Proceedings Paper:

Leon, C.M., Mihaylova, L.S. orcid.org/0000-0001-5856-2223 and Driessen, H. (2017) Tracking of Interacting Targets. In: 2017 20th International Conference on Information Fusion (Fusion). 2017 20th International Conference on Information Fusion, 10-13 Jul 2017, Xi'an, China. IEEE . ISBN 978-0-9964-5270-0

https://doi.org/10.23919/ICIF.2017.8009855

Tracking of Interacting Targets

Carlos Moreno Leon⋆, Lyudmila S Mihaylova†, Hans Driessen⋆
⋆Thales Nederland B.V. - Sensors Development System Engineering, Hengelo, The Netherlands
Email: {carlos.morenoleon, hans.driessen}@nl.thalesgroup.com
†Automatic Control and Systems Engineering, University of Sheffield, UK
Email: l.s.mihaylova@sheffield.ac.uk

***Abstract*—In this paper we present a method for the tracking of interacting targets disregarding whether or not the targets are close to each other. The method relies on parametric modeling of assumptions about targets interactive motion. Our filtering solution incorporates the parameters of the model in the state vector to perform on-line parameter estimation and exploitation. The proposed method is applied in a simulated Multiple Target Tracking application with radar track-before-detect measurements. Numerical experiments show that this approach results in estimation error reduction, allows detection of interactive target behaviors and reduce labeling uncertainty in closely-spaced targets tracking.**

***Keywords*—multi-target tracking (MTT); interacting targets; coordinated motion modeling; interactive behavior detection; closely-spaced targets; labeling uncertainty; track-before-detect measurements; Bayesian estimation; particle filtering.**

I. Introduction

Multiple Target Tracking (MTT) refers to the problem of estimating the state of targets in the scene and it finds its application for instance in maritime traffic monitoring and camera surveillance. Many MTT techniques exist, among which Joint Probabilistic Data Association (JPDA) [1] and Multiple Hypothesis Tracking (MHT) [2] are probably the most well-known. Most of these techniques assume independent motion of the various targets.

This paper deals with the problem of modeling and estimating interacting targets motion. Interacting and even coordinated motion can for instance be found in objects that accidentally run into each other, groups of objects that intentionally move closely together as a group, and even well-separated objects that are executing a joint plan. In these scenarios, incorporation of interactive behavior modeling in the estimation algorithms is expected to improve tracking performance.

Some literature exists on group tracking, which is certainly a particular case of interacting targets tracking. In [3], a framework based on the so called "Evolving Networks" is proposed to perform targets state estimation and group structure discovery. The method relies on a graphical network-like representation where each node represents a different target. When the targets are considered in the same group the evolution model becomes coupled along the coordinates of different objects. However, although it is fairly understandable that different objects traveling in a group are interacting with each other, the method does not cover any kind of interactions between non-closely-spaced targets.

The method presented in this paper is motivated by the idea that all possible interacting behaviors can be modeled relying on two different sources of information. First, the set of points that the targets get attracted or repulsed to. Second, the models of interaction between the targets and these points of attraction/repulsion. By these means, any interacting behavior can be understood as variable accelerations of the targets in relation with the set of attraction/repulsion points. These unknown points can be variable in number, fixed or dynamic, and may even be another target in track.

A complete solution would be the one estimating the attraction/repulsion points, the parameters of interaction between the targets and these points, and the dynamics of the targets, all together in the joint space. In this paper we consider the tracking of interacting objects given that the points of attraction/repulsion are known. Therefore, the focus is on the parametric modeling and estimation of the attraction /repulsion accelerations to these points.

In a sense, we model target interactions by using a very simplified version of the so-called Social Force models applied separately for each direction. Social force models, introduced by Helbing [4], have been used extensively in pedestrian tracking. In these applications, social forces are proven to effectively reduce data association errors as in [5]. Several challenges as handling occlusions [6] or tracking with cameras with no overlapping fields of view [7] have been tackled by using these simple yet powerful models.

The results presented in this paper show the benefits of incorporating estimation and exploitation of objects interactions in an MTT application. First, reduction of estimation errors in the filtering process. Second, detection of changes in the behavior of the targets. Third, reduction of labeling uncertainty in closely-spaced targets tracking. Finally, aid in classification of targets behavior. The experiments are in the context of track-before-detect (TBD) radar measurements. Sequential Monte Carlo methods are selected for the implementation due to the highly non-linear nature of the dynamics and the measurement model.

The paper is organized as follows. In Section II we present the assumptions of the interacting target motion model in continuous time and derive the discretized model. In Section III we discuss the incorporation of the interactive discretized model in a MTT application. First, the measurement model is defined and then, it is put together with our interacting target motion model in the context of recursive Bayesian estimation. In Section IV we present the simulation examples. First, we provide some preliminary results on estimation error reduction when incorporating knowledge about objects interaction. Second, we show that a Multiple Model (MM) filtering implementation encapsulating our method allows detection of switching behaviors. Moreover, we show that labeling uncertainty can be reduced by incorporation of interactive motion modeling. We conclude in section V.

II. Interacting Target Motion Modeling

Independent target motion models are oftentimes used for the tracking of any target disregarding its degree of interaction. For the case of one object moving in 2D, the state vector $s(t)$ comprises two coordinates (x and y) and for each coordinate we consider two dimensions: position ($x(t)$, $y(t)$) and velocity ($\dot{x}(t)$, $\dot{y}(t)$). Then $s(t) = [x(t)\ \dot{x}(t)\ y(t)\ \dot{y}(t)]^T$.

A generic continuous-time dynamic model can be represented by a first order linear differential stochastic system as,

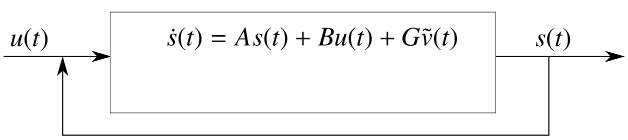

Fig. 1. Generic first order linear dynamic system

where A is called the *system matrix*, B is the (continuous-time) *input gain*, G is the (continuous-time) *noise gain*. $s(t)$ is the state vector, $u(t)$ is the input vector and $\tilde{v}(t)$ is the (continuous-time) process noise. The Continuous White Noise Acceleration model [8] is a widely used independent target motion model. It is characterized by having zero input contribution and matrices,

$$A = diag(A_1, A_2, ..., A_N) \tag{1}$$

where N is the total number of coordinates,

$$A_n = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}; n = 1, ..., N \tag{2}$$

$$G = diag(G_1, G_2, ..., G_N) \tag{3}$$

$$G_n = \begin{pmatrix} 0 \\ 1 \end{pmatrix}; n = 1, ..., N \tag{4}$$

being $\tilde{v}(t)$ continuous-time Standard White Gaussian.

In some more elaborated models the input contribution is considered. When the input vector is fully known, it only causes an explicit time dependency in the deterministic part of the evolution model. However, in Multiple Target Tracking (MTT) applications, the control signals applied on each target are seldomly modeled (if ever) by the tracking system. Therefore, the control vector causes also a dependency in the random component $\tilde{v}(t)$. As we will see in Subsection II-A, the unknown control vector will be considered part of the state vector.

A. Assumptions on targets interaction

Interacting targets perform accelerations depending on the dynamics of other objects and the environment. Therefore, independent target motion modeling is not optimal when tracking interacting targets as coupled accelerations get overlooked. In this Subsection our parametric model of targets interaction is presented.

In this paper we consider a toy scenario where two objects (labeled as "b" and "r") move in 2D. Then, we consider the state vector $s(t) = [s^b(t)\ s^r(t)]^T$. Furthermore, we assume that the target labeled as "b" is the single point of attraction/repulsion.

When coordinates of different objects are assumed to be coupled obtaining an explicit expression of the model in discrete time is not straightforward. Special considerations have to be taken as the calculation of the discrete transition matrix and the covariance of the discrete process noise cannot be decomposed at the coordinate level. Therefore, interacting assumptions need to be defined in continuous time and the discretization of the propagation model needs to be worked out. The assumptions about the structure of the interactive motion are,

$$\ddot{x}^r(t) = k_1(x^b(t) - x^r(t)) - k_2\dot{x}^r(t)$$
$$\ddot{y}^r(t) = k_1(y^b(t) - y^r(t)) - k_2\dot{y}^r(t) \tag{5}$$

with k_1 and k_2 assumed constants with units $[s^{-2}]$ and $[s^{-1}]$ respectively.

The model basically represents that the accelerations performed by the object labeled as "r" are the result of combining two aspects. These are: the desire of target labeled as "r" to interact with the target labeled as "b" and the maneuvering limitations of the target labeled as "r" due to inertia. In this particular case only two parameters are needed (k_1 and k_2). We consider the state vector ordered in the following manner, $s(t) = [s_x(t)\ s_y(t)]^T$ where,

$$s_x(t) = [x^b(t)\ \dot{x}^b(t)\ x^r(t)\ \dot{x}^r(t)]$$

$$s_y(t) = [y^b(t)\ \dot{y}^b(t)\ y^r(t)\ \dot{y}^r(t)]$$

The assumptions in (5) can be directly incorporated as inputs in the generic model from Fig. 1. Choosing to do so allows us to consider the input as a good representation for the intention of the pilot controlling the target. Is this case, matrix B is given by,

$$B = diag(B_n, B_n), \tag{6}$$

$$B_n = (0\ 0\ 0\ 1)^T \tag{7}$$

and,

$$u(t) = Ks(t), \tag{8}$$

where,

$$K = diag(K_n, K_n), \tag{9}$$

and

$$K_n = (k_1\ 0\ -k_1\ -k_2) \tag{10}$$

By these means, we have modeled accelerations depending on control decisions (which ultimately depend on the dynamics of other objects and the environment). Now, we can put this model for the accelerations together with a Continuous White Noise Acceleration model,

$$\dot{s}(t) = As(t) + BKs(t) + G\tilde{v}(t) = (A + BK)s(t) + G\tilde{v}(t) \quad (11)$$

resulting in the system,

$$\dot{s}(t) = A_c s(t) + G\tilde{v}(t) \quad (12)$$

where,

$$A_c = diag(A_p, A_p), \quad (13)$$

being A_p the matrix,

$$A_p = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ k_1 & 0 & -k_1 & -k_2 \end{pmatrix} \quad (14)$$

B. Discretization of the continuous time model

We can split the discretization of the system from (12) in two equal parts and associate the separate results (for "x" and "y" directions) with block chunks in the discrete transition matrix and the covariance matrix of the discrete process noise. The discretization will be derived for one of the two directions and duplicated for the other. If we select the direction "x" for discretization,

$$\dot{s}_x(t) = A_p s_x(t) + G\tilde{v}_x \quad (15)$$

where,

$$\tilde{v}_x = \begin{bmatrix} \tilde{v}_{x^b} \\ \tilde{v}_{x^r} \end{bmatrix} \quad (16)$$

its discrete version can be represented as,

$$s_{x,k+1} = A_b s_{x,k} + v_{x,k} \quad (17)$$

Moreover, the discretization of the entire system in (12) can be expressed as,

$$s_{k+1} = A_d s_k + v_k \quad (18)$$

where A_b can be replicated in the discrete system transition matrix A_d as,

$$A_d = \begin{pmatrix} A_b & 0 \\ 0 & A_b \end{pmatrix} \quad (19)$$

A_b is (see Appendix),

$$A_b = \begin{pmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\lambda_2(1-e^{\delta t\lambda_3})-\lambda_3(1-e^{\delta t\lambda_2})}{\lambda_2-\lambda_3} & A_b(3,2) & \frac{\lambda_2 e^{\delta t\lambda_3}-\lambda_3 e^{\delta t\lambda_2}}{\lambda_2-\lambda_3} & \frac{e^{\delta t\lambda_2}-e^{\delta t\lambda_3}}{\lambda_2-\lambda_3} \\ \frac{\lambda_2\lambda_3(e^{\delta t\lambda_2}-e^{\delta t\lambda_3})}{\lambda_2-\lambda_3} & A_b(4,2) & -\frac{\lambda_2\lambda_3(e^{\delta t\lambda_2}-e^{\delta t\lambda_3})}{\lambda_2-\lambda_3} & \frac{\lambda_2 e^{\delta t\lambda_2}-\lambda_3 e^{\delta t\lambda_3}}{\lambda_2-\lambda_3} \end{pmatrix} \quad (20)$$

where $A_b(3,2) = -\frac{k_1(e^{\delta t\lambda_2}-e^{\delta t\lambda_3}-\delta t\lambda_2+\delta t\lambda_3)+k_2\lambda_2(1-e^{\delta t\lambda_3})-k_2\lambda_3(1-e^{\delta t\lambda_2})}{k_1(\lambda_2-\lambda_3)}$ and $A_b(4,2) = \frac{k_1(\lambda_2-\lambda_3-\lambda_2 e^{\delta t\lambda_2}+\lambda_3 e^{\delta t\lambda_3})-k_2\lambda_2\lambda_3(e^{\delta t\lambda_2}-e^{\delta t\lambda_3})}{k_1(\lambda_2-\lambda_3)}$.

Furthermore, v in (18) is the discrete process noise with zero mean, and covariance matrix given by,

$$cov[v] = C_d = \begin{pmatrix} C_b & 0 \\ 0 & C_b \end{pmatrix} \quad (21)$$

The discrete time random component for direction "x" v_x is given by the integral:

$$v_x(k\delta t) = \int_{k\delta t}^{(k+1)\delta t} e^{A_p((k+1)\delta t-\tau)} G(\tau)\tilde{v}_x(\tau)d\tau \quad (22)$$

which represents a four dimensional multivariate Gaussian distribution parametrized by zero mean and covariance matrix:

$$C_b = \tilde{q}_x \int_0^{\delta t} e^{A_p(\delta t-\tau)} G[e^{A_p(\delta t-\tau)} G]^T d\tau. \quad (23)$$

where $\tilde{q}_x$ is the continuous-time process noise intensity in "x" direction. The explicit expression of C_b as a function of the parameters of the interactive motion model is quite extensive and therefore omitted due to space limitations.

C. Consideration of variable interacting targets motion

Hidden interactions between the objects and the environment may be variable and therefore the unknown parameters k_1 and k_2 should account for this variability as well. Objects may even gradually change from independent motion to interacting motion when e.g., targets that are far away from each other moving independently "decide" to gather and travel in group formation for a while.

Incorporation of the knowledge about objects interacting behavior requires an effort of on-line learning k_1 and k_2. This can be done directly by extending the state vector with k_1 and k_2 and their respective change rate. The extended state vector considered is $\underline{s}(t) = [s(t)\ k_1(t)\ \dot{k}_1(t)\ k_2(t)\ \dot{k}_2(t)]^T$. Indeed, k_1 and k_2 are not considered constants any more. Instead, k_1 and k_2 are considered variables and assumed to evolved according to independent Continuous White Noise Acceleration models.

Then, the model considered for the new state vector can be expressed as,

$$\underline{s}_{k+1} = A_e \underline{s}_k + v_{e,k} \quad (24)$$

where,

$$A_e = \begin{pmatrix} A_d & 0 \\ 0 & A_k \end{pmatrix} \quad (25)$$

the blocks of A_e associated to coordinates k_1 and k_2 are straightforward to calculate as they are uncoupled from any other coordinate,

$$A_k = \begin{pmatrix} A_u & 0 \\ 0 & A_u \end{pmatrix} \ ; where \ \ A_u = \begin{pmatrix} 1 & \delta t \\ 0 & 1 \end{pmatrix} \quad (26)$$

being,

$$v_e = [v\ v_{k1}\ v_{\dot{k}1}\ v_{k2}\ v_{\dot{k}2}]^T \quad (27)$$

which represents a twelve dimensional multivariate Gaussian distribution parameterized by zero mean and covariance matrix,

$$cov[v_e] = \begin{pmatrix} C_d & 0 \\ 0 & C_k \end{pmatrix} \tag{28}$$

$$C_k = \begin{pmatrix} C_u & 0 \\ 0 & C_u \end{pmatrix} \tag{29}$$

again, the blocks C_u associated to coordinates k_1 and k_2 are straightforward to calculate,

$$C_u = \begin{pmatrix} \frac{\delta t^3}{3} & \frac{\delta t^2}{2} \\ \frac{\delta t^2}{2} & \delta t \end{pmatrix} \tilde{q}_k \tag{30}$$

where $\tilde{q}_k$ is the continuous-time process noise intensity of the variabes modeling the interacting motion.

III. Integration of Interacting Motion in a Tracking Application

This Section contains all necessary aspects that are needed to develop a particle filtering algorithm based on our model of targets interaction.

Let $s_k \in \mathbb{R}^d$ denote the state vector and $z_k \in \mathbb{R}^m$ denote the measurement vector at time index k. Z_k denotes the set of measurements up to time k, including z_k: $Z_k = \{z_1, z_2, ..., z_k\}$. The state space model can be represented by two conditional probability densities:

$$s_{k+1} \sim p(s_{k+1}|s_k), \tag{31}$$

$$z_k \sim p(z_k|s_k), \tag{32}$$

where $p(s_{k+1}|s_k)$ can be derived from the dynamic model. In our particular case, $p(s_{k+1}|s_k)$ is represented by Eq. (24). $p(z_k|s_k)$ is the likelihood of the measurements given the state. This likelihood function is presented in Subsection III-A.

A. Model of measurements: radar track-before-detect

For the numerical experiments we consider a radar TBD measurement model as in [9]. The measurement model used defines the power reflected by the two objects for each cell. One measurement z_k is composed of $N_r \times N_b$ power measurements z_k^{ij}, where $k \in \mathbb{N}$ and N_r and N_b are the number of range and bearing cells.

$$z_k^{ij} = |z_{A,k}^{ij}|^2 = |A_k^{(1)} h_A^{(1)ij} + A_k^{(2)} h_A^{(2)ij} + n_I(t_k) + i n_Q(t_k)|^2 \tag{33}$$

where A_k is the complex amplitude of the target.

$$A_k = \tilde{A}_k^{(t)} e^{i\phi_k}, \;\; \phi_k \in U(0, 2\pi) \tag{34}$$

$$h_A^{(t)ij}(s_k^t) = e^{-\frac{(r_i - r_k^t)^2}{2R} - \frac{(b_j - b_k^t)^2}{2B}}, \;\; i = 1, ..., N_r, \;\; j = 1, ..., N_b \tag{35}$$

r_k^t and b_k^t are the range and bearing of the target t and R and B are constants related to the size of the cells

$$r_k^t = \sqrt{(x_k^t)^2 + (y_k^t)^2} \text{ and } \;\; b_k^t = arctan(\frac{y_k^t}{x_k^t}) \tag{36}$$

The noise in Eq. (33) is complex Gaussian, where $n_I(t_k)$ and $n_Q(t_k)$ are independent, zero-mean white Gaussian with variance σ_n^2 accounting for the in phase and quadrature phase respectively. These measurements, conditioned on the state are assumed to be exponentially distributed [10],

$$p(z^{ij}|s_k) = \frac{1}{\mu_0^{ij}} e^{-\frac{1}{\mu_0^{ij}} z_k^{ij}} \tag{37}$$

where

$$\mu_0^{ij} = (\tilde{A}_k^{(1)})^2 (h_A^{(1)ij}(s_k, t_k))^2 + (\tilde{A}_k^{(2)})^2 (h_A^{(2)ij}(s_k, t_k))^2 + 2\sigma_n^2 \tag{38}$$

Assuming that the noise is independent from cell to cell and that the reflections of the two targets are independent, the likelihood function becomes:

$$p(z_k|s_k) = \prod_{ij} p(z^{ij}|s_k) \tag{39}$$

B. Algorithm

In the framework of recursive Bayesian estimation the prior density $p(s_{k+1}|Z_k)$ and the posterior density $p(s_{k+1}|Z_{k+1)})$ can be obtained as,

$$p(s_{k+1}|Z_k) = \int p(s_{k+1}|s_k) p(s_k|Z_k) ds_k \tag{40}$$

$$p(s_{k+1}|Z_{k+1}) = \frac{p(z_{k+1}|s_{k+1}) p(s_{k+1}|Z_k)}{p(z_{k+1}|Z_k)} \tag{41}$$

A particle filter is selected for the implementation due to the non-linearity in the dynamic and measurement models (see Eqs. (24) and (33)). An Importance-Sampling-based filter algorithm suffices to show the benefits of modeling target interactions. In particular, Algorithm 1 presents the SIR filter that will be used in the simulations in Subsection IV-A. The particle cloud representation of the joint probability density is given by the weighted particles $\{s_k^i, w_k^i\}_{i=1}^{N_p}$.

1 $k = 0$
2 Draw N_p samples s_k^i from $p(s_k)$
3 $k = k + 1$
4 Draw N_p samples v_k^i from $p(v_k)$
5 Obtain N_p samples s_k^i from $p(s_k^i|s_{k-1}^i, v_k^i)$
6 Given z_k, obtain $\tilde{w}_k^i = p(z_k|s_k^i)$
7 Normalize weights $w_k^i = \tilde{w}_k^i / \sum_{j=1}^{N_p} \tilde{w}_k^j$
8 Resample from $\hat{p}(s_k|Z_k) = \sum_{i=1}^{N_p} w_k^i \delta(s_k - s_k^i)$
9 Extract point estimates from $\hat{p}(s_k|Z_k)$, e.g. according to (42)
10 go to 3

Algorithm 1: Pseudo-code of the SIR filter.

$$\mathrm{PE}_k = [\bar{x}_k^b, \bar{x}_k^r]^T$$
$$\bar{x}_k^c = \frac{\sum_{i=1}^{N_p} x_k^{i,c}}{N_p} \;\; , \;\; c = \{b, r\} \tag{42}$$

IV. Simulation examples

In this Section we present simulation results for different interacting targets scenarios. In particular, we compare the performance of a traditional filter based on independent motion assumptions (using the model presented in the introduction of Section II) with the proposed method (based on the interacting target motion modeling in Subsection II-C).

A. Preliminary results

The first set of interacting trajectories that we simulate may well represent the motion of a water-skier being pulled by a tow rope attached to a boat. The trajectory of the water-skier and the boat being the continuous red and blue lines in Fig. 2 respectively. The trajectories are generated using the model in (24) and the evolution of k_1 and k_2 is represented by the continuous lines in Fig. 6. Preliminary tracking results are shown in Figs. 2, 3, 4 and 5. The first two show the estimation performance of a traditional filter based on coordinates decoupling assumptions. Figs. 4 and 5 show the estimation performance of our approach.

Although the trajectories are strongly coupled, the traditional filter manages to keep the targets in track as the process noise variance of the model is tunned in relation with the maneuverability of the targets. However, it incurs in large estimation errors, especially when strong accelerations take place (see Fig. 3). Fig. 5 shows the estimation error reduction in the position estimates when the information about target interaction is incorporated. Not surprisingly, once the parameters of the interactive motion are estimated (see Fig. 6), they can be readily exploited lowering the uncertainty in the position space.

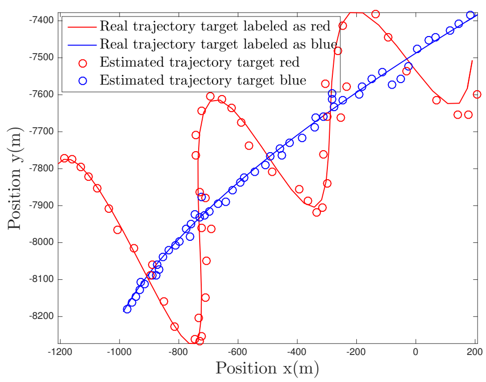

Fig. 2. Position estimates of a traditional filter with independent motion assumptions.

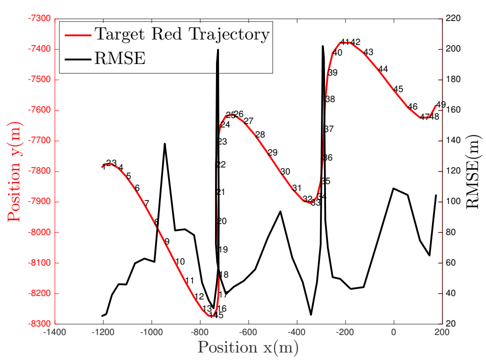

Fig. 3. Error on the position estimates for the target labeled as red by a traditional filter with independent motion assumptions. Estimation error is presented in the same x position axis as the trajectory to show that the maximums in error match the largest accelerations due to targets interaction. Timing information (in seconds) is also provided along the trajectory.

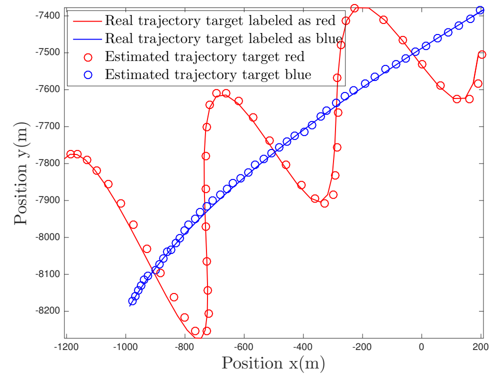

Fig. 4. Position estimates of our filter encapsulating interacting motion assumptions.

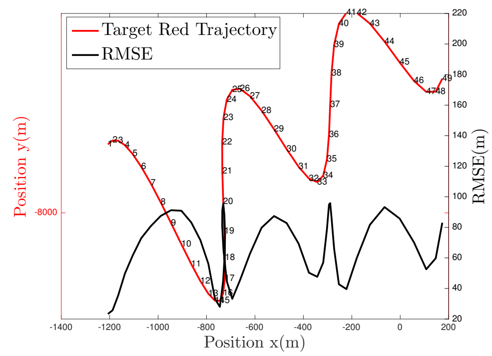

Fig. 5. Error on the position estimates for the target labeled as red by our filter with interacting motion assumptions.

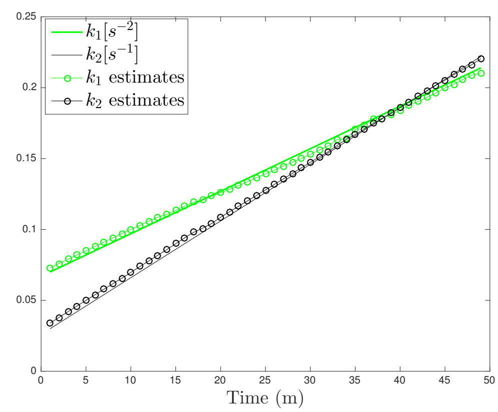

Fig. 6. Estimation of the parameters in the interacting motion model.

B. Final results

In some applications it is desirable to detect certain types of coupled motion. The purpose of the paper is not only improving tracking performance exploiting some knowledge about the interaction of the targets. We also aim at detecting changes between interactive/no-interactive behaviors. A MM filtering implementation serves a useful functionality to integrate both aims. In particular, we compare a traditional filter with a MM filter where the two models used in previous

Subsection are encapsulated. By these means, the posterior mode probabilities can be used as a measure of certainty about whether or not the objects have interacting behaviors. Moreover, the MM mechanism gives more importance to the mode with higher posterior probability. Therefore, reduction in the estimation error is expected disregarding whether of not the objects are interacting.

Let 1 and 2 denote the no-interacting and the interacting models respectively. Moreover, w_k^{i*} and m_k^{i*} are used to denote the weight after resampling and the mode after resampling of particle i at time instant k respectively. Then, the posterior probability of each mode can be calculated using the particle-based approximation as follows,

$$p(m_k = 1|Z_k) \approx \sum_{i:m_k^{i*}=1} w_k^{i*} \approx \frac{\#\{m_k^{i*} = 1\}}{N_p} \tag{43}$$

$$p(m_k = 2|Z_k) \approx \frac{\#\{m_k^{i*} = 2\}}{N_p} \tag{44}$$

Algorithm 2 summarizes the MM SIR filter used in the simulation example in this Subsection.

1 $k = 0$
2 Draw N_p samples s_k^i from $p(s_k)$
3 Draw N_p samples m_k^i from $p(m_k)$
4 $k = k + 1$
5 Draw N_p samples v_k^i from $p(v_k)$
6 Obtain N_p samples s_k^i from $p(s_k^i|s_{k-1}^i, v_k^i, m_{k-1}^i)$
7 Given z_k, obtain $\tilde{w}_k^i = p(z_k|s_k^i)$
8 Normalize weights $w_k^i = \tilde{w}_k^i / \sum_{j=1}^{N_p} \tilde{w}_k^j$
9 Resample from $\hat{p}(s_k|Z_k) = \sum_{i=1}^{N_p} w_k^i \delta(s_k - s_k^i)$:
10 Calculate $p(m_{k-1}|Z_k)$ according to Eqs. (43) and (44)
11 Extract point estimates from $\hat{p}(s_k|Z_k)$, e.g. according to (42)
12 Draw N_p samples m_k^i from $p(m_k|m_{k-1}^i)$
13 go to 4

Algorithm 2: Pseudo-code of the MM SIR filter.

The final results show the performance of a MM particle filter exposed to periodical switches between interacting and independent target motion. The MM filter is compared with a traditional particle filter were no efforts are made to model interactions. For a fair comparison, we run the traditional particle filter with the same amount of particles used in the MM filter. However, the variance of the process noise of the traditional filter should be tuned to a larger magnitude (en relation to targets maneuverability) than in the MM implementation due to the lack of interacting information. Otherwise, the traditional filter will not be able to maintain the targets in track.

We generate trajectories where the objects maneuver coordinately every now and then. The real trajectories are represented by the continuous lines in Fig. 7. The periods of interactive/independent motion can be interpreted considering the information in Fig. 9. Continuous lines represent the ground truth of k_1 and k_2 parameters. These are zero between 43 and 61 seconds, meaning that no inputs are applied and therefore the targets are moving independently in this time-slot. Over the first 42 seconds and the last 39 seconds the targets are interacting with each other.

Figs. 7 and 11 show the point estimates extracted from the MM and traditional filters respectively. The MM particle filter provides more accurate point estimates than the traditional filtering solution as shown in Figs. 8 and 12. In Fig. 11 it is also clear that the large uncertainty when using a traditional filter compromises labeling with respect to target identity if the targets get close enough (both in position and velocity space). This negative effect may get unnoticed at the application level if no special considerations are taken into account. In line with recent literature on the labeling problem, the root cause of labeling uncertainty turns out to be the uncertainty in the joint multi-target space. For an in depth explanation on characterization of labeling uncertainty refer to [11].

Fig. 9 shows the estimation of the variables in the coupled motion model. Periods of interaction/no-interaction are correctly detected relying on the estimated posterior mode probabilities of the MM filter (see Fig. 10).

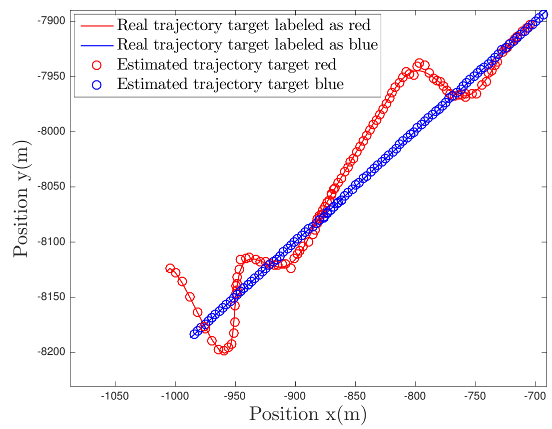

Fig. 7. Estimated position of the targets by the MM filter

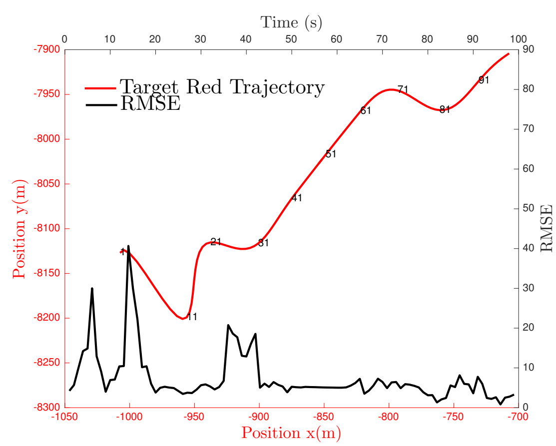

Fig. 8. Error on the position estimates by the MM filter for target labeled as red. Each curve is referenced to the axis scale on its same color. Time scale is also shown over the trajectory.

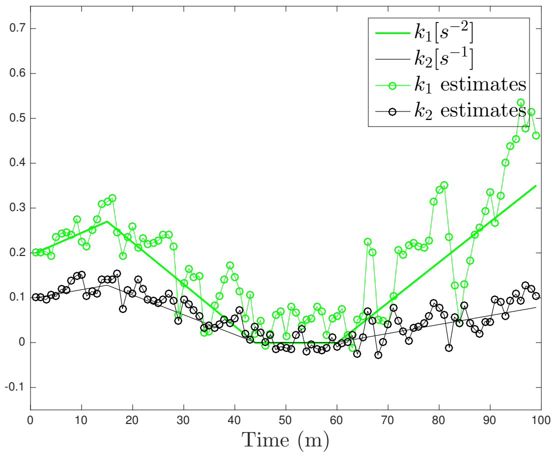

Fig. 9. Estimation of the parameters in the interacting motion model.

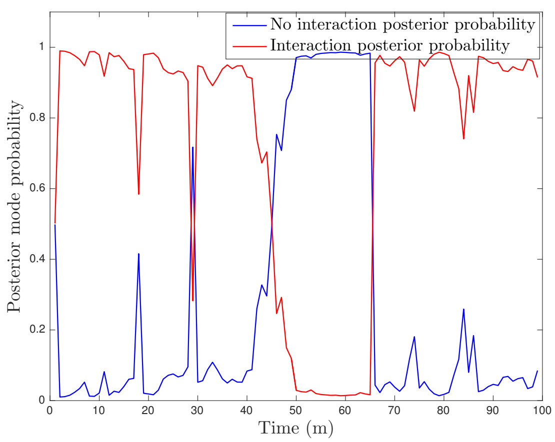

Fig. 10. Estimation of targets behavior

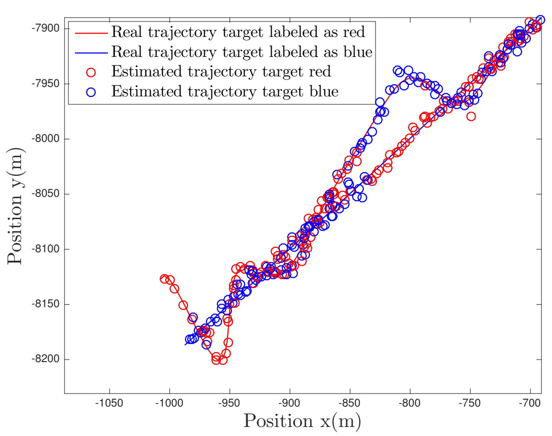

Fig. 11. Position estimates of a traditional filter with independent motion assumptions.

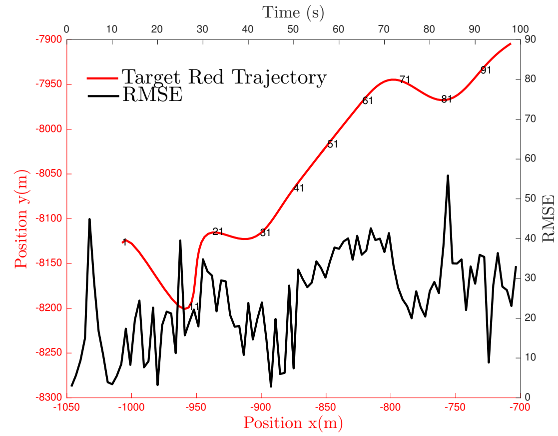

Fig. 12. Error on the position estimates for target labeled as red by a traditional filter with independent motion assumptions.

The "curse" of dimensionality inherited from the particle filter implementation is visible, for instance, in the estimation of k_1 and k_2. However, as the number of targets is small, increasing the number of particles to $3 \cdot 10^5$ produces acceptable results. Since the main contribution is on the modeling of interacting targets, efficiency problems of the implementation (that will appear when considering larger number of targets) fall out of the scope of this paper.

V. Conclusions

In this work we have presented a method to incorporate interacting targets behavior in a MTT application. The method relies on parametric coupling assumptions along coordinates of different targets. In particular, the parameters modeling interactions are estimated as part of the state vector and exploited on-line. Not surprisingly, when the model assumptions fit well to the targets motion, estimation errors can be reduced (in comparison with a traditional model where no efforts are made to model interactions). Moreover, inference about the existence of interactive behavior can be done by encapsulating our method in an MM filtering implementation. Finally, we have shown that in line with recent literature on the labeling problem, reduction of uncertainty in the joint multi-target position space results on reduction of labeling uncertainty. This can also be achieved by incorporating interacting targets motion knowledge.

As for future work, it will be interesting to see if the points of attraction/repulsion can be estimated in the filtering process. This would allow to drop the strongest assumption in the model. By these means, the method would become competitive in more realistic scenarios since, often times, not so much information about targets interaction is available.

Appendix

The solution of the differential stochastic equation in Fig. (1) for "x" direction is,

$$s_x(t) = e^{A_p(t)t} \int G(t)\tilde{v}_x(t)e^{-A_p(t)t}dt \tag{45}$$

The discretization for a revisit time δt results in,

$$s_x((k+1)\delta t) = e^{A_p(t)\delta t} s_x(k\delta t) + \int_{k\delta t}^{(k+1)\delta t} e^{A_p((k+1)\delta t - \tau)} G(\tau)\tilde{v}_x(\tau) d\tau \tag{46}$$

Therefore, the block A_b in (17) is given by the exponential matrix $e^{A_p(t)\delta t}$.

From linear algebra theory we know that for any continuous scalar-valued function $f(x) : \mathbb{C} \to \mathbb{C}$ and for any square matrix $X \in \mathbb{R}^{nxn}$ (or $\in \mathbb{C}^{nxn}$):

$$f(X\delta t) = M \begin{pmatrix} f(J_{k1}(\lambda_1)\delta t) & & 0 \\ & \ddots & \\ 0 & & f(J_{km}(\lambda_m)\delta t) \end{pmatrix} M^{-1}, \tag{47}$$

where δt is a scalar, M is nonsingular and $J_{k1}(\lambda_1)$,...,$J_{km}(\lambda_1)$ are the so-called Jordan Blocks associated to eigenvectors $\lambda_1, ..., \lambda_m$. Each of them defined as an upper triangular matrix such that $J_k(\lambda) \in \mathbb{C}^{kxk}$ is,

$$J_k(\lambda) = \begin{pmatrix} \lambda & 1 & & 0 \\ & \ddots & \ddots & \\ & & \lambda & 1 \\ 0 & & & \lambda \end{pmatrix} \tag{48}$$

$f(J_k(\lambda)\delta t)$ is defined as,

$$f(J_k(\lambda)\delta t) = \begin{pmatrix} f(\lambda\delta t) & \delta t f'(\lambda\delta t) & \frac{\delta t^2}{2!}\delta t f''(\lambda\delta t) & \dots & \frac{\delta t^{k-1}}{(k-1)!}\delta t f^{(k-1)}(\lambda\delta t) \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \frac{\delta t^2}{2!}\delta t f''(\lambda\delta t) \\ & & & \ddots & \delta t f'(\lambda\delta t) \\ 0 & & & & f(\lambda\delta t) \end{pmatrix} \tag{49}$$

Given these definitions, the matrix J is the Jordan canonical form of the matrix X through the similarity transformation M, which existence is ensured in linear algebra theory.

$$J = \begin{pmatrix} J_{k1}(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & J_{km}(\lambda_m) \end{pmatrix} = M^{-1} X M \tag{50}$$

X is, in our particular case, the continuous time system function for any of the "x", "y" directions,

$$A_p = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ k_1 & 0 & -k_1 & -k_2 \end{pmatrix}$$

The similarity transformation,

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & -\frac{k_2}{k_1} & 1 & 1 \\ 0 & 1 & \lambda_2 & \lambda_3 \end{pmatrix}, with\ \lambda_2 = \frac{-k_2 - \sqrt{k_2^2 - 4k_1}}{2}\ ,\ \lambda_3 = \frac{-k_2 + \sqrt{k_2^2 - 4k_1}}{2} \tag{51}$$

satisfies $J = M^{-1} A_p M$ as long as k_1 and k_2 are no zero and $k_2^2 \neq 4k_1$. We can neglect these particular cases as k_1 and k_2 are considered unknowns (part of the state vector) and a particle filter is considered for the implementation. Because of that, components k_1 and k_2 of the particles are affected by the process noise in Eq. (24), therefore the mentioned particular cases have probability zero.

Then, A_b can be put in the form of Eq. (47) as,

$$A_b = e^{A_p\delta t} = M \begin{pmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{\lambda_2\delta t} & 0 \\ 0 & 0 & 0 & e^{\lambda_3\delta t} \end{pmatrix} M^{-1}$$

which results in,

$$\begin{pmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\lambda_2(1-e^{\delta t\lambda_3})-\lambda_3(1-e^{\delta t\lambda_2})}{\lambda_2-\lambda_3} & A_b(3,2) & \frac{\lambda_2 e^{\delta t\lambda_3}-\lambda_3 e^{\delta t\lambda_2}}{\lambda_2-\lambda_3} & \frac{e^{\delta t\lambda_2}-e^{\delta t\lambda_3}}{\lambda_2-\lambda_3} \\ \frac{\lambda_2\lambda_3(e^{\delta t\lambda_2}-e^{\delta t\lambda_3})}{\lambda_2-\lambda_3} & A_b(4,2) & -\frac{\lambda_2\lambda_3(e^{\delta t\lambda_2}-e^{\delta t\lambda_3})}{\lambda_2-\lambda_3} & \frac{\lambda_2 e^{\delta t\lambda_2}-\lambda_3 e^{\delta t\lambda_3}}{\lambda_2-\lambda_3} \end{pmatrix} \tag{52}$$

where,

$$A_b(3,2) = -\frac{k_1(e^{\delta t\lambda_2}-e^{\delta t\lambda_3}-\delta t\lambda_2+\delta t\lambda_3)+k_2\lambda_2(1-e^{\delta t\lambda_3})-k_2\lambda_3(1-e^{\delta t\lambda_2})}{k_1(\lambda_2-\lambda_3)}$$

$$A_b(4,2) = \frac{k_1(\lambda_2-\lambda_3-\lambda_2 e^{\delta t\lambda_2}+\lambda_3 e^{\delta t\lambda_3})-k_2\lambda_2\lambda_3(e^{\delta t\lambda_2}-e^{\delta t\lambda_3})}{k_1(\lambda_2-\lambda_3)}$$

Acknowledgement

This research has been carried out within the **TRAX** project, funded by the EUs Seventh Framework Programme under grant agreement Nº607400.

References

[1] T. Fortmann, Y. Bar-Shalom and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," in *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173-184, Jul 1983.

[2] D. Reid, "An algorithm for tracking multiple targets," in *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843-854, Dec 1979

[3] A. Gning, L. Mihaylova, S. Maskell, S. K. Pang and S. Godsill, "Group Object Structure and State Estimation With Evolving Networks and Monte Carlo Methods," in *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1383-1396, April 2011.

[4] D. Helbing, and P. Molnár, "Social force model for pedestrian dynamics," Phys. Rev. E, American Physical Society, vol. 51, pp. 42824286, 1995

[5] M. Luber, J. A. Stork, G. D. Tipaldi, K. O. Arras, "People tracking with human motion predictions from social forces," in *proceedings of the Robotic and Automation International Conference*, Anchorage, AK, 2010, pp. 464-469.

[6] A. Ur-Rehman, S. M. Naqvi, L. Mihaylova and J. A. Chambers, "Multi-Target Tracking and Occlusion Handling With Learned Variational Bayesian Clusters and a Social Force Model," in *IEEE Transactions on Signal Processing*, vol. 64, no. 5, pp. 1320-1335, March1, 2016.

[7] R. Mazzon, A. Cavallaro, "Multi-camera tracking using a Multi-Goal Social Force Model," *Neurocomputing*, Volume 100, 16 January 2013, Pages 41-50, ISSN 0925-2312.

[8] Y. Bar-Shalom, X. Rong Li and T. Kirubarajan "Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software," John Wiley & Sons, Inc, Canada, 2001

[9] Y. Boers and J. N. Driessen, "Multitarget particle filter track before detect application," in *IEE Proceedings - Radar, Sonar and Navigation*, vol. 151, no. 6, pp. 351-357, 10 Dec. 2004

[10] M. I.Skolnik, "Introduction to Radar Systems" McGraw-Hill, Inc, Singapore, 1981, 2nd. edn.

[11] C. M. Leon, H. Driessen and P. K. Mandal, "Efficient characterization of labeling uncertainty in closely-spaced targets tracking," in *proceedings of the 19th International Conference on Information Fusion*, Heidelberg, 2016, pp. 449-456.