2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Simultaneous Numerical Optimization for Data Association and Parameter Estimation

Anil Damle* and Lucy Pao*

Abstract— **In some tracking environments, there is no clear linkage of measurements to objects, requiring data association algorithms to associate measurements to target tracks. Further complicating the problem is that targets can operate in different modes and hence have time-varying dynamics. It has recently been shown that treating the potential measurement data for an object as a single continuous signal, that contains impulses when the measurement data jumps to a new object, allows the data association problem to be cast as a trajectory optimization problem. This optimization problem can also be expanded to formulate estimates for dynamic parameters of the objects being measured. The contribution of the current paper is the development of a numerical optimization algorithm that allows the recently developed impulse-based data association and parameter estimation methods to be easily applied without a significant amount of scenario-specific analytical calculations.**

I. Introduction

Target tracking is the process of maintaining state estimates of one or several objects over a period of time. These objects can be aircraft, ships, or ground-based objects. One of the difficulties that sets target tracking apart from other estimation tasks is the uncertainty in measurement origin. When measurements can originate from a number of closely-spaced objects, and there is no clear linkage between measurements received and the objects they represent, a data association algorithm is necessary to associate each measurement to its object of origin [1], [2], [3], [4], [5].

The data association problem considered in this paper arises when measurements are taken of an object of interest that is near other objects with similar dynamics. A sensor is used to obtain a single (possibly noisy) continuous measurement, which may be of the object of interest, but may also inadvertently be of the other nearby objects. In practice, the measurements are taken in discrete time steps at relatively long time intervals (compared to the available computer processing speed available); these discrete measurements can be interpolated to yield a continuous measurement stream. The goal of the data association problem here is to determine over what time intervals does the measurement stream correspond with the object of interest. It is assumed that the measurement stream initially correctly corresponds with the object of interest. In addition to the requirement for data association, the object of interest may operate in different modes, and it is also necessary to estimate the object's dynamical parameters that may change in the hybrid system from one time interval to another. Because measurements arrive infrequently relative to the computational power available, the data association and parameter estimation problem is addressed in batch form on windows of data. This is in contrast to other recursive methods which are capable of providing faster real-time target tracking [1], [2], [3], [4], [5].

Recent work [6], [7], [8], [9] has shown that this data association and parameter estimation problem can be cast as a continuous trajectory optimization problem where impulses are used to represent when the origin of measurements changes from one object to another. By determining when impulses must occur in the state trajectory and what dynamical parameters allow the state trajectory to "fit" the measurement data "well," it is possible to determine which intervals of the measurement stream correspond with the object of interest and what parameters must govern the object dynamics over different time intervals. Because the derivatives of a "standard" cost function have been analytically determined with respect to an arbitrary number of impulse times and magnitudes [7] and dynamical parameter estimates [8], [9], it is possible to implement both first- and second-order optimizations of the cost function.

Because analytically computing the derivatives with respect to all of the impulse times and magnitudes and hybrid system dynamical parameters can be tedious and intractable as the number of dynamical parameters increase, in this paper, we evaluate a strictly numerical optimization scheme that can simultaneously optimize over both impulsive switching times and system parameters. An advantage of a strictly numerical optimization scheme is that it makes the implementation significantly more problem agnostic, allowing this trajectory-optimization based approach for data association and parameter estimation to be easily applied to many scenarios without the need to analytically compute numerous derivatives. While the more analytical approach provides some computational savings in implementation, because the applications of interest are those where measurements can be processed in batch fashion, computational efficiency is slightly less important.

Figure 1 shows an example of measurement data from a one-dimensional system trajectory with a nearby object with similar dynamics. Here the dashed line represents the measurement data gathered. Before time $t = 2$, the measurement signal originates from the object of interest. From $t = 2$ to $t = 4$, the measurement signal originates from another nearby object. After $t = 4$, the measurement signal originates from the object of interest again, where this object may now

This work has been supported in part by the US Air Force under Grant FA9453-08-C-0165.

*Department of Electrical, Computer, and Energy Engineering, University of Colorado, Boulder, Colorado 80309 USA, damle@colorado.edu, pao@colorado.edu

be operating in a different mode with different dynamics. The solid line represents the estimated measurement trajectory based on incorrect system parameters and impulse times. The optimization problem that needs to be solved basically amounts to finding the impulse times and hybrid system parameter estimates so that the solid curve will "fit well" with the measurement data. By solving for the parameters that minimize the difference between these two curves, one can recover the times when measurements jump between objects and develop estimates for system parameters.

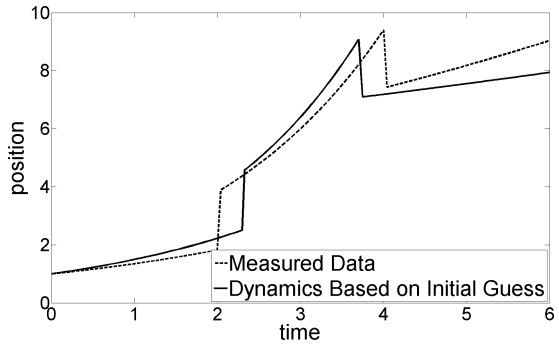

Fig. 1. Example impulsive hybrid system: actual (dashed) and initially estimated (solid) measurement trajectories may be used for switching time and parameter estimation

The remainder of this paper is organized as follows. Section II gives an overview of how the data association problem is cast as a continuous optimization problem, and Section III provides a brief review of previous analytic methods for solving the optimization problem. In Section IV, the case is made for utilizing well known numerical optimization techniques to solve the data association problem. Section V then presents simulated examples along with a comparison to the mostly analytic methods. Finally, Section VI contains concluding remarks.

II. Problem Definition

In order to cast the data association and system parameter estimation problem as an optimization problem, a cost function needs to be developed. To do this, first consider the sampled data to be continuous and denote it as $x_d(t)$ for $t \in [0, t_f]$. If the sampled data is discrete, a continuous $x_d(t)$ can be computed via interpolation. Now define a trajectory x by

$$\begin{aligned} \dot{x} &= f_i(x, \alpha_i, t), \quad \tau_i \leq t < \tau_{i+1}, \\ \tau_1 &= t_0, \\ \tau_{N+1} &= t_f, \\ x(0) &= x_0. \end{aligned} \tag{1}$$

In addition to x having different dynamics in each interval $[\tau_k, \tau_{k+1}]$, impulses are allowed to occur at these so called impulsive switching times. These impulses are described by imposing the conditions $x(\tau_k^+) = x(\tau_k^-) + \delta_k$; note that here the impulse amplitudes δ_k are assumed to be known. Each function f_i also depends on n_i parameters α_i over the interval $[\tau_i, \tau_{i+1}]$.

Notice that x depends on both a set of dynamical parameters and a set of impulsive switching times. Notationally, x will simply be denoted $x(t)$, as carrying around explicit dependence on this large number of parameters would be cumbersome. By assuming that the measurement data represents the true nature of a dynamical system, we can define the cost function

$$J(\cdot) = \int_0^{t_f} \ell(x(s), s) ds \tag{2}$$

where

$$\ell(x(s), s) = (x_d(s) - x(s))^2. \tag{3}$$

By minimizing (2) with respect to the $\sum_1^N n_i$ system parameters and N impulsive switching times upon which x is dependent, good estimates of the true system parameters and impulsive switching times can be obtained. Hence, the total number of parameters that are being optimized over is $N + \sum_1^N n_i$. This formulation simultaneously addresses the problems of determining when measurements jump between objects and what the true parameters of a dynamical object are.

III. Previous Work

Recasting the data association problem as a trajectory optimization problem has been explored previously in [6], [7]. The same type of optimization problem has been addressed by [8], [9]. Specifically, [6] addresses the data association problem where all of the objects have the same dynamics and the optimization only occurs over impulse times; [7] then extends the problem to optimize over unknown impulse amplitudes. The other works [8], [9] use the same approach to address the slightly different problem of computing switching times of a (non-impulsive) hybrid system. In addition, some work has been done with computing switching times and unknown parameters of a hybrid system [10].

All of these previous works have used Newton's Method as the optimization scheme. A key aspect of all these previous works is that all of the derivatives necessary to form the Hessian and gradient are computed efficiently by utilizing a large amount of initial analytic simplification. Using this analytic method has its advantages, including quadratic convergence and reduced computational complexity. However, introducing a potentially large number of unknown parameters to the system makes analytic computation of the necessary elements to use Newton's method unwieldy and adds an element of problem dependence to implementations.

There are two main reasons that we explore the use of a numerical method for this problem. One is rooted in the desire to estimate system parameters. Another reason is to develop a method that can be implemented in a problem agnostic way. Implementation of the more analytic methods previously developed [6], [7], [8], [9] can be very complex and have dependence on the specific problem.

IV. Numerical Approach

A. Motivation

To motivate the use of a numerical scheme, the problem of analytically computing the necessary derivatives of J is first explored. Given that we want to optimize (2), to do so analytically requires the computation of the Hessian of J. Parts of this Hessian, namely $\frac{\partial J}{\partial \tau_i}$ and $\frac{\partial^2 J}{\partial \tau_i \partial \tau_j}$, can be computed in a very computationally efficient way by using some initial analytic work [6], [8], [9].

However, attempting to extend this largely analytic scheme to do simultaneous parameter estimation leads to some complexity issues. To illustrate these issues, a sample computation of $\frac{\partial J}{\partial \alpha_i}$ will be explored. First note that the trajectory can be written as

$$\begin{aligned} x_k(t) &= x_{k-1}(\tau_k^-) + \delta_k \\ &\quad + \int_{\tau_k^+}^{t} f_k(x_k(s), \alpha_k, s)ds \;\; \tau_k \leq t < \tau_{k+1}. \end{aligned} \tag{4}$$

From this it can be seen that $x(t)$ only depends on α_i if $t > \tau_i$.

Now take the derivative of the cost function

$$J(\cdot) = \int_{t_0}^{\tau_2^-} \ell(x(s), s)ds + \cdots + \int_{\tau_N^+}^{t_f} \ell(x(s), s)ds \tag{5}$$

with respect to α_i as

$$D_{\alpha_i} J(\cdot) = \int_{\tau_i}^{t_f} D_1 \ell(x(s), s) D_{\alpha_i} x(s) ds. \tag{6}$$

This means that to compute $\frac{\partial J}{\partial \alpha_i}$ requires the computation of $\frac{\partial x}{\partial \alpha_i}$.

To do this, utilize the integral representation of $x(t)$ and perturb the parameter set α_i in some direction β, and notice the explicit dependence of $x(t)$ on α_i is now present in the notation,

$$\begin{aligned} x_k(t, \alpha_i + \epsilon\beta) &= x_{i-1}(\tau_i^-) \\ &\quad + \int_{\tau_i^+}^{\tau_{i+1}^-} f_i(x_i(s, \alpha_i + \epsilon\beta)), \alpha_i + \epsilon\beta, s)ds \\ &\quad + \int_{\tau_{i+1}^+}^{t} f(x(s, \alpha_i + \epsilon\beta)), s)ds. \end{aligned} \tag{7}$$

Defining $\psi_i = D_{\alpha_i} x$ and $\alpha_i + \epsilon\beta = \tilde{\alpha}_i$, we can take a derivative of (7) with respect to ϵ and get

$$\begin{aligned} D_\epsilon x_k(t, \tilde{\alpha}_i) \circ \beta &= D_1 f_i(x_i(s, \tilde{\alpha}_i), \tilde{\alpha}_i, s)\psi \circ \beta \\ &\quad + D_2 f_i(x_i(s, \tilde{\alpha}_i)), \tilde{\alpha}_i, s) \circ \beta \\ &\quad + \sum_{\{k>i | t>\tau_k\}} D_1 f_k(x_k(s, \tilde{\alpha}_i), s)\psi \circ \beta. \end{aligned} \tag{8}$$

This leads to the ordinary differential equation (ODE)

$$\begin{aligned} \dot{\psi} &= D_1 f_i(x_i(s, \alpha_i + \epsilon\beta), \alpha_i + \epsilon\beta, s)\psi \\ &\quad + \sum_{\{k>i | t>\tau_k\}} D_1 f_k(x_k(s, \alpha_i + \epsilon\beta), s)\psi \\ &\quad + D_2 f_i(x_i(s, \alpha_i + \epsilon\beta)), \alpha_i + \epsilon\beta, s) \circ \beta. \end{aligned} \tag{9}$$

While in principal this ODE could be solved for α_i numerically, a large amount of analytic work is needed to be able to do so. In addition, unlike the case for switching times presented in [6], it is not possible to couple all of the α_i computations into a single numerical integration. Thus, the benefits of reduced computational complexity through preliminary analytic computation are lost. Furthermore, the number of system parameters that are being estimated could be quite large and grow much faster than the number of switching times. Because of these added complexities, a strictly numerical optimization of (2) is explored.

B. Numerical Optimization Scheme

There are a large number of numerical methods for optimization, and it should be noted that other schemes may also perform well on the type of problems presented here. Specifically, a quasi-Newton method is used so that all that is required at each step is a numerical computation of the gradient. Further justification for the use of a quasi-Newton method stems from the use of Newton's method in previous works [6], [7], [8], [9]. These works show that descent methods appear to work well for the problem type being addressed. Specifically, we use the so called BFGS quasi-Newton method (independently attributed to Broyden, Fletcher, Goldfarb, and Shanno) [11], [12], [13], [14]. This method uses the computation of the gradient at each step to generate an approximation to the Hessian matrix and then uses a Newton's method style step. Furthermore, while the convergence is not quadratic, faster than linear convergence is generally achieved.

A brief overview of the BFGS for minimizing $f(x)$ as stated in [15] is given here. Start with an initial guess x_0 and an initial positive definite guess B_0 for the Hessian matrix (here the identity is used). Then, repeat the following until some stopping criteria is reached.

$$\begin{aligned} g_k &= \nabla f(x_k) \\ d_k &= -B_k^{-1} g_k \\ \alpha_k &= \underset{\alpha_k}{\operatorname{argmin}}\; x_k + \alpha_k d_k \\ x_{k+1} &= x_k + \alpha_k d_k \\ y_k &= \nabla f(x_{k+1}) - g_k \\ s_k &= \alpha_k d_k \\ B_{k+1} &= B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}. \end{aligned}$$

The one ambiguity in the BFGS algorithm is the use of a line search to compute $\underset{\alpha_k}{\operatorname{argmin}}\; x_k + \alpha_k d_k$. A large number of possible line search techniques exist and could be used.

The BFGS implementation used here implements a simple backtracking line search, in which you start with some initial guess for how far to move and reduce the guess geometrically until conditions known as the Wolfe Conditions are met for the search distance α_k [16]. These conditions are

$$f(x_k + \alpha_k d_k) - f(x_k) \quad \leq \quad c_1 \alpha_k d_k^T g_k \tag{10}$$

and

$$d_k^T \nabla f(x_k + \alpha_k) \quad \geq \quad c_2 d_k^T g_k, \tag{11}$$

where $c_1 \leq c_2$ are constants that are chosen in the range $(0, 1)$.

Finally, several end conditions are used for the BFGS implementation used here. Specifically, the algorithm is terminated if $J(\cdot)$ is below some value, or if the line search yields a very small movement. Additionally, if $\nabla f(x_k)$ is sufficiently small, and $J(\cdot)$ is not too large, the algorithm is terminated.

C. Computational Complexity

Utilizing this BFGS method for minimization of $J(\cdot)$ requires a large number of computations of $\nabla J(\cdot)$. Because a large number of methods can be used to numerically compute derivatives, the idea of computational complexity is abstracted from the number of function evaluations to the number of gradient computations. Ignoring the line search, the number of gradient computations is equal to the number of iterations, meaning that if the line search is computed efficiently the number of necessary gradient evaluations will be small. Because the Wolfe conditions are used, the line search algorithm requires a gradient computation for each new guess of α_k. By choosing the factor of geometric reduction in the backtracking line search to be large, there should be very few gradient computations per line search. The disadvantage of fast geometric reduction in the backtracking line search is that the overall number of iterations may increase. This is a trade off that is not easily quantifiable, and as such parameters for the line search are best chosen experimentally.

To get at least some sense of the computational complexity of the implementation used here, note that a simple forward difference was used to compute derivatives, so as to keep the number of functional evaluations small. The number of geometric reductions per line search was capped at 20. This was based on the initial distance guess being 1 with a reduction factor of 0.5, so more than 20 steps would yield a very small change for the iteration. This means that the number of gradient computations is at most 20 per iteration. However, as seen in the example below, this maximum is never reached. One last note is that a single functional evaluation requires a numerical solution of $x(t)$.

V. Numerical Examples

A. Noise-free Case

An example problem was run using the aforementioned implementation of the BFGS numerical optimization. Here the dynamics on each section are

$$f_1 \quad = \quad \cos(\alpha_1 x), \tag{12}$$

$$f_2 \quad = \quad \alpha_2 x, \tag{13}$$

$$f_3 \quad = \quad \alpha_3 x + \cos(x). \tag{14}$$

At the impulsive switching times τ_1 and τ_2, there are impulses of magnitudes 4 and -4, respectively. The measured data comes from the parameter values $\tau_1 = 2$, $\tau_2 = 4$, $\alpha_1 = 0.5$, $\alpha_2 = 0.1$, and $\alpha_3 = 0.3$. The BFGS algorithm was run with an initial guess of $\tau_1 = 2.1$, $\tau_2 = 4.2$, $\alpha_1 = 0.4$, $\alpha_2 = 0.2$, and $\alpha_3 = 0.25$. Figure 2 shows the dynamics with the impulses from both the true parameter values, and the initial guess for the parameter values.

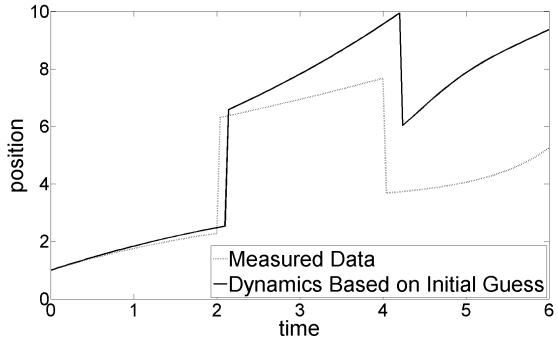

Fig. 2. Example impulsive hybrid system: actual (dashed) and initially estimated (solid) measurement trajectories may be used for switching time and parameter estimation

Using this initial guess, the BFGS implementation was used and converged in 19 iterations with error on the order of 10^{-7}. While this is very quick convergence, the more interesting result is that only 37 gradient computations were used. This means that the line search was fairly efficient at each step. Figure 3 shows the error at each iteration. Consider p as a vector containing the true parameter values, in this case $p = (\tau_1 \quad \tau_2 \quad \alpha_1 \quad \alpha_2 \quad \alpha_3)^T$, and $\tilde{p}$ is a vector containing estimated values of the same parameters. Using this notation the error considered here is $e = ||p - \tilde{p}||_\infty$.

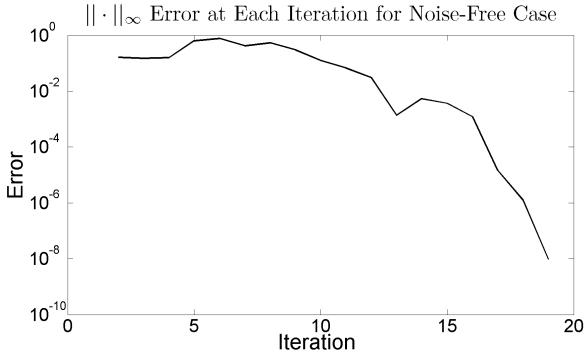

Fig. 3. $||\cdot||_\infty$ error at each iteration for noise-free case

Overall, the method appears to work very well in this example in which the measured data was assumed to contain

no noise. While it is interesting that at some point the error slightly increases before going down again, what is more telling is the fairly rapid convergence of the method once the algorithm gets close to the true parameter values. One possible reason for the error to increase is because the search direction is computed numerically and may be inaccurate. Further, note that here the infinity norm of the error is shown and it is possible that even with an exact gradient the infinity norm of the error could increase after an iteration.

B. Noisy case

A natural question given the method's performance on a noise free example is how will the method perform in the presence of noise. Figure 4 shows the noisy signal that is now considered to be the measurement data. The noise is generated randomly and the noise level is around 5% relative to the true signal.

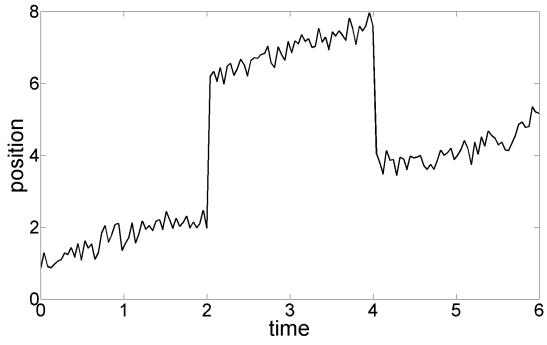

Fig. 4. Noisy example impulsive hybrid system measurement trajectory for switching time and parameter estimation

Given the presence of noise, it is necessary to treat the problem more carefully. Specifically, it is important to notice that noise will affect the estimation of system parameters and switching times differently. Provided that the impulse magnitudes dwarf the noise, a reasonable assumption, it is not expected that the noise will severely impact the computation of the switching times. This is supported by results in [6]. However, system parameters could be more severely impacted by noise. Consider the first region of dynamics in Figure 4, where small changes in the system parameter value governing this region will be masked by the noise. While there may be situations where system dynamics are very sensitive to system parameter changes, in general the assumption here is that the presence of noise will directly impact the accuracy possible in the estimation of system parameters.

Using the same true parameters and initial guess as in the noise free case, the BFGS algorithm was run using the noisy data as the measured data. Because of the aforementioned effects of noise on the system, the error is broken into two parts, error in switching times and error in system parameter estimates. Figure 5 shows the infinity norm of the error for the system parameters and impulse times independently.

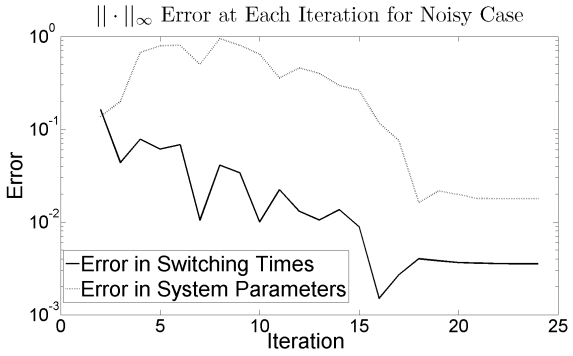

Fig. 5. $||\cdot||_\infty$ error at each iteration for noisy case

The method ran for 24 iterations and required 61 gradient evaluations. While this is more than in the noiseless case, it is not an unreasonable increase considering that 5% noise was added. Further notice that impulse times were more accurately computed than the system parameters, as was expected. Even though the error in impulse times was on the order of 10^{-2}, that is still sufficient given that in this case there were only 25 samples per second, so the achieved accuracy is greater than the distance between samples.

Table I contains a summary of the numerical results from the previous examples. The initial guesses for the switching times contain 5% error, and the inital guesses for the system parameters all have error larger than 16%. Notice that in the noiseless case, all of the parameters are estimated to within $2.4 \times 10^{-6}\%$. In the case with noise, the switching times are estimated to within 0.1% and the system parameters are estimated to within 4%.

C. Large numbers of parameters

We now summarize an example in which there is a larger number of dynamic parameters. Here the dynamics used are

$$f_1 = \alpha_1 x + \alpha_2 \cos(x), \quad (15)$$

$$f_2 = \alpha_3 x + \sin(\alpha_4 x) - \alpha_5 e^{-x} + \alpha_6 \cos(x), \quad (16)$$

$$f_3 = \alpha_7 x - \alpha_8 \sin(x). \quad (17)$$

Initial values of the parameters, $a_1, \ldots, a_8$, were chosen, and then we randomly perturbed by 1%-5% to get an initial guess for use with the optimization scheme. In this example, the method took longer to converge to an error of less than 0.1%, needing 229 iterations, and 416 function evaluations. While this is much longer than before, the dynamics here are much more complicated. Despite the increase in iterations, this experiment does show that numerical optimization may be a viable process for parameter estimation and switching time optimization. In the process of running this experiment, it was observed that the model's sensitivity on parameters impacts the ability of the method to estimate those parameters.

VI. Conclusions

The problem of simultaneous data association and parameter estimation has been addressed through the use of a

TABLE I
NUMERICAL RESULTS

Parameter	τ_1	τ_2	α_1	α_2	α_3
True value	2	4	0.5	0.1	0.3
Initial guess	2.1	4.2	0.4	0.2	0.25
% Error in initial guess	5	5	20	100	16.67
% Error in estimate without noise	1.1×10^{-7}	6.5×10^{-9}	1.9×10^{-6}	5×10^{-7}	2.4×10^{-6}
% Error in estimate with noise	0.04	0.09	3.6	4	3.4

strictly numerical optimization scheme. As evidenced by the presented examples, utilization of a numerical optimization technique for data association is possible. Furthermore, in the case where there are a large number of unknown dynamic parameters to be estimated, the method is preferable to a more analytic approach.

It should be noted that the analysis of this paper is not limited to the numerical scheme used here. In fact, it is entirely possible that other numerical schemes could achieve better results. We simply present the case for the use of a numerical scheme in a data association problem, and assess the performance of one specific scheme on some examples.

The provided examples show that the method performs very well in an ideal situation. Furthermore, when 5% noise is added, the method still provides reasonable results. This is important given the inherent existence of noise in any measurement data.

The previously developed analytic based methods [6], [7] for data association perform very well in the case where it is only desired to compute jumps between objects. As shown in this paper, however, the method does not analytically scale well once system parameter estimation is required. This is a main advantage of using a numerical scheme. While the purely numerical scheme may be more computationally expensive, it scales reasonably well to an increase in the number of parameters. Furthermore, implementation of the purely numerical method is mostly problem agnostic.

REFERENCES

[1] L. Pao and R. Powers, "A comparison of several different approaches for target tracking with clutter," in *Proc. American Control Conference*, vol. 5. IEEE, 2003, pp. 3919–3924.

[2] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *Control Systems Magazine, IEEE*, vol. 29, no. 6, pp. 82–100, 2009.

[3] Y. Shalom and T. Fortmann, *Tracking and data association*, 1988.

[4] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, no. 1, pp. 5–18, 2004.

[5] M. Kalandros, L. Trailovic, L. Pao, and Y. Bar-Shalom, "Tutorial on multisensor management and fusion algorithms for target tracking," in *Proc. American Control Conference*, vol. 5. IEEE, 2004, pp. 4734–4748.

[6] M. Travers, T. Murphey, and L. Pao, "Impulse optimization for data association," in *Proc. IEEE Conference on Decision and Control*, 2010, pp. 2204–2209.

[7] M. J. Travers, T. D. Murphey, and L. Y. Pao., "Impulsive data association with an unknown number of targets," in *Hybrid Systems: Computation and Control*, 2011.

[8] T. Caldwell and T. Murphey, "Switching mode generation and optimal estimation with application to skid-steering," *Automatica*, vol. 47, no. 1, pp. 50–64, 2011.

[9] E. Johnson and T. Murphey, "Second order switching time optimization for time-varying nonlinear systems," in *Proc. IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference*. IEEE, 2009, pp. 5281–5286.

[10] L. Miller and T. D. Murphey, "Simultaneous parameter and state estimation with applications to multiple contact mode systems," preprint, submitted for the *Proc. American Control Conference*, 2012.

[11] C. Broyden, "The convergence of a class of double-rank minimization algorithms 1, general considerations," *IMA Journal of Applied Mathematics*, vol. 6, no. 1, p. 76, 1970.

[12] R. Fletcher, "A new approach to variable metric algorithms," *The Computer Journal*, vol. 13, no. 3, p. 317, 1970.

[13] D. Goldfarb, "A family of variable metric updates derived by variational means," *Mathematics of Computing*, vol. 24, no. 109, pp. 23–26, 1970.

[14] D. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Mathematics of Computation*, vol. 24, no. 111, pp. 647–656, 1970.

[15] Y. Dai, "Convergence properties of the BFGS algorithm," *SIAM Journal on Optimization*, vol. 13, no. 3, pp. 693–701, 2003.

[16] P. Wolfe, "Convergence conditions for ascent methods," *SIAM Review*, vol. 11, no. 2, pp. 226–235, 1969.