed along with tracking algorithms in the tutorial series [14–16]. For example, in [14] an information filter is used to start tracks instead of two-point differencing. Such an algorithm allows process noise to be incorporated into the covariance estimate of the initial state. In [15], track initiation in the presence of ray-traceable atmospheric refraction is demonstrated through the use of cubature integration methods to obtain unbiased Cartesian-converted measurements and statistically consistent covariance matrices. Additionally, in [16] track initiation on a non-linear dynamic model is demon-
strated, where Doppler-measurements are used to refine an 
initial estimate obtained without such measurements using a 
maximum-likelihood estimation method.

In all of the aforementioned scenarios, the measurements 
available to start the track could be converted from the mea-
surement domain into complete 3D position measurements. 
The problem of track initiation given incomplete measure-
ments, on the other hand, is more difficult. Incomplete-
measurement track initiation is a necessary part of many pas-
sive tracking algorithms. For example, in [23], the problem of 
track initiation of aircraft using bistatic range-only measure-
ments is considered. In the full 3D case including no prior 
information on the target altitude, an initial position estimate 
was obtained using three simultaneous bistatic measurements 
by finding a function that algebraically transforms three 
ranges into a Cartesian position; the covariance associated 
with that estimate was obtained using a linearization.\(^2\)

In [4], two general strategies for track initialization using 
incomplete measurements based on maximum \(a \text{ posteriori}\) 
estimation are provided. One method is based on lineariz-
ation, similar to what was done in [23], and the other is 
based on the use of the “unscented transformation,” which 
is a second-order cubature integration technique. Unlike in 
[4], the focus of this article is on directly obtaining the first 
two moments of an estimate given a sufficient number of 
measurements, and not on obtaining the maximum \(a \text{ poste-
riori}\) estimate. Moreover, this article assumes that no prior 
information is available (beyond just assuming the target is 
within the viewing region of the sensor).

The method of track initiation chosen for this paper is a 
generalization of the method used in [15] for initiation in a 
refraction-corrupted environment. It consists of the following 
steps:

1. Find a function to convert from the measurement domain 
to the target-state domain. For example, a function to convert 
bistatic range-only measurements into a Cartesian position or 
position and velocity.
2. Assume that available measurements are corrupted with 
Gaussian noise having known covariance matrices and cal-
culate the first two moments of the target state given the 
measurements using cubature integration.

During the track-initiation phase, the effects of process noise in 
the dynamic model are neglected. Unlike the method in 
[15], the examples in this paper do not all just convert the 
measurements into Cartesian measurements and then pass 
them to an information filter. Rather, in two instances, the 
entire state is estimated at once given the measurements. The 
use of cubature integration to perform this task is described 
in Section 2.

This leaves the problem of finding a function that converts 
a set of measurements into the target-state domain. It shall 
not always be assumed that multiple sensors simultaneously 
observe the target so that one can directly obtain Cartesian 
positions (assuming that the target state is Cartesian). Addi-
tionally, it is desirable to obtain a solution that will work with 
general dynamic models. For example, when using multiple 
model estimators, such as those described in [2, Ch. 11.6], 
initial estimates are needed for all possible dynamic models.

In many instances given nonlinear target dynamics, a function 
converting from the measurement domain into the target-state 
domain can be explicitly found. For example, given two 
position measurements, one can get a velocity measurement 
assuming a Keplerian orbital dynamic model (standard two-
point differencing would be inaccurate) by solving Lambert’s 
problem as in [33, 34]. Similarly, assuming a Keplerian 
dynamic model, three directional observations can be trans-
formed into a full target-state vector as in [35, 36], though 
multiple solutions might exist. However, the focus of this 
paper is on instances where finding an explicit solution is 
either very difficult or is too time consuming.

In high-precision models such as described in [19], where 
ray-tracing might be used to account for complicated atmo-
spheric or acoustic propagation and a wide variety of 
effects might come into play in the dynamic model or in 
establishing a high-fidelity coordinate system. It would take 
an unreasonable amount of time to derive an explicit conver-
sion from the measurement domain into the domain of the 
target state. Thus, as described in Section 3, a probability-1 
homotopy method is used in this article. The simplest type 
of probability-1 homotopy methods are generally designed 
to solve the problem \(f(x) = 0\) or \(f(x) = x\). They are 
“probability-1” because, given a random initialization, they 
will converge to a solution in a finite number of iterations 
with probability 1.\(^3\) Such techniques can be applied to a wide 
variety of problems, and good implementations can be quite 
fast for many types of problems.

Measurement conversion and/or track initiation discussed in 
this paper is demonstrated through three challenging scenar-
ios:

1. Section 4: Measurement conversion using simultaneous 
bistatic range measurements and track initiation using non-
simultaneous bistatic range measurements assuming a linear 
dynamic model.
2. Section 5: Track initiation using non-simultaneous time-
delay-of-arrival of measurements assuming a linear dynamic 
model.
3. Section 6: Measurement conversion using simultaneous 
bistatic surface-wave TDOA measurements.

Another practical application of probability-1 homotopy 
methods in tracking is in the bistatic range-only orbital track-
initiation problem, which is addressed in [38, 42, 60] and is 
not further considered in this paper. Though probability-1 
homotopy methods are very powerful, Section 7 notes prob-
lems that can arise due to to a poor formulation of a problem 
and/or finite-precision errors. The results are concluded in 
Section 8.

\section{2. Cubature Integration}

Cubature integration is a technique of exactly or approxi-
mately evaluating multivariate integrals in which a weighting 
function, typically a Gaussian probability density function (PDF), 
multiplies another function. For a general multi-

\(^2\)In [23], the algebraic method for transforming from the measurement to 
the state domain was not provided. Rather it was solved in Mathematica 
and the expression was too long to publish. It is thus worth noting that in 
[49], an explicit expression (the spherical-intersection method) is provided 
for the general range-only target-position estimation problem along with 
a linearization for a covariance matrix.

\(^3\)There can exist a few degenerate initial estimates or problem solutions 
where the algorithm can fail. However, in some instances, these are easy 
to guess. For example, setting the initial estimate to be the location of the 
sensor is generally bad.
mensional integral, one can write
\[ \int_{\text{Re}^{d_x}} w(x)g(x) \, dx = \sum_{i=0}^{N_x} \omega_i g(\xi_i) \] (1)
where \( w \) is a scalar weighting function that takes the \( d_x \)-dimensional vector parameter \( x \), \( g \) is a vector or scalar function, and \( \omega_i \) are cubature weights and \( \xi_i \) are \( d_x \)-dimensional cubature points. The cubature points and weights are designed for a particular weighting function so that the equality in (1) is true for all multivariate (vector) polynomials \( g \) up to a certain degree. For non-polynomial or higher-order functions, the sum on the right-hand side of (1) can be taken to be an approximation.

Cubature points and weights for many choices of weighting functions up to many degrees and orders are available. Many cubature formulae are given in the (out-of-print) monograph [62] and are listed on the website http://nines.cs.kuleuven.be/ecf, which is documented in [12]. Cubature integration forms the basis of the cubature Kalman filter as discussed in [14]. Cubature integration can be easily used to approximate expected values and covariance matrices of nonlinear transformations of multivariate Gaussian random variables. That is,
\[ \hat{x} \triangleq E \{ g(x) \} \approx \sum_{i=0}^{N_x} \omega_i g(\xi_i) \] (2)
\[ E \{ (g(x) - \hat{x})(g(x) - \hat{x})' \} \approx \sum_{i=0}^{N_x} \omega_i (g(\xi_i) - \hat{x})(g(\xi_i) - \hat{x})' \] (3)
where \( E \) is the expected-value operator. Note that when approximating covariance matrices, only cubature formulae using all positive points and weights should be used to guarantee that the resulting matrix is always positive definite or positive semi-definite.

For the track-initiation problems considered in this paper, process noise is neglected over the time period during which the measurements used for track initiation are collected. This means that the measurements are deterministically related to the target state at different times. If the dimensionality of the collected measurements equals the dimensionality of the state and the state is uniquely determined by the measurements, then there exists a function \( g \) that can transform a set of measurements into a state at the time of the final measurement. In this paper, the function \( g \) is not found explicitly; rather, a probability-1 homotopy method as in Section 3 is used. Given some way to evaluate \( g \), then cubature integration as in (2) and (3) can be used to get a mean and covariance matrix for an initial target-state estimate.

For all of the examples in this paper, fifth-order cubature points and weights are used. The fifth-order cubature points of Algorithm 3-5 in [62, Pg. 317] as formulated for a Gaussian distribution in [14] are used in all sections except Section 6. Since the cubature points of [62, Pg. 317] require that the dimensionality of the state be at least two, they are not suitable for the two-dimensional surface-wave problem in Section 6. Thus, in Section 6, the one-dimensional cubature points of [31] are used with the quadrature rule of [62, Ch. 2], as described in Appendix D of [18]. Appendix A provides the necessary formulae so that the results can be reproduced.

3. PROBABILITY-1 HOMOTOPY METHODS

The General Differential Equation

Probability-1 homotopy methods for solving systems of nonlinear equations are discussed in the tutorials [54, 68–70]. As noted in [48, 69], they can even be applied to constrained optimization problems. Given an \( N \times 1 \) vector-state \( x \), and a function \( f(x) \) providing an \( N \times 1 \) output, the homotopy methods considered in this paper solve the problems \( f(x) = 0 \) or \( f(x) = x \).

Most general homotopy formulations for finding the zero of a function \( f(x) \) begin with
\[ 0 = \lambda f(x) + (1 - \lambda) g(x) \] (4)
where one is given an initial estimate \( x_0 \) that is the zero of the function \( g(x) \). Thus, (4) holds for \( \lambda = 0 \) at the point \( x_0 \). Homotopy methods increase the homotopy parameter \( \lambda \) from zero to one, while keeping (4) satisfied. Thus, when \( \lambda = 1 \), the problem has been solved.

The most common homotopy method presented in textbooks, such as [9, Ch. 10.5], is where one writes a differential equation assuming that \( x \) depends on the homotopy parameter \( \lambda \) and then integrates over \( \lambda \) from zero to one. For example, differentiating (4), one can write the differential equation in \( x \) as
\[ 0 = f(x) + \lambda (\nabla f(x))' \frac{dx}{d\lambda} - g(x) + (1 - \lambda)(\nabla g(x))' \frac{dx}{d\lambda} \] (5)
where the gradient operator is defined to be
\[ \nabla \triangleq \left[ \frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} \ldots \frac{\partial}{\partial x_n} \right]' \] (6)
thus, the Jacobian of a function \( f(x) \) is
\[ (\nabla f(x))' = \left[ \begin{array}{ccc} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \ldots & \frac{\partial f_n}{\partial x_n} \end{array} \right] \] (7)
Solving (5) for \( \frac{dx}{d\lambda} \), one gets
\[ \frac{dx}{d\lambda} = \left( \lambda (\nabla f(x))' + (1 - \lambda)(\nabla g(x))' \right)^{-1} (g(x) - f(x)) \] (8)
where starting at \( x_0 \) for \( \lambda = 0 \), one could numerically solve the initial-value problem, integrating from \( \lambda = 0 \) to \( \lambda = 1 \) to solve for \( f(x) = 0 \). However, it can be seen that problems could arise causing the algorithm to fail if the matrix being inverted is singular. In homotopies appearing in the target-tracking literature, integration directly over \( \lambda \) is the most common approach. For example, such a technique forms the traditional approach to moving particles in a Daum-Huang homotopy particle filter [10, 22], where a slightly different homotopy is used. However, such an approach is not suitable for the problem of finding an inverse function, which is the
nullspace can be implemented using a singular-value decomposition as described in [30, Ch. 2.4]. The matrix should only have a single vector in the nullspace, so in the event there is more due to finite-precision problems, just choose one.

2. The derivative to be used in the integration is

\[
\frac{du}{ds} \pm \left[ \frac{d\lambda}{ds} \right] = \frac{n}{\|n\|}
\]

where \( p = 1 \) or \( p = -1 \) and \( p \) is chosen in a routine from the \( k \)th step to the \( (k+1) \)-th step. That routine integrates along the path in \( \lambda \)-\( x \) space such that the angle between \( du_{k+1}/ds \) and \( du_k/ds \) is less than 180°. That is, the direction along the path is not reversed. The condition that the angle between \( u_{k+1} \) and \( u_k \) be less than 180° is equivalent to stipulating that \( p = -1 \) if \( \left( \frac{du_{k+1}}{ds} \right)^\top \left( \frac{du_k}{ds} \right) < 0 \). For the first step, use \( du_0/ds = [1, 0]' \) to initially start in the direction of increasing \( \lambda \). Sometimes, other solutions can be found by starting with \( du_0/ds = [-1, 0]' \).

The above approach for determining the derivatives is used in the Fortran algorithms for solving homotopy problems in [71, 72] and is essentially the suggested approach in [67]. Note that while integrating over \( s \), \( \lambda \) might not increase monotonically from 0 to 1. That is, the function might double-back a few times before reaching 1. The implementation of an algorithm to integrate over \( s \) until \( \lambda = 1 \) is discussed in Section 3.

Specific Homotopies

The specific problem that needs to be solved is \( f(x) = z \) where \( z \) is a given set of observations, \( x \) is the target-state vector, and \( f \) transforms \( x \) into the measurement domain. In all of the examples in this paper, the combined dimensionality of all of the measurements equals the dimensionality of the target-state vector. There are multiple choices for \( g(x) \) and \( f(x) \) which may be used to solve the problem. In writing this paper, the most relevant choices that were considered are as follows: (a in the following homotopies is some value that satisfies the problem when \( \lambda = 0 \))

1. The homotopy used in the estimator described in [60] for performing bistatic range-only initiation of orbital objects:

\[
0 = \lambda (z - f(x)) + (1 - \lambda)(x - a)
\]

2. A modification of the homotopy for solving for the zeros of a function from [70]:

\[
0 = \lambda(z - f(x)) + (1 - \lambda)(a - f(x))
\]

which is equivalent to

\[
0 = a + \lambda(z - a) - f(x).
\]

3. A modification of an algorithm to solve for the zeros of a function, with a special focus on solving for all real roots of a nonlinear equation, from [53], which expands on the work of [37]. The work of [37] focuses on finding all (real and complex) roots. The original homotopy is

\[
0 = \lambda f(x) + (1 - \lambda)((x - a) + (f(x) - f(a)))
\]

which, if the \( f(a) + a \) term is lumped together into just \( \tilde{a} \) is
equivalent to
\[ 0 = \lambda (\tilde{a} - x) + x + \tilde{f}(x) - \tilde{a}. \tag{22} \]

The modification for finding \( \tilde{f}(x) = z \) instead of \( \tilde{f}(x) = 0 \) is to substitute \( \tilde{f}(x) - z \) for \( \tilde{f}(x) \) in (22).

While solving the examples in this paper, it was observed that the second homotopy above generally had the best performance. Thus, that is the one used in the rest of the paper.

In [53], the third homotopy is used to find all real roots of polynomial equations. In such an instance, one does not stop once \( \lambda = 1 \) but instead continues on to discover other instances when \( \lambda = 1 \). Generally, paths trace out closed loops in \( \lambda \) (assuming no solutions at \( \pm \infty \)). Thus, if there are only two solutions, one can find a second solution by starting along the path in the opposite direction for the second root. That is one must trace out the curve both with \( u_0 = [1, 0]^t \) and \( u_0 = [-1, 0]^t \).

Implementing a Solution

Probability-1 homotopy methods converge in a finite number of steps given a random initialization (finite-precision issues notwithstanding). This is significantly better than using Newton’s method to solve the problem, as a bad initialization can cause the estimator to diverge. The implementation of such algorithms simply consists of integrating a differential equation. For example, when considering the second homotopy in Section 3, the differential equation to be solved is the normalized nullspace basic vector, solving

\[
\left[ (z - a) - (\nabla f(x))^t \right] \left[ \frac{d\lambda}{ds} \right] = 0 \tag{23}
\]

as described in Subsection 3, where \( I_{N \times N} \) is an \( N \times N \) identity matrix.

Initially, one would assume that a simple off-the-shelf algorithm for integrating differential equations, such as the \texttt{ode45} or \texttt{ode113} functions in Matlab, could be used to solve the homotopy problem. However, this is not the case, because the choice of \( p \) in Equation (17) depends on past derivative information that is not made available to the path-following algorithm. Additionally, even on problems that one could successfully solve by always assuming \( p = +1 \), the difficulty arises that one is not integrating over a known span of \( s \). One does not know how far the optimal solution is from the (presumably very bad) initial estimate. All one can say is that the minimum possible distance to the optimal solution is 1, which would be the necessary change in \( \lambda \) if the initial estimate \( x_0 \) were the correct solution to the problem. When using \texttt{ode45} in Matlab, one can integrate until \( \lambda = 1 \) is one using “events.” However, the algorithm is very slow.

A popular package for solving homotopy optimization problems is HOM4PS, which is described in [43] and whose website is at http://hom4ps.org. The package is used for a number of passive localization problems in [65] and for Doppler-only localization in [44]. Unfortunately, the package requires that the problem be specified in terms of a multivariate polynomial system, and thus is not suitable for some of the more general problems considered in this paper.\(^4\)

The solution chosen for use in this article is similar to that used in the HOMPACK Fortran function \texttt{FIXPDF} [71, 72].\(^5\) The HOMPACK package includes a number of different types of homotopy algorithms. In [71], it was observed that while sometimes more computationally demanding than alternative methods, the technique based on using an ordinary differential-equation solver in \texttt{FIXPDF} was the least likely to fail. Thus, such a method is chosen here. Note that the most commonly used routines for solving homotopy problems appear to be based on using a Newton’s method predictor-corrector algorithm. Examples include the routine in HOMPACK and the orbital estimation algorithm of [60].

The homotopy algorithm used in this paper is shown in Fig. 1 with necessary integrator interpolation constants given in Fig. 2. The algorithm is similar to the algorithm described in [67] and consists of running a differential-equation solver until one passes \( \lambda = 1 \) and then performing interpolation between the final two points of the differential-equation-solving routine to find the value of \( x \) when \( \lambda \) is exactly equal to one. However, as noted in [67], no matter how good a differential-equation solver is, the solution will slowly diverge from the constraint in (22) due to finite-precision errors. To ameliorate such problems, the algorithm of [67] recomputes the constant factor (in this instance \( a \)) every few iterations to enforce the constraint. For example, using the second homotopy, the constraint is given by

\[
a = \frac{f(x) - z\lambda}{1 - \lambda} . \tag{24}\]

In the implementation of Fig. 1, this reinitialization occurs after the number of iterations since the last reset (counted in \texttt{nSinceReset} exceeds a constant \texttt{USERSTEP} number of steps. Also, if the algorithm passes \( \lambda = 1 \) and has not reset, then it is reset once before finishing to try to get the most precise solution.

Note that an alternative to using (24) to force the homotopy to hold is to never change \( a \), but rather to use Newton’s method to force the estimate back onto the path. The iteration for Newton’s method with the second homotopy is

\[
x_k = x_{k-1} - \left( - \left( \nabla f(x) \right)^t \right)^{-1} \left( a + \lambda (z - a) - f(x) \right) . \tag{25}\]

This correction is used in the simulations. However, it is used (10 iterations) in place of (24) to force compliance with the homotopy condition when making plots of the homotopy curves. A correction like this (which also changes \( \lambda \)) forms the basis of the predictor-correction family of probability-1 homotopy algorithms, as described in [71].

In the HOMPACK routine \texttt{FIXPDF} [72], the differential-

\(^4\) While writing this paper, it was found that the links on the HOM4PS website to the latest stable version of HOM4PS, which is HOM4PS-2, did not work.

\(^5\) Note that code given in the collected algorithms of the Association for Computing Machinery (ACM) at http://netlib.org/toms/is subject to a license agreement that precludes commercial use without permission from the ACM, thus greatly limiting the utility of the code library and presumably precluding the publication of any U.S. Government-developed algorithms in ACM Transactions on Mathematical Software as U.S. Government work is not subject to U.S. copyright. Thus, the actual HOMPACK Fortran routines were not used in any simulations for this paper. Additionally, the homotopy used in HOMPACK is different than the one used in this paper.
Figure 1. A summary of the algorithm used for finding one solution via the probability-1 homotopy method. Two equal signs mean equality, one means assignment. Additional solutions can be found by starting the algorithm with different initial values except for the distance traveled in that parameter can be discarded. The dependence on that parameter is arbitrary but given no additional prior information should be less than \( s \) if \( \Delta_0 \) is correct. Initializing the algorithm with the sign of \( \Delta a \) flipped should allow one to find a second solution if two solutions exist as would traveling farther along the path after passing the first occurrence of \( \lambda = 1 \).
converge in a given MAXITERNewton iterations, it is declared a failure. The algorithm is embedded, because results for computing the secondary-order solution are reused when computing the primary-order solution. The F indicates the fact that the last function evaluation (in this case the u vector in (17)) per integration step, and F denotes the fact that the last function evaluation of the past step is the first function evaluation of the next step, allowing one to avoid an extra function evaluation after the first step. The M indicates that the algorithm was developed with the objective of minimizing a certain error norm. The algorithm is embedded, because results for computing the primary-order solution are reused when computing the secondary-order solution, thus reducing the total number of derivative evaluations required. The use of Newton’s method on the interpolating polynomial to find the exact value when \( \lambda = 1 \) (within a tolerance TOL\(_\lambda\)) is Dormand’s suggested routine in [27, Ch. 6.6]. If Newton’s method does not converge in a given MAXITER\(_\text{Newton}\) iterations, it is declared a failure.

The step size in the implementation of the homotopy algorithm is determined using the Fehlberg method, which requires estimates from both orders of the Dormand-Prince pair at each step. The Fehlberg method is described simply in [9, Ch. 5.5] and involves the assumption that the error in the estimate is proportional to the difference between the primary and secondary orders of the Runge-Kutta methods. However, the implementation shown in Fig. 1 uses constants called ABSTOL and RELTOL to compare the value \( \epsilon \) to a relative tolerance (a fraction of the parameter size) or to an absolute tolerance, if the parameter becomes too small. Generally, ABSTOL < RELTOL. The error estimate used in [9, Ch. 5.5] uses only an absolute tolerance. However, the Fehlberg step-size correction formula is the same as in [9, Ch. 5.5], with the following ad-hoc changes:

1. The step size is not allowed to increase by more than a factor of 10, to address issues if \( \epsilon \approx 0 \).
2. The step size is not allowed to decrease by more than a factor of 10 the first time a step is rejected, to avoid taking too small a step.
3. The step size is halved rather than using Fehlberg’s routine after the first rejection, so as to avoid the case where Fehlberg’s reduction is too small.

For the simulations in this article, the following values were used for the constants needed in the algorithm, unless otherwise specified:

- RELTOL = \( 10^{-8} \)
- ABSTOL = \( 10^{-10} \)
- MAXITER = 512
- MAXITER\(_\text{Newton}\) = 25
- RESETSTEP = 0
- TOL\(_\lambda\) = \( 10^{-8} \)

Note that the implementation of the homotopy solver in HOMPACK [72], terminates with a failure if the \( \lambda \) term ever goes negative. However, it was noticed that with certain combinations of homotopies and problems \( \lambda \) can go negative while still on a valid path to the correct answer. Thus, the implementation used in this paper allows \( \lambda \) to take on negative values. Moreover, in problems with exactly two real solutions, one must allow \( \lambda \) to be negative in order to get the second solution by starting in the opposite direction along the path; that is, starting with \( \frac{dx}{d\lambda} = [-1, 0]^T \).
4. BISTATIC RANGE-ONLY TRACK INITIATION

The Scenario

The first scenario considered is the problem of bistatic range-only track initiation. In this scenario, as well as in all of the subsequent ones, the effects of light-time are ignored. That is, the motion of the target in the time-interval between when a pulse was transmitted from the receiver to when it hit the target is assumed negligible. Additionally the motion of the target in the interval between when the pulse is scattered by the target to when it is picked up by the receiver is also ignored. These are reasonable assumptions except for certain times when tracking orbital objects. Issues concerning the breakdown of traditional tracking algorithms due to light-time uncertainty are discussed in [18].

The problem of bistatic range-only track initiation is addressed in the context of passive tracking in [23]. However, the solution involves converting multiple simultaneous bistatic range measurements into a Cartesian position. However, as one would expect the target detection probability to be particularly low in passive applications, one would assume that it is generally not ideal to require that multiple receivers simultaneously observe the target in order to start a track. Thus, the primary focus of this scenario is the instances where few or none of the measurements are simultaneous.

In order to handle non-simultaneous measurements, a target dynamic model must be assumed. In this case, a constant-velocity motion model, such as is described in [2, Ch. 6.2.2] is used. The model is

\[ x_{k+1} = F_k x_k \]

(26)

where \( x \) is the target state and \( k \) indicates the discrete-time step (step times coincide with measurements). The target state consists of Cartesian position and velocity components and is ordered \( x = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T \). The matrix \( F \) is the state transition matrix and is

\[
F_k = \begin{bmatrix}
1 & 0 & 0 & T_k & 0 & 0 \\
0 & 1 & 0 & 0 & T_k & 0 \\
0 & 0 & 1 & 0 & 0 & T_k \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

(27)

where \( T_k \) is the time interval between discrete step \( k \) and step \( k+1 \). The position components can be extracted by multiplying by the matrix \( H \) as

\[ r_{\text{Tar}}^k = H x_k \]

(28)

where

\[
H = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
\]

(29)

Because there is no process noise in the model, it is simple to propagate the target state being estimated to any time when there is an observation. In the discussion below and in the simulations, the state is at time \( k = 1 \). For a transmitter at time \( k \) at position \( r_{\text{Tar}}^k \) and a receiver at position \( r_{\text{Rx}}^k \), the bistatic range is

\[ r_k = ||r_{\text{Rx}}^k - r_{\text{Tar}}^k|| + ||r_{\text{Rx}}^k - r_{\text{Tar}}^k|| \]

(30)

where

\[
r_{\text{Tar}}^k = \begin{bmatrix}
H x_1 \\
H F_1 x_1 \\
H F_1 F_1 x_1 \\
\vdots
\end{bmatrix}
\]

(31)

and where \( x_1 \) is the target state at the time of the first measurement. Similar equations can be derived by flipping the sign of \( T_k \) and reversing the order of \( k \) in the equations if one wishes to express \( r_{\text{Tar}}^k \) in terms of \( x_N \), which can be more useful for recursive tracking applications.

The function \( f(x) \) in the homotopy method for this problem is \( f(x) = [f_1, f_2, \ldots, f_N]^T \). The homotopy method of Section 3 requires partial derivatives that are assembled into the matrix \( \nabla f(x)^T \) in (7), for which this model can be explicitly solved. The partial derivative with respect to component \( x_p \) of \( x \) is

\[
\frac{\partial r_k}{\partial x_p} = \frac{(r_{\text{Tar}}^k - r_{\text{Rx}}^k)^T \partial r_{\text{Tar}}^k}{||r_{\text{Tar}}^k - r_{\text{Rx}}^k||} + \frac{(r_{\text{Rx}}^k - r_{\text{Tar}}^k)^T \partial r_{\text{Rx}}^k}{||r_{\text{Rx}}^k - r_{\text{Tar}}^k||}
\]

(32)

where

\[
\frac{\partial r_{\text{Tar}}^k}{\partial x_p} = \begin{bmatrix}
H \frac{\partial x_1}{\partial x_p} \\
H F_1 \frac{\partial x_1}{\partial x_p} \\
H F_1 F_1 \frac{\partial x_1}{\partial x_p} \\
\vdots
\end{bmatrix}
\]

(33)

Note that \( \frac{\partial x_1}{\partial x_p} \) is a vector of zeros with a 1 in the place of component \( x_p \). For a general \( F_k \) matrix, the order of the multiplications in the general product in (33) matters. In the problem at hand, the successive multiplications are equivalent to forming an \( F \) matrix as in (27) except \( T_k \) is replaced with the sum of the different \( T_i \) values of the different \( F_k \) matrices that are multiplied together.

Equations similar to those given above are necessary for solving the problem of localizing a target given three simultaneous bistatic range measurements. Initially, the use of a homotopy method to solve such a problem would appear to be pointless given that an explicit solution (with an approximate covariance matrix) is given in [49]. However, as shall be demonstrated, the homotopy method appears to be less susceptible to finite-precision errors and the use of cubature integration produces a more statistically consistent covariance matrix.
Consider three scenarios for localization:

1. All sensors are located in the same plane. Thus, their \( z \)-coordinates are zero.
2. All sensors are at an elevation of 10 m.
3. The first sensor is at an elevation of 1 km and the other sensors are at an elevation of 10 m.

The range-only localization problem has two solutions. This should be clear from the first and second scenarios for localization where there is nothing to differentiate whether the solution is above or below the \( x - y \) plane. The algorithm of [49] can provide both solutions. However, in the first scenario, a singular matrix must be inverted meaning that a pseudoinverse must be used.

The probability-1 homotopy algorithm described in Fig. 1 provides a single solution. If run with the sign of \( \frac{du_0}{d\lambda} \) flipped, it might provide a second solution. However, in Scenario 2, no solution was obtained when the algorithm was run in the direction of decreasing \( \lambda \). Rather, to obtain the second solution, one must continue after reaching the \( \lambda = 1 \) point and search for a second crossing of the \( \lambda = 1 \) threshold. Thus, the method used to obtain all solutions is to attempt to obtain two solutions in either direction. However, it is possible that one obtains up to two solutions in both directions (4 total). This is not because the problem has four solutions, but because the homotopy doubles back and the extra solutions are within finite-precision bounds of other solutions. Thus, for all examples in this paper, a solution had to be at least 10 m away from an existing solution to be declared unique.

Table 1 lists the position estimates obtained by the spherical-intersection routine as well as the probability-1 homotopy method given noise-free bistatic range measurements from Sensors 1 through 3 in Fig. 3 in each of the scenarios. The summed absolute difference between the ranges obtained with the estimates and the true ranges are also provided. It can be seen that the spherical-intersection algorithm does extremely poorly in the first scenario, where a pseudoinverse has to be used to handle a singular matrix. The spherical-intersection method also is subject to more finite-precision errors in the Scenarios 2 and 3. Thus, there is value in using a probability-1 homotopy method even when an explicit solution is available.

Figure 4 shows the components of the path that is followed by the homotopy method when finding the solutions for the first scenario. It can be seen that the homotopy curves form closed loops that pass through all of the solutions. The ability of a single homotopy curve to provide all solutions is useful and is utilized, for example, in [60] to find all solutions to the bistatic range-only orbit determination problem.

Now the problem of obtaining an unbiased estimate of the target location and an accurate covariance matrix given noisy measurements, is considered. The spherical-intersection algorithm of [49] generates a covariance matrix based on a Taylor series approximation. This is chosen as a baseline for comparison to other methods. Also, only the third scenario is considered. Though two solutions are obtained each time, only the first solution with a positive \( z \)-coordinate is used. In practice, if the sensors remain on or near the ground, there should not be any ambiguity as to the solution to use. It is assumed that the range measurements are corrupted with independent Gaussian noise having a standard deviation of 10 m. Figure 5 plots the root-mean-squared error (RMSE) and the average normalized estimation-error squared (ANEES) of the estimates for 500 Monte-Carlo runs as a function of the \( y \) value of the target location. The ANEES is a measure of

---

**Simulation Examples**

Two examples are considered for analyzing the combination of a probability-1 homotopy method with a cubature routine. The first is a simple range-only localization problem involving one transmitter and three receivers that try to localize the target in three dimensions. It is interesting to compare the results of the combined cubature integration and probability-1 homotopy method as applied to this problem with an explicit solution given in [49], which comes with a covariance estimate. The second example is the more complicated problem of using bistatic measurements over time to estimate the position and velocity of a moving target. The basic scenario used is illustrated and described in Fig. 3.

**Bistatic Range-Only Localization**

In the first problem considered, the first receiver in Fig. 3 broadcasts and Receivers 1 through 3 receive simultaneously. Given three bistatic range-only measurements, the location of the target can be ascertained. In this instance, a combination of the probability-1 homotopy method with cubature integration can be used to obtain an unbiased estimate of the Cartesian target location as well as a covariance matrix. The mean and covariance matrix could for example be passed to a simple Kalman filter for tracking, as in [14].

Since atmospheric refraction is not taken into account, this problem is algebraically simple. It might seem that there is no point in using a probability-1 homotopy method to solve it as a closed-form solution (the spherical-intersection method) is given in [49]. Thus, it shall first be demonstrated that the probability-1 homotopy method is more robust.

Consider three scenarios for localization:

1. All sensors are located in the same plane. Thus, their \( z \)-coordinates are zero.
2. All sensors are at an elevation of 10 m.
3. The first sensor is at an elevation of 1 km and the other sensors are at an elevation of 10 m.

The range-only localization problem has two solutions. This should be clear from the first and second scenarios for localization where there is nothing to differentiate whether the solution is above or below the \( x - y \) plane. The algorithm of [49] can provide both solutions. However, in the first scenario, a singular matrix must be inverted meaning that a pseudoinverse must be used.

The probability-1 homotopy algorithm described in Fig. 1 provides a single solution. If run with the sign of \( \frac{du_0}{d\lambda} \) flipped, it might provide a second solution. However, in Scenario 2, no solution was obtained when the algorithm was run in the direction of decreasing \( \lambda \). Rather, to obtain the second solution, one must continue after reaching the \( \lambda = 1 \) point and search for a second crossing of the \( \lambda = 1 \) threshold. Thus, the method used to obtain all solutions is to attempt to obtain two solutions in either direction. However, it is possible that one obtains up to two solutions in both directions (4 total). This is not because the problem has four solutions, but because the homotopy doubles back and the extra solutions are within finite-precision bounds of other solutions. Thus, for all examples in this paper, a solution had to be at least 10 m away from an existing solution to be declared unique.

Table 1 lists the position estimates obtained by the spherical-intersection routine as well as the probability-1 homotopy method given noise-free bistatic range measurements from Sensors 1 through 3 in Fig. 3 in each of the scenarios. The summed absolute difference between the ranges obtained with the estimates and the true ranges are also provided. It can be seen that the spherical-intersection algorithm does extremely poorly in the first scenario, where a pseudoinverse has to be used to handle a singular matrix. The spherical-intersection method also is subject to more finite-precision errors in the Scenarios 2 and 3. Thus, there is value in using a probability-1 homotopy method even when an explicit solution is available.

Figure 4 shows the components of the path that is followed by the homotopy method when finding the solutions for the first scenario. It can be seen that the homotopy curves form closed loops that pass through all of the solutions. The ability of a single homotopy curve to provide all solutions is useful and is utilized, for example, in [60] to find all solutions to the bistatic range-only orbit determination problem.

Now the problem of obtaining an unbiased estimate of the target location and an accurate covariance matrix given noisy measurements, is considered. The spherical-intersection algorithm of [49] generates a covariance matrix based on a Taylor series approximation. This is chosen as a baseline for comparison to other methods. Also, only the third scenario is considered. Though two solutions are obtained each time, only the first solution with a positive \( z \)-coordinate is used. In practice, if the sensors remain on or near the ground, there should not be any ambiguity as to the solution to use. It is assumed that the range measurements are corrupted with independent Gaussian noise having a standard deviation of 10 m. Figure 5 plots the root-mean-squared error (RMSE) and the average normalized estimation-error squared (ANEES) of the estimates for 500 Monte-Carlo runs as a function of the \( y \) value of the target location. The ANEES is a measure of
Table 1. The two estimates obtained using the probability-1 homotopy method and the spherical-intersection method to localize a target given 3 error-free bistatic measurements for each of the three scenarios of Subsection 4. The summed absolute errors of the ranges implied by the solutions are also compared to the ranges due to the true target (located at $(-20 \text{ km}, 100 \text{ km}, 7 \text{ km})$). Due to finite-precision problems and issues regarding matrix singularity, the simple explicit spherical-intersection algorithm performs notably worse than the probability-1 homotopy method. All computations were obtained using double-precision arithmetic. In all instances, an (extremely bad) initial estimate of $x_0 = [10 \text{ m}, 10 \text{ m}, 10 \text{ m}]$ was used to initialize the homotopy method.

![Coordinate Value (km)](image)

Figure 4. The evolution of the estimates in the probability-1 homotopy method for finding the two solutions of the noise-free version of Scenario 1 of Subsection 4. The (extremely bad) initial estimate of $x_0 = [10 \text{ m}, 10 \text{ m}, 10 \text{ m}]$ is used. The plot traces out the evolution of each of the coordinates as a function of $\lambda$. The solution follows a closed loop in homotopy space.

Table 2 lists the two solutions obtained from the probability-1 homotopy method given error-free range estimates. Unlike in the triangulation case, both solutions have positive values of $z$. Therefore, finding the correct solution is more difficult, though it could be made simpler using Doppler information.

To demonstrate the use of the combination of the probability-1 homotopy method with cubature integration, the target is given with a true initial state of $\mathbf{x} = [-20 \text{ km}, y, 7 \text{ km}, 0, 0, -150 \text{ m/s}]$, where $y$ ranges from 20 km to 50 km. A total of 25 Monte-Carlo runs were performed. Each time, the algorithm was run it yielded two solutions. Solutions obtained using the same initial conditions (i.e. using the same value of $\Delta t$ initially and arriving in the same order when multiple solutions along a single path were found) were assumed to be the same across cubature points.

Solutions were associated across subsequent Monte-Carlo runs by pairing the ones with the smallest summed Mahalanobis distances between the current and previous Monte-Carlo run. That is, given current state estimates $\hat{x}_1$ and $\hat{x}_2$,

\[ d = \sqrt{(\hat{x}_1 - \hat{x}_2) \cdot R^{-1} (\hat{x}_1 - \hat{x}_2)^T} \]

where $R$ is the true covariance matrix of the target.

![Diagram](image)

Figure 5. The ANEES is a quick method of observing covariance matrix consistency. However, it has its shortcomings. A more thorough analysis would make use of more advanced covariance consistency measures such as the noncredibility index and the inclination indicator of [46].
The summed range error is explained in [3, Ch. 2.3.2]. The relationship between the covariance matrices and ellipsoid confidence regions based on the covariance matrix obtained by cubature integration, assuming that the estimates are Gaussian distributed. Twenty-five Monte-Carlo runs were performed. The probability-1 homotopy method never failed to solve the inverse problem for any of the cube sizes in any of the Monte-Carlo runs. A range standard deviation of 10 m was used to generate the independent Gaussian noise corrupting the measurements. The block horizontal lines in (b) are the 95% confidence region of the ANEES. The combination probability-1 homotopy method/cubature integration produced consistent results, whereas the Taylor series-based covariance matrix for the spherical-intersection method was extremely large (from 181.76 to 1443.2), outside of the bounds of the plot. The initial estimate for the probability-1 homotopy method was fixed to \( \hat{x}_0 = [10 \text{ m}, 10 \text{ m}, 10 \text{ m}]^T \).

\[
\begin{array}{cccccc}
 x & y & z & x & y & z \\
-20.000 \text{ km} & 100.00 \text{ km} & 7.0000 \text{ km} & 16.396 \text{ nm/s} & 24.752 \text{ nm/s} & -150.00 \text{ m/s} \\
-19.867 \text{ km} & 100.28 \text{ km} & 2.7653 \text{ km} & -6.5607 \text{ m/s} & -10.390 \text{ m/s} & -190.04 \text{ m/s} \\
\end{array}
\]

Table 2. The two target state solutions obtained using the probability-1 homotopy algorithm with noise-free bistatic range-only measurements in the problem of Subsection 4. The summed absolute errors of the ranges implied by the solutions are also compared to the ranges due to the true target (with a state of \( \hat{x}_0 = [10 \text{ m}, 10 \text{ m}, 10 \text{ m}/\text{s}, 10 \text{ m}/\text{s}, 10 \text{ m}/\text{s}]^T \)). Their small size indicates that the solutions are both valid; the first solution is clearly the correct one. Attempts to continue farther along the solution curve produce repeats of these, indicating that the homotopy forms a closed loop.

\[ \hat{x}_2 \text{ with associated covariance matrices } P_1 \text{ and } P_2, \] the association cost function is given by

\[
\text{cost} = (\hat{x}_j - \hat{x}_1^{\text{prev}}) (P_1 + P_1^{\text{prev}})^{-1} (\hat{x}_j - \hat{x}_1^{\text{prev}}) \\
+ (\hat{x}_j - \hat{x}_2^{\text{prev}}) (P_2 + P_2^{\text{prev}})^{-1} (\hat{x}_j - \hat{x}_2^{\text{prev}}) 
\]

(34)

where the terms with the superscript “prev” are the estimates from the previous Monte-Carlo run, and the assignment of \( i \) and \( j \) to 1 and 2 changes the cost. Thus, if the lowest cost is the one including the term \( \hat{x}_1 - \hat{x}_1^{\text{prev}} \), then the current Monte-Carlo run is assigned in the same order as the last one; otherwise it is flipped.

Figure 6 shows the RMSE and the NEES of the simulations with their 99.96% confidence regions based on the covariance matrix obtained by cubature integration, assuming that the estimates are Gaussian distributed. Twenty-five Monte-Carlo runs were performed. Figure 7 shows plots of the average location and confidence region of the estimates. The relationship between the covariance matrices and ellipsoid confidence regions is explained in [3, Ch. 2.3.2].

Although the algorithm was able to start tracks at the points used in Fig. 7, difficulties arose at some other points due to the use of a non-random initialization. Consider, for example, the observation vector

\[
\mathbf{r} = \begin{bmatrix}
6.549395717733673 \\
3.397559887761309 \\
6.041497354208824 \\
6.408764048254604 \\
3.185113537131574 \\
5.98776176393080
\end{bmatrix} \times 10^4 \text{ m.}
\]

(35)

This vector arose in a simulation when running the probability-1 homotopy method on a cube point originating from a noisy measurement in a scenario where \( y = 1.11 \times 10^4 \text{ m} \). The probability-1 homotopy algorithm, described in Section 3, failed to find a solution. Yet, the plot of the homotopy curve shown in Fig. 8 appears to show two valid solutions. However, the curve never actually touches the \( \lambda = 1 \) point, most likely due to the initialization not being good enough and/or finite-precision errors. When run
Figure 6. The RMSE and NEES of the estimate (of the two solutions) closest to the true estimate for the range-only track-initiation (3D position and velocity) problem. The true-target state was always \( x = [-20 \text{ km}, y, 7 \text{ km}, 0, 0, -150 \text{ m/s}] \), where \( y \) was varied. The bistatic range measurements (taken as described in the text) always had a standard deviation of 10 m and were corrupted with Gaussian noise. The sensors used are as described in the text. 25 Monte-Carlo runs were performed. The horizontal lines are the 95\% confidence region of the NEES. It can be seen that the results are statistically consistent with the covariance matrix.

Figure 7. The two solutions obtained (position components shown in 2D) for the multistatic non-simultaneous range-only track-initiation (3D position and velocity) problem for different \( y \) values of the true solutions using the combined cubature and probability-1 homotopy methods averaged over 25 Monte-Carlo runs. The blue solutions (left) are associated with the true solution and the red (right) are corresponding ghost solutions that would have to be eliminated through further track filtering or using additional measurement components, such as Doppler. The ellipses represent the 99.97\% confidence regions based on the obtained covariance matrix and assuming a Gaussian distribution for the estimates. The same Monte-Carlo runs as used in Fig. 6 were used here.

Figure 8. A plot of the evolution of the three Cartesian position coordinates given the initial conditions for the example where the probability-1 homotopy algorithm as described in the text failed for the non-simultaneous bistatic range-only track-initiation problem. The initial state was \( x_0 = [10 \text{ m}, 10 \text{ m}, 10 \text{ m/s}, 10 \text{ m/s}, 10 \text{ m/s}]^\top \). It looks like the homotopy curves cross the \( \lambda = 1 \) boundary twice, corresponding to the two solutions that one would expect. However, the curves only get close without getting within \( \text{TOL}_\lambda \) of 1. This is due to the use of a poor (non-random) initialization. Using the truncated random estimate of \( x_0 = [2638 \text{ m}, 1455 \text{ m}, 1360 \text{ m}, 8692 \text{ m/s}, 5797 \text{ m/s}, 5498 \text{ m/s}]^\top \), the algorithm successfully produces two reasonable estimates when using \( \text{TOL}_\lambda = 10^{-5} \).
using a random initialization and different parameters (as described in the caption to Fig. 8), the algorithm succeeded. Section 5 demonstrates a much more egregious example of a poor, non-random initialization causing the algorithm to fail. Probability-1 homotopy methods can fail for initializations chosen in certain finite regions, which is why a robust algorithm might extend the present algorithm with multiple initializations.

5. BISTATIC
TIME-DELAY-OF-ARRIVAL-ONLY TRACK
INITIATION

The Scenario

The problem of track initiation using multistatic TDOA measurements is similar to that of range-only initiation, but has other applications. In general, TDOA measurements are useful when passively tracking emitting objects, because one cannot obtain a range measurement when the time of target transmission. TDOA-based localization (multilateration) is particularly useful in detecting attempts to disrupt civilian air traffic control systems that rely on Automatic Dependent Surveillance-Broadcast (ADS-B) information sent out by transponders on aircraft.

ADS-B signals identify aircraft and contain information regarding the location of aircraft based on global navigation satellite system (GNSS) signals received by the aircraft. However, as noted in [13, 56], the system can be easily disrupted using hardware costing under $2,000 U.S. to create false targets. Thus, as noted in [58, 61], a common method to verify that the estimated location is within a sufficient error tolerance of the location reported by the transponder.

In [29], an explicit closed-form solution to the multilateration problem is provided. The application of cubature integration with probability-1 homotopy methods can extend the applicability of the multilateration solution to more complex environments. For example, there has long been interest in satellite-based transponder tracking of aircraft as evinced by patents filed prior to the advent of ADS-B [21] and by the patent [59] filed after the advent of ADS-B related to satellite multilateration techniques. Using a ray-traced atmospheric model, one can go beyond such work to account for atmospheric refraction and delay to get a more accurate measurement model than the geometric models used in [29] and standard texts on multilateration for aviation. Though an explicit solution to such models would not be practical, a combination of a probability-1 homotopy method and cubature integration could be used to get an unbiased estimate of the source location along with a confidence estimate in the form of a covariance matrix. This section demonstrates the basic principle of this idea in terms of the same geometric model used in [29], as the development of a full ray-traced propagation model is beyond the scope of this article.

We wish to consider both the multilateration case, which is in the literature, as well as the case where not enough simultaneous TDOA measurements are available at one time (for example, one sensor has a missed detection). In the latter case, TDOA measurements from multiple times are used to get a complete state vector consisting of position and velocity. This is a generalization of the TDOA positioning problem that is solved using a homotopy method in [65]. Such an example might arise in aircraft surveillance in the case when an aircraft crosses between the edges of multiple surveillance regions. The formulation of the problem allows the sensors to move as well as the target, making it general enough to be representative of a satellite-surveillance scenario. As was the case in Section 4, a constant-velocity model is assumed for the aircraft.

The model for a TDOA measurement where the time difference is taken between sensor $a$ located at $r^a_k$ at discrete time $k$ and sensor $b$ located at $r^b_k$ is

$$
\Delta t^{(a,b)}_k = \frac{1}{c} (||r^a_k - r^T_k|| - ||r^b_k - r^T_k||)
$$

(36)

where $c$ is the (assumed-constant, i.e. no refraction) speed of propagation of the signal (e.g. the speed of light) and $r^T_k$ is the location of the target at time $k$. For simplicity, it is assumed that the delays are multiplied by $c$ to get range differences

$$
\Delta r^{(a,b)}_k = ||r^a_k - r^T_k|| - ||r^b_k - r^T_k||.
$$

(37)

The partial derivative of a range difference with respect to a component $x_p$ of a target state $x$ (which can consist of position and velocity components) is

$$
\frac{\partial \Delta r^{(a,b)}_k}{\partial x_p} = \frac{(r^T_k - r^a_k)' \frac{\partial r^T_k}{\partial x_p} - (r^T_k - r^b_k)' \frac{\partial r^T_k}{\partial x_p}}{||r^T_k - r^a_k|| - ||r^T_k - r^b_k||}.
$$

(38)

The partial derivatives $\frac{\partial r^T_k}{\partial x_p}$ are the same as in (33) in Section 4.

Simulation Example

The problem of track initiation using 6 non-simultaneous TDOA measurements is considered. The simulation scenario is the same as Scenario 1 used in Section 4. Ship 1 is always used as the receiver against which all time delays are references. Measurements from the other ships come $T_b = 1$ second apart in time. The other ships receive in the order 2,3,4,2,3,4. No simultaneously received measurements from the other ships are considered. The simulation is run using the pseudo-range values of (37) with a measurement standard deviation of 10 m.

As in Section 4, the algorithm is run with true state vectors of $x = [-20 \text{ km}, 7 \text{ km}, 0, 0, -150 \text{ m/s}]'$, where $y$ was varied. Unlike in Section 4, the probability-1 homotopy algorithm performs poorly using a constant initialization of $x_0 = [10 \text{ m}, 10 \text{ m}, 10 \text{ m/s}, 10 \text{ m/s}, 10 \text{ m/s}]'$. Thus, the algorithm was run with an initialization of $x_0^{(0)} = [10 \text{ m}, 10 \text{ m}, 0 \text{ m/s}, 0 \text{ m/s}, 0 \text{ m/s}]'$, and if that failed to provide two solutions, then $x_0^{(1)} = [10 \text{ km}, 10 \text{ km}, 0 \text{ m/s}, 0 \text{ m/s}, 0 \text{ m/s}]'$ was used. Fig. 9 shows the results of the algorithm for a single Monte-Carlo run.

When run as described using two initializations, the algorithm appears to be less likely to fail than the range-only non-simultaneous case of Section 4. However, in this instance it is enlightening to see how the homotopy algorithm fails with the two different initializations used.
goal is to localize a ship given simultaneous multistatic TDOA-only measurements. However, in this instance it is assumed that measurements of a ship are taken using multiple high-frequency surface-wave (HFSW) radars. Thus, the simple geometric relationship for the time delays in terms of Cartesian position used in Section 5 will not hold.

The HFSW measurement models used in much of the tracking literature are generally low fidelity, becoming increasingly worse as the distance from the radars increases. For example, in [1, 7, 8, 25, 32, 51, 55], HFSW radars are modeled as providing range and direction-of-arrival measurements as two polar coordinates in a local polar-coordinate system residing in the local tangent plane to the Earth. The papers do not address how such an approximation could be mapped to actual locations on the surface of the Earth.

Much of the literature on the propagation of electromagnetic waves over the Earth’s surface focuses on how inhomogeneities in the composition of the surface diffract waves (for example, how radio is received over hills), such as in [66]. However, the literature on surface waves across objects other than the Earth does employ forms of ray tracing, which would be amenable to use in target-tracking algorithms. For example, in [5] it is demonstrated that under certain assumptions electromagnetic waves moving across a curved surface can be traced as rays whose propagation follows Fermat’s principle of least time. Fermat’s principle of least time states that the time taken for a photon to travel between two points will be a local minimum when considering possible trajectories. Fermat’s principle was used in [15] to derive a ray-tracing measurement model for observing airborne targets in the absence of surface-wave effects. The principle shall be used here for modeling surface-wave measurements.

Assuming a constant propagation speed across the surface of an ideal, flat ocean on an ellipsoidal Earth, the solution to the surface-wave propagation problem is the same as the direct or indirect geodetic problems of navigation, depending on exactly which type of measurement problem one wishes to solve. The direct geodetic problem of navigation answers the question where one will end up after traveling a certain distance on the straightest possible path on the curved Earth, given an initial heading. As demonstrated in [20], the solution is directly tied to methods for simulating maneuvering targets or above the curved Earth. The indirect geodetic problem, on the other hand, takes two points of the surface of the Earth and answers the question of how far and in which initial direction one must travel to get from one point to the other along the shortest path.

The indirect geodetic problem can be efficiently solved using the algorithm described in [40, 41], which can be downloaded from http://geographiclib.sourceforge.net. The range returned by the solution to the indirect geodetic problem is the distance along the surface of the curved Earth from the target to the receiver. Denote the range obtained by solving the direct geodetic problem from the target to receiver $a$ as $r^a(Tar)$. The TDOA of a signal from the target received by receiver $a$ and $b$, taking receiver $b$ as the reference, is

$$\Delta T^{(a,b)} = \frac{1}{c} \left( r^a(Tar) - r^b(Tar) \right)$$

where $c$ is the assumed-constant speed of propagation of the surface wave. For simulation purposes, it is assumed that

---

**Figure 9.** Estimates and uncertainty ellipses produced in the $x - y$ coordinates of a full (3D position and velocity) target-state estimate obtained by the combination of the probability-1 homotopy method and cubature integration. The results for a single Monte-Carlo run as given in the context of the TDOA estimation scenario in Subsection 5 are shown. Two solutions were obtained for each run. They are shown in red (left) and blue (right) with the solution on the right corresponding to the true solution to the problem (The tracking algorithm does not know this is the true solution and would need to consider multiple hypotheses).

Consider the case of the following set of pseudo-range measurements

$$r = \begin{bmatrix} -0.380474297506655 \\ 2.361436287573945 \\ 1.8890895833787 \\ -0.383883067594687 \\ -2.384769818163341 \\ 1.894406704554572 \end{bmatrix} \times 10^4 \text{ m.}$$  \(39\)

Figure 10 shows the homotopy curves when the probability-1 homotopy algorithm is initialized with two different initializations. In the first instance, the homotopy curve is so far away from $\lambda = 1$ that the failure to cross $\lambda = 1$ cannot be attributed to finite-precision errors. In the second instance, the homotopy curve very clearly crosses $\lambda = 1$ twice, making the estimation problem quite simple. Probability-1 homotopy algorithms converge with probability-1 with a random initialization. This means there can exist certain regions that are bad initializations, as illustrated in this case.

### 6. Bistatic TDOA-Only Surface-Wave Target Localization

**The Scenario**

One might be able to obtain closed-form solutions to the other examples of difficult track initiation in this paper, provided enough effort was spent simplifying and solving the systems of nonlinear equations. However, this section demonstrates that probability-1 homotopy methods are well-suited for solving problems in more difficult scenarios. The
A Simulation Example

The use of the algorithm for surface-wave localization is demonstrated on an example near Hawaii. The sea surface is approximated as matching the U.S. Department of Defense’s World Geodetic System 1984 (WGS84) reference ellipsoid defined in [24], which has a semi-major axis of 6378137.0 m and a flattening factor of 1/298.257223563. The algorithms described in [40, 41] were used to solve the direct geodetic problem. The target was assumed to be on the surface of the ocean and had its state estimated directly in geodetic latitude and longitude (an ellipsoidal coordinate system). The reference receiver was located on the island of Maui at 20.765382° North latitude and −155.978592° East longitude (that is +155.978592° West longitude). The other receivers were at latitude, longitude pairs of (20.231756°, −155.761268°) and (20.126059°, −155.555275°), which are both on the island of Hawaii. The true target was located at (25.519360°, −139.887397°), which is a geodetic distance of approximately 1,728 km away from the radar on Maui. Numerical differentiation necessary for the probability-1 homotopy algorithm was performed using $\epsilon = 10^{-4}$ radians (the target state was computed in radians, not degrees). The measurements were taken assuming a pseudorange standard deviation of $\sigma_r = 10$ m. Averaging in latitude and longitude in the cubature routine should be done accounting for the spherical nature of the globe as described in [17], though the effect does not matter much if one is far from the poles and the ±180° longitude point.

Figure 11 shows the results of the combination of the probability-1 homotopy algorithm and cubature integration routine initialized with the point (20.505977°, −155.883491°), which lies between the islands of Maui and Hawaii. The algorithm was able to successfully localize the target and provide a reasonable covariance ellipse. The algorithm can actually produce two solutions. However, the second solution appears to always be on the other side of the globe. For example, in this instance, the second solution produced by the combination probability-1 homotopy method and cubature integration was at (−20.500898879499871°, 24.181575060002753°), which is located in the South Atlantic. In general, it appears that the solution obtained by starting with $du/ds = [1, 0]^T$ is going to be the one in the Pacific, so there should be no need to search for additional solutions.

7. Recognizing More Difficult Problems

The great success of cubature integration coupled with probability-1 homotopy methods for track initiation in the previous sections when given a sufficiently random initialization can make it seem as if the technique can be used on almost any problem. However, due to finite-precision issues, the structure and formulation of the problem can be a major determining factor in whether the homotopy method of Section 3 will succeed.

A simple example of how the formulation of a problem can cause problems with probability-1 homotopy methods is exemplified by the two formulations of a problem taken from Problem 3 of [37] involving kinetics in a stirred reactor.
Figure 11. A sample Monte-Carlo run of the multistatic TDOA-only surface-wave localization algorithm. The true target is the green ×. The red ellipse (it looks like a line at this scale) with the red dot is the 99.97% uncertainty ellipse for the estimate and the mean value. The true-target location is within the uncertainty ellipse. The sensors are the yellow dots on the Hawaiian islands. The TDOA measurements were taken using the northernmost receiver (on the island of Maui) as a reference. The standard deviation of the noise for the pseudorange measurements corresponding to the TDOA measurements was 10 m. The background image is the land and bathymetry of the area taken from an equirectangular projection provided by Visible Earth, which is run by the EOS Project Science Office at the NASA Goddard Space Flight Center. Full credits for the image are at [63, 64]. The brightness of the image has been increased from its original format. Only one solution is shown. Another (false) solution on the other side of the globe was also found.

which is used as an example in [53]. The first formulation of the problem is to find the zero of the scalar function (given $T > 0$)

$$f_1(T) = \frac{1}{T^2} e^{\frac{10500}{T^4}} - 1.11 \times 10^{11}$$

(42)

where the necessary derivative with respect to $T$ required to use the probability-1 homotopy method is

$$\frac{df_1(T)}{dT} = -2 \left( \frac{10500 + T}{T^4} \right) e^{\frac{10500}{T^4}}.$$  

(43)

The second formulation of the same problem is

$$f_2(T) = 2T \ln\{T\} - 21000 + \ln\{1.11 \times 10^{11}\} T$$

(44)

where the derivative is

$$\frac{df_2(T)}{dT} = 2 + \ln\{1.11 \times 10^{11}\} + 2 \ln\{T\}.$$  

(45)

Solving the above two formulations of the same problem using the homotopy method presented in this paper with an initial guess of $T = 273.15$ yields a solution of approximately 376.6790 using $f_1$ and 551.7738 using $f_2$. The solution using $f_2$ is essentially correct within precision bounds, whereas the solution using $f_1$ is extremely bad. An examination of the steps within the algorithm shows that the tolerance bounds in $f_1$ are not sufficiently small to enforce the homotopy condition in (22) after each step. If the initial guess were taken to be $T = 10$, then the solution using $f_1$ would fail due to finite-precision errors causing a NaN value to arise, and the solution using $f_2$ would still succeed. Thus one can see that the scaling of a problem can greatly influence the accuracy and success of the algorithm.

While the scaling of the problem in the above example makes the finite-precision problems quite clear, other problems can be difficult due to the stiffness of the differential equations involved.

8. CONCLUSIONS

The combination of a probability-1 homotopy algorithm and cubature integration can solve unbiased measurement-conversion problems in scenarios for which no closed-form solution is available. This was demonstrated with the example of obtaining a position estimate with associated covariance matrix of a target on the ocean using only bistatic surface-wave TDOA measurements. Such a problem does

8Note that double-precision is used in all simulations.
not appear to have been previously solved in the literature and can be increasingly used for passively detecting floating and shipborne surface-wave radars, which have been studied by numerous authors [6, 28, 39, 45]. Moreover, though the simulation did not model correlations between the measurement components, such correlations can be easily taken into account by modifying the covariance matrix used with the cubature points.

More detailed analyses were provided with the examples of bistatic range-only target localization and track initiation. It was demonstrated that bistatic range-only localization using a probability-1 homotopy method combined with cubature integration is more resilient in the face of finite-precision errors than a closed-form solution. Additionally, it produces significantly more consistent covariance matrices than methods of obtaining a covariance matrix based on a Taylor-series approximation.

Non-simultaneous bistatic range-only and TDOA-only track initiation (unbiased position and velocity estimation with an associated covariance matrix) were also successfully demonstrated. However, it was also shown that the use of hand-chosen non-random inputs can cause the algorithm to fail. The advantage in using a probability-1 homotopy algorithm to solve inverse problems is that it will converge with probability 1 given a random initial estimate. In practice, however, some difficulties can arise as there are occasionally degenerate scenarios, and some initial estimates can fail to converge. In the case of TDOA-only track initiation, multiple (extremely awful) deterministic initial estimates were chosen to assure convergence of the probability-1 homotopy algorithm (the convergence region of which is massively larger than Newton’s method).

This work appears to be the only published instance of complete track initiation using non-simultaneous range-only and TDOA-only measurements (i.e. not enough measurements are available at any one time to obtain a complete position estimate). Probability-1 homotopy methods are powerful in that they can find multiple solutions to a problem when reinitialized properly after finding a previous solution. The probability-1 homotopy algorithm used in this paper stands in contrast to that used for Doppler-only track initiation in [44] in that it does not require that the measurement model be polynomial, allowing for much more general models such as the surface-wave track-initiation example.

Probability-1 homotopy algorithms can also be used in situations where extra measurements are available if one uses the least-squares estimate instead of directly converting the measurement. An example of using a (polynomial-only) probability-1 method for solving such a least-squares problem is given in [65] for a problem given in multidimensional polynomial form. However, finite-precision errors and the presence of stiff differential equations can cause certain general problems (not considered here) to be more difficult to solve without resorting to extended-precision arithmetic and more sophisticated differential-equation integration routines.

All together, the combination of a probability-1 homotopy method with a cubature integration technique can form a powerful approach to track target initialization in very difficult scenarios.

**ACKNOWLEDGEMENTS**

This research is supported by the Office of Naval Research through the Naval Research Laboratory Base Program.

**APPENDICES**

**A. FORMULAE FOR THE CUBATURE POINTS**

This appendix summarizes the formulae for the cubature points used in this paper. The general fifth-order cubature points for approximating a $d > 2$-dimensional multivariate standard normal ($\mathcal{N}(0,1)$) distribution given in [14] (but originally taken from [62, pg. 317, No. 5-3]) were used for the 3D localization problem of Section 4 and are reproduced here as

<table>
<thead>
<tr>
<th>Weight ($\omega_i$)</th>
<th>Point ($\xi_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{(d+2)^{d+2}}$</td>
<td>$[\pm a]$</td>
</tr>
<tr>
<td>$\frac{2}{(d-2)^{d+2}}$</td>
<td>$(\pm b, \pm b, \ldots, \pm b)$</td>
</tr>
</tbody>
</table>

The points are given as shown above, where

$$a = \sqrt{\frac{d + 2}{2}} \quad b = \sqrt{\frac{d + 2}{d - 2}} \quad (46)$$

and $d$ is the dimensionality of the points generated. The $\pm$ indicates that all possible combinations of negative and positive elements should be used. The bracket notation for the first set of points indicates that all possible vectors with that single nonzero element should be generated. There are $2d$ points of the first type and $2^d$ points of the second type. These points can be used for integrals involving an arbitrary Gaussian weighting with $d > 2$.

Note that $14$ cubature points are necessary to evaluate a 3D integral. For evaluating a 6D integral, 76 cubature points would be needed. Thus, the alternative set of cubature points given in [14] (but originally from [47]) that require only 57 points were used. These are

<table>
<thead>
<tr>
<th>Weight ($\omega_i$)</th>
<th>Point ($\xi_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{2^{d+1}}$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\frac{d^2(7-d)}{2(d+1)^2(d+2)^2}$</td>
<td>$\pm a_1 c_1$ $\forall 1 \leq i \leq (d + 1)$</td>
</tr>
<tr>
<td>$\frac{2(d-1)^2}{(d+1)^2(d+2)^2}$</td>
<td>$\pm (a_1 + a_2) c_2$ $\forall 1 \leq i &lt; j \leq (d + 1)$</td>
</tr>
</tbody>
</table>

where

$$c_1 = \sqrt{d + 2} \quad c_2 = \frac{1}{2} \left( \frac{d(d + 2)}{d - 1} \right) \quad (47)$$

This work is supported by the Office of Naval Research through the Naval Research Laboratory Base Program.
and there are \( d + 1 \) \( d \)-dimensional \( \alpha_i \) vectors for \( i = 1 \) through \( i = d + 1 \) of the form \( \alpha_i = [\alpha_i(1), \alpha_i(2), \ldots, \alpha_i(d)]' \) with

\[
\alpha_i(k) = \begin{cases} 
\sqrt{d+1} & \text{for } k < i, \\
\frac{d+1}{(d-k+2)(d-k+1)} & \text{for } k = i, \\
\frac{(d+1)(d-i+1)}{d(d-i+2)} & \text{otherwise.}
\end{cases}
\] (48)

There is one point from the first row in the table (the origin), 2\((d+1)\) points for the second row and \(d(d+1)\) points for the third row. These points and weights have been modified from their original form in [47] so that they can be used for integrals involving an arbitrary Gaussian weighting with \( d \gtrsim 4 \).

For the surface-wave localization problem of Section 6, to be able to handle a \( d \gtrsim 2 \)-dimensional standard normal distribution, the arbitrary-order cubature points and weights as described in Appendix D of [18] (originally obtained by combining the linear formula of [31] with the product rule of [62, Ch. 2]) are used. These are found as follows:

Arbitrary-Order Cubature Points and Weights

The procedure is to find an order \( k = 2m - 1 \) cubature formula where \( m \) is an integer.

- First, find a scalar formula.
  Find the eigenvectors and eigenvalues of the \( m \times m \) matrix \( J \)
  \[
  [J]_{ij} = \begin{cases} 
  \sqrt{i} & \text{if } j = i + 1, \\
  \sqrt{j} & \text{if } i = j + 1, \\
  0 & \text{otherwise.}
\end{cases}
\] (49)

using, for example, the \texttt{eig} command in Matlab. Let \( V \) be the \( m \times m \) matrix where each column is an eigenvector and \( D \) be a matrix where the diagonal elements are eigenvalues. The \( i \)-th eigenvalue corresponds to \( \xi_i \), which is the \( i \)-th scalar cubature point, and the corresponding weight for the cubature point is the first element of the corresponding eigenvector \( \tilde{\omega}_i \) (\( \tilde{\omega}_i = V(1,i) \times 2 \) in Matlab notation.)

- Next, find the multidimensional points. The \( j \)-th multidimensional point and its corresponding weight are

\[
\omega_j = \prod_{i=1}^{n} \tilde{\omega}_{c_j(i)} \quad \xi_j = \begin{bmatrix} \tilde{\xi}_{c_j(1)} \\ \vdots \\ \tilde{\xi}_{c_j(d)} \end{bmatrix}.
\] (50)

where the indices \( c_j = [c_j(1), c_j(2), \ldots, c_j(d)]' \) form the \( j \)-th permutation of \( m \) items in \( d \) slots. This begins with \( c_1 = [1, 1, \ldots, 1]' \). Given \( c_1 \ldots c_{d-1} \), \( c_d \) can be found using the following routine:

```matlab
curChoice=1;
c(curChoice)=c(curChoice)+1;
while (c(curChoice)>=m)
c(curChoice:=(i+1))=1;
cChoice=curChoice+1;
c(curChoice)=c(curChoice)+1;
end
```

The above routine should not be passed the last possible value of \( c \) as it will read past the end of the vector.

To make the aforementioned cubature points and weights for the standard normal distribution suitable for use with an arbitrary \( N(\mu, \Sigma) \) distribution, the weights are transformed as

\[
x^{(i)} = \mu + \Sigma \tilde{\xi}_i.
\] (51)

B. Simple Numerical Differentiation

A number of the examples in this paper use models for which it would be difficult to derive explicit derivatives for the homotopy algorithm. Therefore, numerical differentiation is used. This appendix summarizes how central numerical differentiation routines of an arbitrary order can be derived.

In [9, Ch. 4.1], numerical differentiation formulae are derived via Lagrange interpolating polynomials and a number of low-order solutions are provided. The order of the derivative interpolation increases linearly with the number of points used. The solution can be generalized to an arbitrary number of points via Equation (4.2) of [9], which expresses the points as a first derivative of a specific Lagrange interpolating polynomial. For the purpose of this paper, the simple three-point central difference formula

\[
f'(x) \approx \frac{f(x + \epsilon) - f(x - \epsilon)}{2\epsilon}
\] (52)

is used, where \( \epsilon \) is a small quantity. The error in the formula is \( O(\epsilon^2) \).

REFERENCES


[67] L. T. Watson, “A globally convergent algorithm for computing fixed points of $C^2$ maps,” Applied Mathe-


