A Particle Filter Multi Target Track Before Detect Application: Some Special Aspects

Yvo Boers and Hans Driessen
THALES NEDERLAND
Hengelo, The Netherlands
Email: {yvo.boers,hans.driessen}@nl.thalesgroup.com

Abstract – *In this paper we describe some special aspects of a particle filter multi target track before detect application. Track Before Detect (TBD) for radar, even in a single target setting is a hard nonlinear non-Gaussian tracking problem. Here we will concentrate on a multi target TBD application, where the objects of interest might be closely spaced. We will focus on a typical setting, in which a hard inequality constraint on the target speed naturally arises. We implemented the constraint in two different ways and discuss the effects on detection and accuracy. Furthermore, we give some attention to the important issue of sample size depletion in a particle filter. We show that in the application under consideration the effective sample size may dramatically decrease. This is also illustrated by means of an example. Also, methods to alleviate this effect are discussed.*

Keywords: Particle Filters, Nonlinear Filtering, Target Tracking, Radar, Constraints

1 Introduction

Classical tracking methods take as an input so called plots, that typically consist of range, bearing, elevation and range rate (doppler) measurements, see [1] and [2]. In this classical tracking setting tracking consists of estimating kinematic state properties, e.g. position, velocity and acceleration on the basis of these measurements.

In the classical setup the measurements are the output of the extraction, see figure 1. In this setup there is a processing chain before the tracking, this processing chain consists of a local detection stage, in which thresholding takes place, a clustering stage and an extraction stage, see figure 1.

In the method, that we consider here, we will use as measurements, the raw non-thresholded measurement data, e.g. reflected power, see figure 1. The method is called Track Before Detect (TBD), see also [11], [18] and [19].

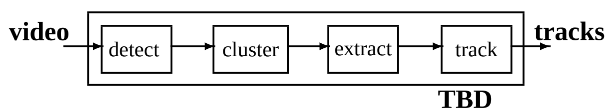

Figure 1: Classical data and signal processing (seperate boxes) and TBD (large box)

If we look at figure 1 we see that in classical tracking (i.e. the separate blocks) a local detection decision is made directly on the basis of the raw measurement from one single scan. This means that already at the beginning of the processing chain a hard decision is made. Note that this decision is made instantaneously, i.e without using information from the (near) past.

In TBD the decision is made at the end of the processing chain, i.e. when all information has been used and integrated over time. Note that information in a track has been obtained by integrating over time). This method is especially suitable for tracking weak targets, i.e. targets that in the classical setting often will not lead to a detection.

In this paper we will show some special aspects of a multi target track before detect application. The multi target TBD application under consideration has been extensively described in [23] and [25]. Here, we will focus on two special topics. The first one is the topic of constraints. We will show that a constraints on parts of the state space naturally enter the problem formulation. We will propose two methods of incorporating the constraints and discuss the advantages and disadvantages of both methods, see also [3]. Furthermore, we will show that the performance of the system is actually enhanced by incorporating these constraints.

The second topic that we will discuss is the topic of the effective sample size, see e.g. [3], [8] and [17]. We will show that in the application at hand the effective sample size may drop dramatically at a Markov jump, i.e. a mode transition. The severity will be shown to depend on the SNR. We will illustrate this and give possibilities to alleviate this effect.

Also in the very recent book on applications of particle filters, see [9], an entire chapter, i.c. chapter 11 is devoted to TBD and its solution by particle filter algorithms.

2 System setup and general particle filter algorithm

We will define here a very general system setup that is used as the core system description for the remainder of this paper.

Consider a general nonlinear jump Markov system:

$$\boldsymbol{s}_{k+1} = f(t_k, \boldsymbol{s}_k, m_k, \boldsymbol{w}_k), \quad k \in \mathbb{N} \tag{1}$$

$$\text{Prob}\{m_{k+1} = i \mid m_k = j\} = [\Pi(t_k)]_{ij} \tag{2}$$

$$\boldsymbol{z}_k = h(t_k, \boldsymbol{s}_k, m_k, \boldsymbol{v}_k), \quad k \in \mathbb{N} \tag{3}$$

Where $\boldsymbol{s}_k \in \mathcal{S} \subset \mathbb{R}^{n_s(d_k)}$ is the base state of the system, $m_k \in \mathcal{M} \subset \mathbb{N}$ is the modal state of the system, $\boldsymbol{z}_k \in \mathbb{R}^p$ is the measurement, $t_k \in \mathbb{R}$ is time, $\boldsymbol{w}_k$ is the process noise and $p_{\boldsymbol{w}(k,d_k)}(w)$ is the probability distribution of the process noise, $\boldsymbol{v}_k$ is the measurement noise and $p_{\boldsymbol{v}(k,d_k)}(\boldsymbol{v})$ is the probability distribution of the measurement noise, f is the system dynamics function, h is the measurement function, $\Pi(t_k)$ is the Markov transition matrix.

Define $\boldsymbol{Z}_k = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_k\}$

The optimal hybrid filtering problem, see [20], can be formulated as follows.

Problem 2.1 (Optimal hybrid filtering problem)
Consider the system represented by the equations (1), (2) and (3). Assume that the initial pdf $p(\boldsymbol{s}_0, m_0)$ is available. The hybrid filtering problem is the problem of constructing the a posteriori pdf

$$p(\boldsymbol{s}_k, m_k \mid \boldsymbol{Z}_k) \tag{4}$$

Particle filter solution

A standard particle filter algorithm for hybrid systems, see [20], will result in an approximation of the a posteriori filtering distribution

$$p(\boldsymbol{s}_k, m_k \mid Z_k)$$

The algorithm from [20], is a 'standard' particle filter implementation, without bells and whistles. Different and more efficient algorithms for (multiple model) particle filters exist, see e.g. [8].

It is known that the standard algorithm guarantees convergence in an *almost sure* sense, of the empirical distribution

$$\hat{p}(\boldsymbol{s}_k, m_k \mid Z_k) := \sum_{j=1}^{N} \frac{1}{N} \delta((\boldsymbol{s}, m) - (\tilde{\boldsymbol{s}}_k^i, \tilde{m}_k^i))$$

to the true, but unknown, a posteriori distribution of interest, i.e. $p(\boldsymbol{s}_k, m_k \mid Z_k)$, if the number of particles, N, tends to infinity. See [7], for more detail and a good overview of convergence results.

3 System setup - application

In this section we will describe the models that will be used in the TBD application. We will describe the system dynamics models and the measurement models. More details on the description can be found in [23] and [25]

Dynamics:

A widely used model is the constant velocity model, see e.g. [1, 2]. This model is used to describe the position and velocity using Cartesian coordinates. Furthermore the model has an additive process noise term. The discrete-time system dynamics of this model is of the form:

$$\boldsymbol{s}_{k+1} = f(t_k, \boldsymbol{s}_k, m_k) + g(t_k, \boldsymbol{s}_k, m_k)\boldsymbol{w}_k \tag{5}$$

where

$$f(t_k, \boldsymbol{s}_k, m_k) = \begin{pmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \boldsymbol{s}_k \tag{6}$$

with the state vector $\boldsymbol{s}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T$ where x_k and y_k are the positions and $\dot{x}_k$ and $\dot{y}_k$ are the velocities. The process noise w_k is assumed to be standard white Gaussian noise. T is the revisit time, which has been assumed to be constant here, this assumption can be relaxed.

The process noise input model is given by

$$g(t_k, \boldsymbol{s}_k, m_k) = \begin{pmatrix} \frac{1}{2}(\frac{1}{3}a_{x,max})T^2 & 0 \\ 0 & \frac{1}{2}(\frac{1}{3}a_{y,max})T^2 \\ \frac{1}{3}a_{x,max}T & 0 \\ 0 & \frac{1}{3}a_{y,max}T \end{pmatrix} \tag{7}$$

with maximum accelerations $a_{x,max}$ and $a_{y,max}$.

Measurements:

The measurements are measurements of reflected power. One measurement $\boldsymbol{z}_k$ consists of $N_r \times N_d \times N_b$ power measurements z_k^{ijl}, where N_r, N_d and N_b are the number of range, doppler and bearing cells. In figure 2 we have plotted power measurements for a fixed bearing angle, but as a function of different range and doppler. The power measurements in this figure correspond to a target that has a SNR of 13 dB. The power measurements per

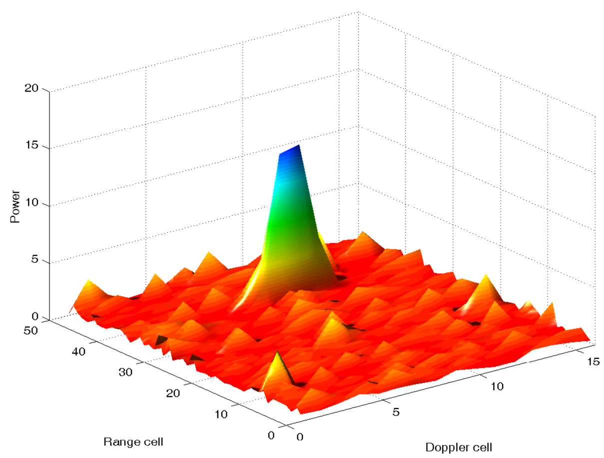

Figure 2: Power of a target in noise (SNR 13dB) as a function of range and doppler cells

range-doppler-bearing cell are defined by

$$\boldsymbol{z}_k^{ijl} = |\boldsymbol{z}_{A,k}^{ijl}|^2 \qquad k \in \mathbb{N} \tag{8}$$

where $\boldsymbol{z}_{A,k}^{ijl}$ is the complex amplitude data of the target which is

$$\boldsymbol{z}_{A,k} = A_k h_A(\boldsymbol{s}_k, t_k) + \boldsymbol{n}(t_k), \qquad k \in \mathbb{N} \tag{9}$$

where $\boldsymbol{n}(t_k)$ is complex Gaussian noise.

$$A_k = \tilde{A}_k e^{\imath \phi_k}, \qquad \phi_k \in (0, 2\pi) \tag{10}$$

is the complex amplitude of the target and $h_A(\boldsymbol{s}_k, t_k)$ is the reflection form that is defined for every range-doppler-bearing cell by

$$h_A^{ijl}(\boldsymbol{s}_k, t_k) = e^{-\frac{(r_i - r_k)^2}{2R} L_r - \frac{(d_j - d_k)^2}{2D} L_d - \frac{(b_l - b_k)^2}{2B} L_b}, \tag{11}$$

$i = 1, \ldots, N_r,\ j = 1, \ldots, N_d,\ l = 1, \ldots, N_b$ and $k \in \mathbb{N}$

with

$$r_k = \sqrt{x_k^2 + y_k^2} \tag{12}$$

$$d_k = \dot{r}_k = \frac{1}{\sqrt{x_k^2 + y_k^2}} (x_k \dot{x}_k + y_k \dot{y}_k) \tag{13}$$

$$b_k = \arctan\left(\frac{y_k}{x_k}\right) \tag{14}$$

which are the range, doppler and bearing respectively of the target. R, D and B are constants, related to the size of a range, a doppler and a bearing cell. L_r, L_d and L_b represent constants of losses.

This setup is easily extended to a multi target setup, see [23] and [25] for more details. In the proceeding we identify three modes: mode 0, i.e the mode corresponding to no target present, mode 1, the mode corresponding to one target present and mode 2, the mode corresponding to two targets present. The Markov matrix for the mode transitions is given by

$$\Pi(t_k) = \begin{pmatrix} 0.90 & 0.10 & 0.00 \\ 0.10 & 0.80 & 0.10 \\ 0.00 & 0.10 & 0.90 \end{pmatrix}$$

4 Dealing with constraints

In many target tracking applications constraints on the state (or part of the state) of the system are imposed. Examples are:

- Tracking of ships in a coastal environment, using airborne sensors. The ship positions are limited to that part of the state space, representing the sea, see e.g. [3].
- GMTI tracking. Targets might be forced to stick to a road, e.g. in case of a bridge, this imposes hard constraints on the target position, see [5] and [6].
- Incorporation of flight envelope information into the tracking, see [4].
- Tracking moving launch platforms, e.g. a fighter or helicopter with the capability of firing a missile. In this case it is realistic to impose that the radial inbound velocity of the missile is bigger than that of the platform.

All the above examples are examples of applications, in which inequality constraints arise. In this paper the application will be of the fourth type. An example, outside the field of target tracking, of dealing with inequality constraints in a dynamical system is found in [16].

In a general form an inequality constraint on the state can be imposed by:

$$\boldsymbol{G}(\boldsymbol{s}_k, m_k, t_k) \leq 0 \tag{15}$$

where $\boldsymbol{G}$ is a vector valued function that might be mode and time dependent.

It is well known, that a constraint of the form (15) cannot be incorporated into a classical (Kalman) filter algorithm. However, the constraint (15) represents *a priori* information that one should exploit.

A particle filter can deal with inequality constraints in an easy and straightforward manner. Care has to be taken with the implementation of constraints, in order to assure a high enough number of effective samples after the constraint has been enforced upon the system. Some information and discussion on how to use constraints and how to implement them, can be found in [3].

We will illustrate the use of a constraint by means of a multi target track before detect example, see [23] and [25] for the multi target modelling details. We deal with a target that flies radially inbound, e.g. a fighter and that has the capability to fire a weaker secondary target, e.g. a missile at the observer. As was already stated in [23] and [25], the early detection and tracking of the primary and secondary target is crucial.

Given the fact that we expect a missile being launched from the moving platform, i.e. the fighter, we may pose a constraint upon the multi target state, e.g. imposing the radial inbound speed of the secondary target being bigger than the radial inbound speed of the primary target, i.e. the platform. Basically, this constraints represents the fact that the missile being fired from the fighter cannot be fired 'backward'. The constraint has been implemented in two

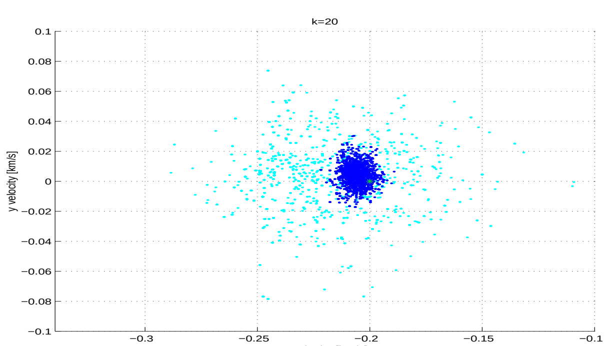

Figure 3: Target Velocity, Particles at time step 20

ways. The first way is by means of the likelihood of the

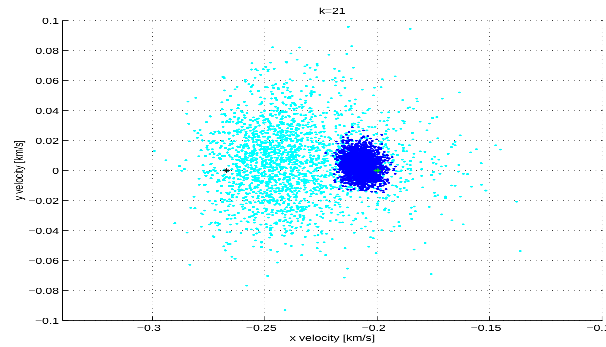

Figure 4: Target Velocity, Particles at time step 21

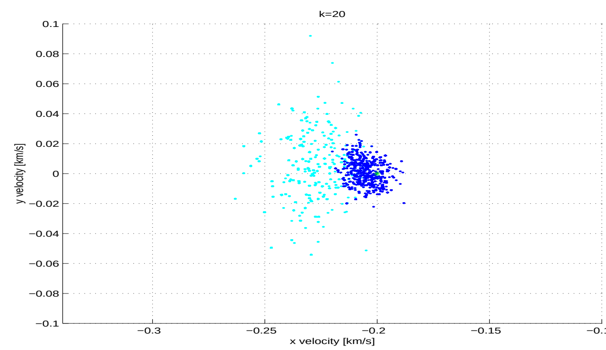

Figure 5: Target Velocity, Particles at time step 20 - Constraint incorporated

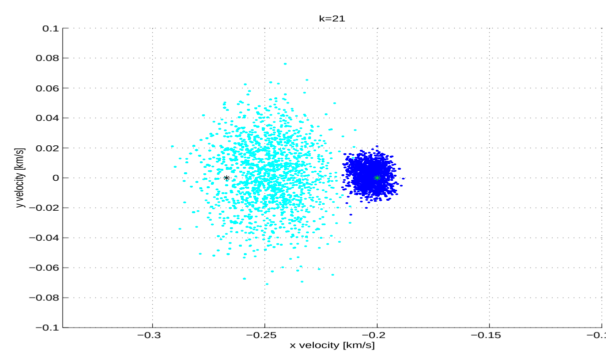

Figure 6: Target Velocity, Particles at time step 21 - Constraint incorporated

system. This basically means that the weight of a particle that does not meet the constraint is set equal to zero. The advantage of this first method is that it is generally implementable. The disadvantage is that the 'effective number of particles' is reduced, we will come back to this later.

The second way is through the process noise, this implementation has the (numerical) advantage that it does not suffer from the decrease of the effective number of particles. The disadvantage is that it might be potentially diffi-

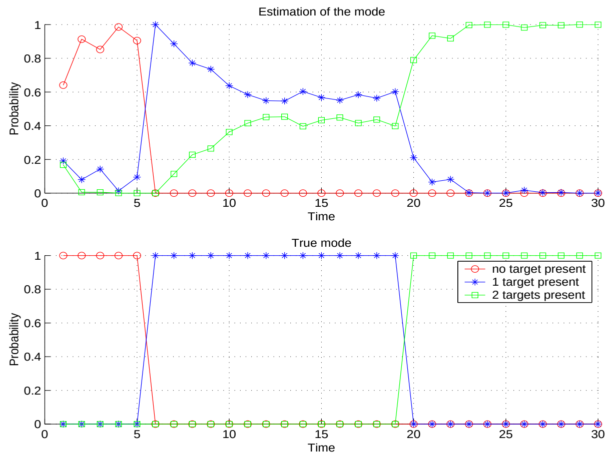

Figure 7: Mode probabilities

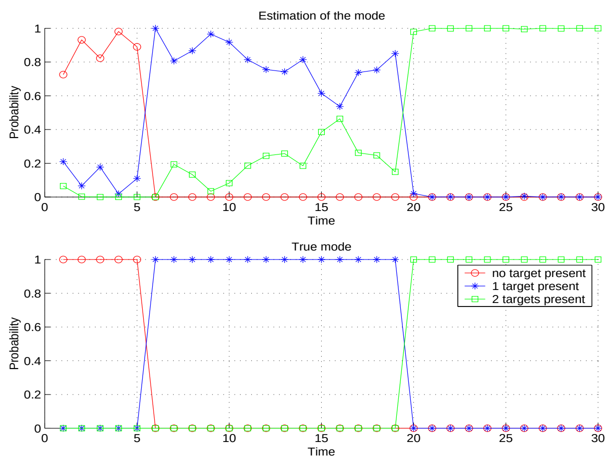

Figure 8: Mode probabilities - Constraint incorporated

cult to generate samples from a non-standard distribution.

We have simulated the multi target system, where the exact same setup as in [23] and [25] has been used. The number of particles is 3000. The SNR of the primary target has been set to 10dB and the SNR of the secondary target has been set to 7dB.

The dynamics of both targets are captured by a constant velocity model, but with different values for their maximal acceleration. For the first target we set $a_{max,x} = a_{max,y} = 5ms^{-2}$. For the second target we choose $a_{max,x} = 35ms^{-2}$ and $a_{max,y} = 5ms^{-2}$.

Until time step 5 no targets are present. At time step 5 the primary target appears and after 20 time steps the secondary target separates from the first one. The targets are flying radially inbound at about 80km from the radar, the first target moves at $200ms^{-1}$ and the secondary target accelerates from $200ms^{-1}$ to $300ms^{-1}$ in three time steps.

In the first simulation series the constraint has not been incorporated, but in the second series we have incorporated the constraint on the radial velocity of the secondary target, through incorporation of the constraint by means of the likelihood.

In the figures 3 and 4 we see the velocity part of the particle cloud for the multi target state, for the case of no constraints. The fact that the constraint has not been applied is seen by the light (blue) colored particles to the right of the dark (blue) ones. These particles correspond to multi target state hypotheses in which the secondary target has a smaller radial inbound velocity than the primary one or stated otherwise they correspond to the hypotheses that the missile has been fired 'backwards'.

In the figures 5 and 6 we see similar clouds. The only difference is that the particles for which the radial inbound velocity of the primary target is higher than the radial inbound velocity of the secondary target have been discarded, i.e. their likelihoods or weights have been set to zero, such that they are not re-sampled. Visually this can be seen in the figures by the fact that there are no light blue colored particles to the right of the dark blue colored ones. By implementing the constraint we are forcing the particles to 'look ahead' for a secondary target and we do not spend effort on picking up a secondary target 'behind' the moving platform. By comparing figures 7 and 8, we see that this strategy pays off, in the sense that the filter is better able to 'detect' the secondary target when the constraint has been enforced.

As we already mentioned before, care must be taken w.r.t. the effective sample size, when incorporating a constraint through the likelihood. If one incorporates a constraint by setting the likelihood or weight of a particle equal to zero, the effective sample size

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (q_k^i)^2} \quad (16)$$

where q_k^i are the normalized particle weights. Obviously the 'best value' is attained when all particles have weight $\frac{1}{N}$ and the 'worst value' is obtained when all, but one, particles have weight zero. As the implementation of a constraint through the likelihood, implies that the weights of constraint violating particles are set to zero, the effective sample size is being reduced.

We have also implemented the constraint through the process noise density function. This implementation has been performed by drawing samples from the process noise for the secondary target in the radial direction according to a one sided density. Thus, only samples with a positive inbound radial acceleration are generated. This guarantees that the missile (i.e. the secondary target) is not fired backwards. W have as well incorporated a history into this density, namely that this one sided density is chosen for particles that are (obviously) in mode 2, but no longer than three time steps. So the process noise model of a particle that has been in mode two for a time period longer than three time steps is set (back) to a 'standard' two sided density function.

We have performed simulations for this setup as well. The results for the 'likelihood method' and the 'process noise method' are similar for 'large' numbers of particles (say 3000) but as the number of particles decreases the 'likelihood method' becomes numerically unstable, where the 'process noise' method remains stable. In fact 1000 particles are sufficient to stably run the filter with the settings defined above, see figures 9 and 10 as well as figure 11 for details.

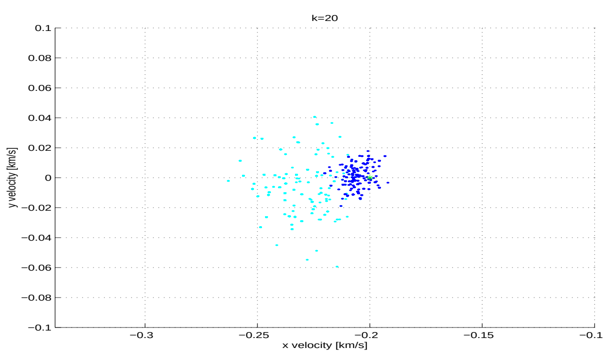

Figure 9: Target Velocity, Particles at time step 20 - Constraint incorporated (through process noise)

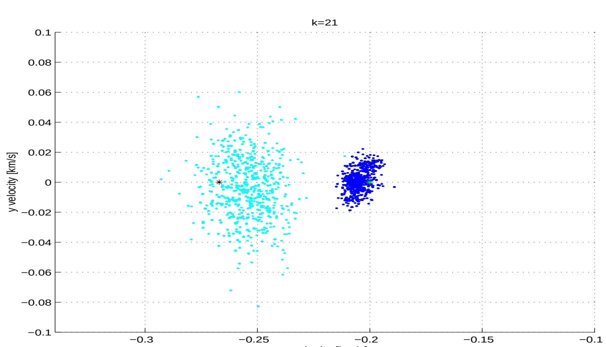

Figure 10: Target Velocity, Particles at time step 21 - Constraint incorporated (through process noise)

An additional important issue is the filter accuracy. In figure 12 we see the radial velocity estimation accuracy of

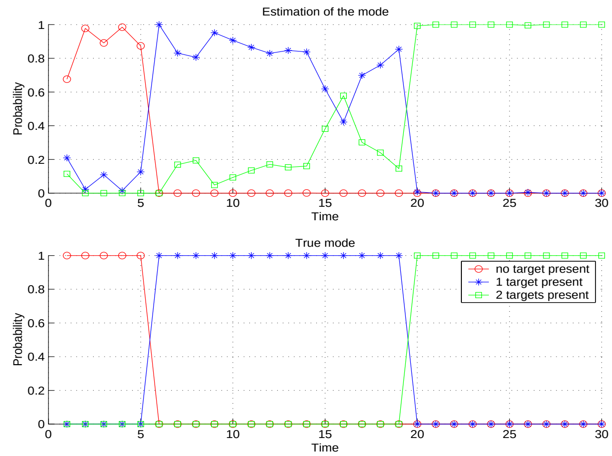

Figure 11: Mode probabilities - Constraint incorporated (through process noise)

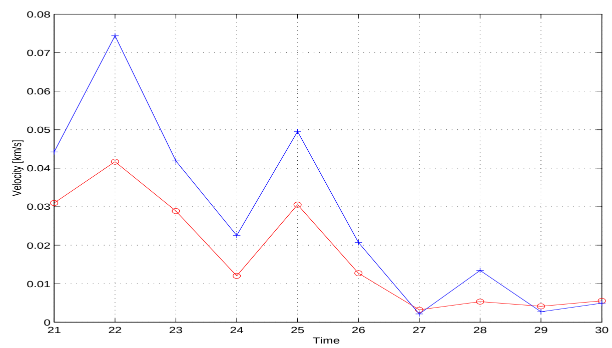

Figure 12: Radial Velocity Estimation Accuracy for Secondary Target: Unconstrained (+), Constraint incorporated (through process noise) (o)

the secondary target, for the case in which the constraint has been incorporated and the case in which the constraint has not been incorporated. We observe quite some gain in the radial velocity estimation accuracy, due to the incorporation of the constraint. For the position accuracy the results were not as dramatic, i.e. not much was gained through incorporation of the constraint.

5 The effective sample size

In this section we will have a more detailed discussion on the issue of the effective sample size. This topic needs special attention when using a particle filter for a jump Markov system.

As an illustration of the 'potential danger' of the decrease of the effective sample size, look at figure 13. We observe from this figure that the effective sample size is only two at a certain point, namely at the time step, where the first target appears. Just after the mode transition, almost all particles are in mode zero, i.e. the non-target mode, when the target appears almost no particles, but those already in mode one, the one target mode, result in a likelihood that is significantly different from zero. A similar effect, although not as dramatic, is also seen at time step of the other mode transition, i.e. the time step where the secondary target appears. Here the effective sample size is 25. The effect is not so severe at this transition, because already a considerable number of particles 'live' in mode two, i.e. the two target mode.

The above phenomenon is well known and is potentially present in any standard jump Markov multiple model particle filter application, see [15] and [20]. This phenomenon leads to numerical instability, i.c. in the above example, at a certain time instant the entire multi target state probability density is represented effectively by only two particles. It is obvious that this number is too low to be useful to make confident inferences about the multi target state.

A possible solution would be to increase the total number of particles. Assume that the minimum number of effective particles in this application would be 50, than for the time step, where the effective number is 2 the total number needs to be increased to a value of 25000. This approach is not practical and computationally very expensive.

Another approach is the so called particle filter controller and has been introduced in [3], but also this method would result in a considerable boost of the total number of particles.

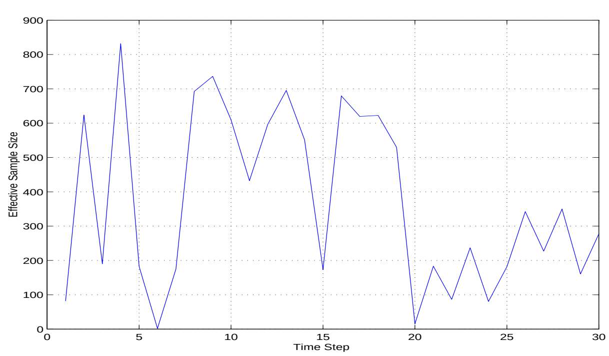

Figure 13: Effective Sample Size - Constraint incorporated (through process noise)

Observe that a more difficult scenario, in terms of SNR, alleviates the effect of sample impoverishment, i.e. small effective sample size, in this example. The reason for this is that in such a relative low SNR setting, the filter will already have a considerable amount of particles in the mode that represents the 'one target present' situation, this is of course, because the distinction between the two modes is less clear than in the case of a higher SNR.

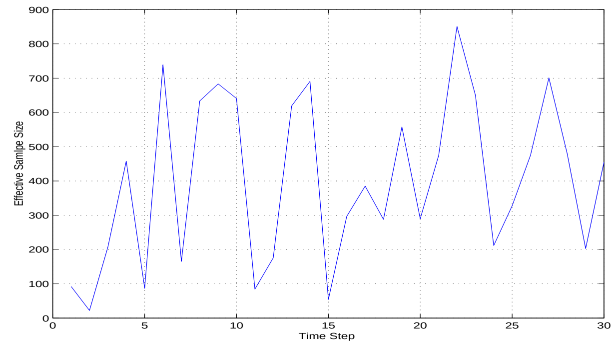

Figure 14: Effective Sample Size - Constraint incorporated (through process noise) - SNR=2

In figure 14, we see the effective sample size for a run, where the SNR for the primary target was equal to two and for the secondary target the SNR was equal to one. Now, at the time step of the first transition, the effective sample size is equal to 87, note that it was 2 in the situation, where the SNR of the primary target was 10. Of course, in this second example the price to pay for the a better and numerically more stable filter is a deteriorated filter performance in terms of accuracy. This is not a desirable situation.

A theoretically sound way to circumvent these problems with the effective sample size for multiple model systems is described in [26]. In [26] a (constant) number of particles, chosen by the designer, is used in each mode without distorting or violating the jump Markov modelling assumption.

6 Conclusions

We have considered a multi target TBD setup and solved it by means of a particle. We have given specific attention to the incorporation of constraints into the filter. The constraints have been incorporated in two different ways. The implementation through the process noise method turned out to be better from a numerical point of view. We have shown the effect of the incorporation of the constraint on the tracking accuracy. Furthermore, we have discussed the matter of sample depletion or impoverishment in a jump Markov system and we have indicated how to solve this problem.

References

[1] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, MA, 1999.

[2] Y. Bar-Shalom and X.R. Li. *Estimation and Tracking Principles, Techniques and Software*, Artech House, Norwood, MA, 1993.

[3] R. Karlsson. *Simulation Based Methods for Target Tracking*, Licencate Thesis No. 930, University of Linköping, 2002.

[4] S. Challa and N. Bergman. Target tracking incorporating flight envelope information. *In Procedeedings of FUSION 2000*, Paris, France, July, 2000.

[5] S. Arulampalam, N. Gordon, M. Orton and B. Ristic. A Variable Structure Multiple Model Particle Filter for GMTI Tracking. *In Procedeedings of FUSION 2002*, Annapolis, MD, July, 2002.

[6] M. Mallick (organizer), Ground Target Tracking. Organized Session at FUSION 2003, Cairns Australia, July, 2003.

[7] D. Crisan and A. Doucet. A Survey of Convergence Results on Particle Filtering Methods, *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 736-746, 2002.

[8] A. Doucet, J.F.G. De Freitas and N.J. Gordon, Eds. *Sequential Monte Carlo Methods in Practice* Springer-Verlag, New York NY, 2001.

[9] B. Ristic, S. Arulampalam and N. Gordon, *Beyond the Kalman Filter - Particle Filters for Tracking Applications*, Artech House, Boston, MA, 2004.

[10] N.J. Gordon, D.J. Salmond and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *Proceedings IEE. F*, vol. 1400, no. 2, pp. 107-113, 1993.

[11] D.J. Salmond and H. Birch. A Particle Filter for Track-Before-Detect, *In Proceedings of the American Control Conference*, Arlington, VA, June 25-27, 2001.

[12] D.J. Ballantyne, H.Y. Chan and M.A. Kouritzin. A Branching Particle-Based Nonlinear Filter for Multi-Target tracking. *In Proceedings of the FUSION 2001 Conference*, Montreal, Canada, August 7-10, 2001.

[13] J.H. Zwaga, J.N. Driessen and W.J.H. Meijer. Track-Before-Detect for Surveillance Radar: a Recursive Filter Based Approach. *In Proceedings of the Small Targets Conference at SPIE AEROSENSE 2002*, Orlando, Fl., April 2-5, 2002.

[14] C. Kreucher, K. Kastella, and A.O. Hero III. Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probabiltiy Density. *In Proceedings of the Small Targets Conference at the SPIE Annual Meeting 2003*, San Diego, CA., August 2-4, 2003.

[15] S. McGinnity and G.W. Irwin. Multiple Model Bootstrap Filter for Maneuvering Target Tracking. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 3, pp. 1006-1012, 2000.

[16] A. Lj. Juloski, W.P.M.H. Heemels, Y. Boers and F. Verschure. Two Approaches to State Estimation for a Class of Piecewise Affine Systems. In Proceedings of *42th IEEE Conference on Decision and Control*, Maui, HI, December 9-12, 2003.

[17] Y. Boers. On the number of samples to be drawn in particle filtering. *In proceedings of the IEE colloquium on Target Tracking*, London, UK, 1999.

[18] Y. Boers and J.N. Driessen. Particle Filter Based Detection For Tracking, *In Proceedings of the American Control Conference*, Arlington, VA, June 25-27, 2001.

[19] Y. Boers, J.N. Driessen and K. Grimmerink. Particle filter based detection schemes. *In Proceedings of the Small Targets Conference at SPIE AEROSENSE 2002*, Orlando, Fl., April 2-5, 2002.

[20] Y. Boers and J.N. Driessen. Hybrid state estimation: a target tracking application. *AUTOMATICA*, vol. 38, no. 12, pp. 2153-2158, 2002.

[21] Y. Boers and J.N. Driessen. A Particle Filter Based Detection Scheme. In *IEEE Signal Processing Letters*, vol. 10, no. 10, pp. 300-302, 2003.

[22] Y. Boers and J.N. Driessen. Particle Filter Based Track Before Detect Algorithms. *In Proceedings of the Small Targets Conference at the SPIE Annual Meeting 2003*, San Diego, CA., August 2-4, 2003.

[23] Y. Boers, J.N. Driessen, F. Verschure, W.P.M.H. Heemels and A. Juloski. A Multi Target Track Before Detect Application. On the CDROM of the *IEEE Conference on Computer Vision and Pattern Recognition - Workshop on Multi Object Tracking*, Madison, WI, June 16-22, 2003.

[24] Y. Boers and J.N. Driessen. Interacting Multiple Model Particle Filter. *IEE Proceedings Radar, Sonar and Navigation*, vol. 150, no. 5, pp. 344-349, 2003.

[25] Y. Boers and J.N. Driessen. A Multi Target Particle Filter Track Before Detect Application. Submitted to *IEE Proceedings Radar, Sonar and Navigation*, 2004.

[26] J.N. Driessen and Y. Boers. An efficient particle filter for nonlinear jump Markov systems. In proceedings of the *IEE Seminar Target Tracking: Algorithms and Applications*, Sussex, UK, March 23-24, 2004.