Proceedings Paper:

Kim, J orcid.org/0000-0002-3456-6614 (2016) Nonlinear projection filter for target tracking using range sensor & optical tracker. In: IFAC-PapersOnLine. 20th IFAC Symposium on Automatic Control in Aerospace - ACA 2016, 21-25 Aug 2016, Sherbrooke, Quebec, Canada. Elsevier , pp. 438-443.

https://doi.org/10.1016/j.ifacol.2016.09.075

Nonlinear projection filter for target tracking using range sensor & optical tracker ⋆

Jongrae Kim *

* *Institute of Design, Robotics & Optimisation (iDRO), School of Mechanical Engineering, University of Leeds, Leeds LS2 9JT, UK (e-mail: menjkim@leeds.ac.uk)*

Abstract: Target tracking filters have a variety of applications in various areas. Typically, a radar provides the range measurement and an optical sensor measures the orientation of a target. The measurements provided by the sensors have very strong nonlinearities with the states of the target given in the Cartesian coordinates while its dynamics is linear parameter time-varying. The time-varying component exists because of the unknown acceleration input in the target. Nonlinear projection filter provides a solution to the nonlinear estimation problem by approximating the solution as a linear combination of orthogonal basis functions. The analytic expression for propagating the joint probability density function is derived for the target tacking problem and this reduces large amount of computation times, where the filter equations are normally obtained numerically. The effectiveness of the filter is demonstrated by a numerical simulation.

Keywords: Nonlinear estimation, Nonlinear projection filter, Target tracking

1. INTRODUCTION

Moving target tracking has been studied extensively in the past and it has many areas of applications for tracking various moving objects such as aircraft (Spingarn and Weidemann, 1972), ships (Chen and Huang, 2013), mobile robots (Benavidez and Jamshidi, 2011). One of the earliest studies includes α-β filter (Rogers, 1987), α-β-γ filter (Gray and Murray, 1993) derived from the steady-state Kalman filter (Kalman, 1960). A stability analysis of α-β-γ filter is shown in Tenne and Singh (2002). Target tracking using particle filter is shown in Gustafsson et al. (2002). A good survey of the target tracking problem for various types of acceleration models is found in Li and Jilkov (2003).

One of the main difficulties in target tracking filter design stems from its strong nonlinearities in the measurement equation. The range measurements from a radar and the azimuth/elevation angles from an optical tracker are typical sensor outputs in the target tracking problems. These measurements have strong nonlinear relations with a Cartesian coordinates dynamics of the moving target (Park, 1999).

Nonlinear projection filter solves the nonlinear estimation problem directly from the Fokker-Planck equation (Beard et al., 1999). The Fokker-Planck equation is in general difficult to solve in real-time because of its high computational demand. Nonlinear projection filter provides the way to reduce the computational complexity through the approximation of the solution as a linear combination of a set of orthogonal basis functions. Recently, the filter is further improved so that it requires less computation and fits better to an array of parallel sensors and parallel computation in (Single-Liertz et al., 2015).

In the following, firstly, a target tracking problem is introduced in terms of its dynamics and measurement equation; secondly, the nonlinear projection filter is summarised; thirdly, the filter is developed for the target tracking; fourthly, simulation results are discussed; and finally, conclusions are presented.

2. TARGET TRACKING

2.1 Dynamics & Measurements

A target dynamics for each axis can be written as follows:

$$\frac{d}{dt}\begin{bmatrix} r \\ \dot{r} \\ w_r \end{bmatrix} = \begin{bmatrix} \dot{r} \\ -\tau\dot{r} + u + w_r \\ -\alpha w_r \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v \tag{1}$$

where r is equal to x, y or z, which is the component of each direction in the Cartesian coordinates, d/dt is the time derivative, r is the position of the target in the given axis, $\dot{r}$ is the time derivative of r, τ is the target drag coefficient in r-direction, u is the target control input in r-direction, which is in the known bound in $[\underline{u}, \bar{u}]$, w_r is the uncertainty in the target acceleration (Singer, 1970), α is the inverse of the manoeuvre time constant, and v is zero-mean white noise Gaussian with the variance, σ_v^2. Or, in a compact form

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, u)dt + Gd\beta = (A\mathbf{x} + \mathbf{u})\, dt + Gd\beta \tag{2}$$

where $\mathbf{f}(\mathbf{x}, u) = A\mathbf{x} + \mathbf{u}$,

⋆ This research is supported by EPSRC Research Grant, EP/N010523/1, Balancing the impact of city infrastructure engineering on natural systems using robots.

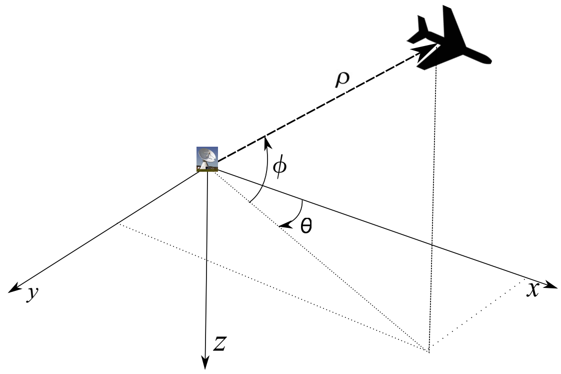

Fig. 1. Tracking coordinates, x, y, and z, and the measurements, range, ρ, elevation, ϕ, and azimuth, θ

$$\mathbf{x} = \begin{bmatrix} r \\ \dot{r} \\ w \end{bmatrix}, A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\tau & 1 \\ 0 & 0 & -\alpha \end{bmatrix}, \mathbf{u} = \begin{bmatrix} 0 \\ u \\ 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad (3)$$

and $\mathrm{E}(\beta^2) = \sigma_v^2 dt$.

The measurement given by a range sensor and an optical tracker are

$$\mathbf{y}_k = \begin{bmatrix} \rho_k \\ \phi_k \\ \theta_k \end{bmatrix} = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \qquad (4)$$

where each component of the measurement represents the range measurement from radar, ρ_k, the elevation, ψ_k, and the azimuth, θ_k, measurements from optical trackers, respectively, as shown in Figure 1. And, they satisfy the following equations with the Cartesian coordinates:

$$\mathbf{h}(\mathbf{x}_k) = \begin{bmatrix} \sqrt{x_k^2 + y_k^2 + z_k^2} \\ \tan^{-1}\left(-\dfrac{z_k}{\sqrt{x_k^2 + y_k^2}}\right) \\ \tan^{-1}\left(\dfrac{y_k}{x_k}\right) \end{bmatrix}, \ \mathbf{v}_k = \begin{bmatrix} v_\rho \\ v_\phi \\ v_\theta \end{bmatrix} \qquad (5)$$

and x_k, y_k, z_k are the Cartesian coordinates at the k-th sampling time. Note that (1) can be discretised and the filtering problem is considered purely in the discrete domain together with the discrete measurement, (4). Without introducing the discrete approximation, the original continuous-discrete mixed filter problem is to be solved.

2.2 Nonlinear projection filter

For a standard form of nonlinear stochastic differential equation given in (2) with a fixed u, the corresponding joint pdf (Probability Density Function), $p(t, \mathbf{x})$, follows the Fokker-Planck equation given by

$$\frac{\partial p}{\partial t} = -\sum_{i=1}^{n} \frac{\partial (pf_i)}{\partial x_i} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 \left[p\left(GQG^T\right)_{i,j}\right]}{\partial x_i \partial x_j} \qquad (6)$$

where x_i and f_i are i-th element of the n-dimensional vectors $\mathbf{x}$ and $\mathbf{f}(\mathbf{x}, u)$, respectively, $(GQG^T)_{i,j}$ is i-th row and j-th column element of the matrix, GQG^T, and Q is equal to σ_v^2.

The nonlinear projection filter assumes that the joint pdf has the following form:

$$p(t, \mathbf{x}) = \mathbf{c}^T(t)\boldsymbol{\phi}(\mathbf{x}) \qquad (7)$$

where $\mathbf{c}$ is a time-varying vector, whose dimension is N_p, and $\boldsymbol{\phi}$ is N_p-dimensional vector, whose elements form a set of orthogonal basis functions. Substituting the approximation into the Fokker-Planck equation gives the propagation equation for $\mathbf{c}(t)$.

Substituting the approximation into the following Bayes' theorem, (Arulampalam et al., 2002), provides the update equation for $\mathbf{c}(t)$:

$$p(t_k^+, \mathbf{x}|Y_k) = \eta \ p(\mathbf{y}_k|\mathbf{x}) \ p(t_k^-, \mathbf{x}|Y_{k-1}) \qquad (8)$$

where η is the normalising constant, $p(\mathbf{y}_k|\mathbf{x})$ is the sensor model.

The full details on the derivation of nonlinear projection filter for a general form of nonlinear equation can be found in Beard et al. (1999).

2.3 Filter Model & Basis Functions

Not all parameters in the target dynamics shown in (1) are known perfectly and they might change with time, for example, depending on the acceleration input, u, or the manoeuvre profile given by α. It is, hence, that the following simpler equation would be more attractive in deriving the projection filter:

$$d\begin{bmatrix} r \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dot{r} \\ u_r \end{bmatrix} dt + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d\gamma \qquad (9)$$

where u_r is a priori estimated acceleration bound for a given target and $\mathrm{E}(\gamma^2) = \sigma_\gamma^2 dt$, and σ_γ^2 is the variance.

In the projection filter, the pdf is approximated by a finite sum of basis functions as follows:

$$p(t, r, \dot{r}) \approx \mathbf{c}_\mathbf{r}^T(t)\boldsymbol{\phi}(r, \dot{r}) \qquad (10)$$

where $\mathbf{c}_\mathbf{r}(t)$ is an N^2-dimensional time-varying vector, $(\cdot)^T$ is the transpose,

$$\boldsymbol{\phi}(r, \dot{r}) = \boldsymbol{\xi}(r) \otimes \boldsymbol{\xi}(\dot{r}) \qquad (11)$$

$\otimes$ is the Kronecker product, $\boldsymbol{\xi}(s)$ is a N-dimensional vector for $s = r$ or $\dot{r}$. The i-th component of $\boldsymbol{\xi}(s)$ is the element of a set of orthogonal basis functions as follows:

$$\xi_i(s) = \begin{cases} 1/\sqrt{\Delta_s} & \text{for } i = 1, \\ \sqrt{\dfrac{2}{\Delta_s}} \cos\left[\kappa(i)(s - a_s)\right] & \text{for } 2 \le i \le N_s \end{cases} \qquad (12)$$

where Δ_s is equal to $b_s - a_s$, a_s is the lower bound of s, b_s is the upper bound of s, i.e., $s \in [a_s, b_s]$, N_s is the number of basis functions, and

$$\kappa_s(i) = \frac{(i-1)\pi}{\Delta_s} \qquad (13)$$

In addition, $\xi_i(s)$ satisfies the following orthogonality condition:

$$\int_{a_s}^{b_s} \xi_i(s) \ \xi_j(s) \ ds = \begin{cases} 1 & \text{for } i = j, \\ 0 & \text{for } i \ne j \end{cases} \qquad (14)$$

for $i, j = 1, 2, 3, \ldots, N_s$.

The following formula for the derivatives of the basis functions are useful later in deriving the projection filter equation. Take the derivative of $\boldsymbol{\xi}(s)$ by s

$$\boldsymbol{\xi}^{'}(s) = \frac{d\boldsymbol{\xi}(s)}{ds} = -\sqrt{\frac{2}{\Delta_s}} \begin{bmatrix} 0 \\ \kappa_s(2)\sin[\kappa_s(2)(s-a_s)] \\ \kappa_s(3)\sin[\kappa_s(3)(s-a_s)] \\ \vdots \\ \kappa_s(N)\sin[\kappa_s(N)(s-a_s)] \end{bmatrix} \tag{15}$$

and in a compact form,

$$\boldsymbol{\xi}^{'}(s) = -K_s\boldsymbol{\psi}(s) \tag{16}$$

where

$$K_s = \mathrm{diag}\left[\kappa_s(1)\ \kappa_s(2)\ \ldots\ \kappa_s(N)\right], \tag{17a}$$

$$\boldsymbol{\psi}(s) = \sqrt{\frac{2}{\Delta_s}} \begin{bmatrix} 0 \\ \sin[\kappa_s(2)(s-a_s)] \\ \sin[\kappa_s(3)(s-a_s)] \\ \vdots \\ \sin[\kappa_s(N)(s-a_s)] \end{bmatrix}, \tag{17b}$$

and diag[...] is the diagonal matrix, whose diagonal terms are given in the bracket. In addition, the second derivative can be easily shown to equal to the followings:

$$\begin{aligned} \boldsymbol{\xi}^{''}(s) &= \frac{d}{ds}\boldsymbol{\xi}^{'}(s) = -\frac{d}{ds}K_s\boldsymbol{\psi}(s) \\ &= -\sqrt{\frac{2}{\Delta_w}}K^2 \begin{bmatrix} 0 \\ \cos[\kappa_s(2)(s-a_s)] \\ \cos[\kappa_s(3)(s-a_s)] \\ \vdots \\ \cos[\kappa_s(N)(s-a_s)] \end{bmatrix} \\ &= -K_s^2 \begin{bmatrix} \frac{1}{\sqrt{\Delta_s}} \\ \sqrt{\frac{2}{\Delta_s}}\cos[\kappa_s(2)(s-a_s)] \\ \sqrt{\frac{2}{\Delta_s}}\cos[\kappa_s(3)(s-a_s)] \\ \vdots \\ \sqrt{\frac{2}{\Delta_s}}\cos[\kappa_s(N)(s-a_s)] \end{bmatrix} \end{aligned} \tag{18}$$

where the first element is zero as $\kappa(1) = 0$. Hence,

$$\boldsymbol{\xi}^{''}(s) = -K_s^2\boldsymbol{\xi}(s) \tag{19}$$

2.4 Propagation: target position & velocity

Apply the Fokker-Planck equation to (9)

$$\begin{aligned} \frac{\partial p(t,r,\dot{r})}{\partial t} &= -\frac{\partial\left[p(t,r,\dot{r})\ \dot{r}\right]}{\partial r} - \frac{\partial\left[p(t,r,\dot{r})\ u_r\right]}{\partial \dot{r}} \\ &+ \frac{1}{2}\frac{\partial^2\left[p(t,r,\dot{r})\ \sigma_\alpha\right]}{\partial \dot{r}^2} \end{aligned} \tag{20}$$

Substitute (10) with (11) into (20)

$$\begin{aligned} \dot{\mathbf{c}}_{\mathbf{r}}^T[\boldsymbol{\xi}(r)\otimes\boldsymbol{\xi}(\dot{r})] &= \dot{r}\ \mathbf{c}_{\mathbf{r}}^T[K_r\boldsymbol{\psi}(r)\otimes\boldsymbol{\xi}(\dot{r})] \\ &+ u_r\ \mathbf{c}_{\mathbf{r}}^T[\boldsymbol{\xi}(r)\otimes K_{\dot{r}}\boldsymbol{\psi}(\dot{r})] \\ &- \frac{\sigma_\alpha^2}{2}\ \mathbf{c}_{\mathbf{r}}^T[\boldsymbol{\xi}(r)\otimes K_{\dot{r}}^2\boldsymbol{\xi}(\dot{r})] \end{aligned} \tag{21}$$

Multiply $[\boldsymbol{\xi}(r)\otimes\boldsymbol{\xi}(\dot{r})]^T$ both sides and integrate over $r\in[a_r,b_r]$ and $\dot{r}\in[a_{\dot{r}},b_{\dot{r}}]$

$$\begin{aligned} \dot{\mathbf{c}}_{\mathbf{r}}(t) = \iint \Big\{ & [\boldsymbol{\xi}(r)\otimes\boldsymbol{\xi}(\dot{r})][K_r\boldsymbol{\psi}(r)\otimes\dot{r}\boldsymbol{\xi}(\dot{r})]^T \\ & + u_r\ [\boldsymbol{\xi}(r)\otimes\boldsymbol{\xi}(\dot{r})][\boldsymbol{\xi}(r)\otimes K_{\dot{r}}\boldsymbol{\psi}(\dot{r})]^T \\ & - \frac{\sigma_\alpha^2}{2}\ [\boldsymbol{\xi}(r)\otimes\boldsymbol{\xi}(\dot{r})][\boldsymbol{\xi}(r)\otimes K_{\dot{r}}^2\boldsymbol{\xi}(\dot{r})]^T \Big\}\, drd\dot{v}\ \mathbf{c}_{\mathbf{r}}(t) \end{aligned} \tag{22}$$

Using the following identities

$$(A\otimes B)(C\otimes D) = (AC)\otimes(BD) \tag{23a}$$

$$(A\otimes B)^T = A^T\otimes B^T \tag{23b}$$

where all dimensions are appropriate for the matrix multiplication,

$$\begin{aligned} \dot{\mathbf{c}}_{\mathbf{r}}(t) &= \Big\{ \int[\boldsymbol{\xi}(r)K_r\boldsymbol{\psi}^T(r)]dr \otimes \int[\dot{r}\boldsymbol{\xi}(\dot{r})\boldsymbol{\xi}^T(\dot{r})]d\dot{r} \\ &+ u_r\ I_N\otimes\int[\boldsymbol{\xi}(\dot{r})K_{\dot{r}}\boldsymbol{\psi}^T(\dot{r})]d\dot{r} - \frac{\sigma_\alpha^2}{2}\ I_N\otimes K_{\dot{r}}^2 \Big\}\ \mathbf{c}_{\mathbf{r}}(t) \\ &= A(u_r)\ \mathbf{c}_{\mathbf{r}}(t) \end{aligned} \tag{24}$$

where $A(u_r)$ is N^2-dimensional square matrix, which is defined by the terms inside the bracket in front of $\mathbf{c}(t)$ in the right hand side of the equation. The discrete time transition equation is given by

$$\mathbf{c}_{\mathbf{r}}(t_{k+1}) = \Phi(t_{k+1},t_k,u_r)\mathbf{c}_{\mathbf{r}}(t_k) \tag{25}$$

where

$$\Phi(t_{k+1},t_k,u_r) = e^{A(u_r)\Delta t_k}, \tag{26}$$

and $\Delta t_k = t_{k+1} - t_k$. Note that the transition matrix can be calculated off-line and stored in on-board computer for a finite set of samples of u_r in $[\underline{u}_r, \bar{u}_r]$.

The integrations in (24) have analytic expressions. For brevity, all the integrations from now on are assumed to be performed over the appropriate sampling space such that $s\in[a_s,b_s]$, where $s = r$, or $\dot{r}$, or indicated otherwise. For $i\geq 2$, $j\geq 2$ and $i\neq j$, the ith-row and jth-column element of

$$\left\{\int[\dot{r}\boldsymbol{\xi}(\dot{r})\boldsymbol{\xi}^T(\dot{r})]d\dot{r}\right\}_{ij\text{ or }ji} \tag{27}$$

$$= \begin{cases} 0, \text{ for both } i,j \text{ even or odd} \\ -\frac{2\Delta_{\dot{r}}}{\pi^2}\left[\frac{1}{(i-j)^2} + \frac{1}{(i+j-2)^2}\right], \text{ otherwise} \end{cases}$$

or for $i = 1$ and $j > 1$, i.e. the first row elements,

$$\left\{\int[\dot{r}\boldsymbol{\xi}(\dot{r})\boldsymbol{\xi}^T(\dot{r})]d\dot{r}\right\}_{1j} = \begin{cases} 0, \text{ for } j \text{ is odd} \\ \frac{-2\sqrt{2}\Delta_{\dot{r}}}{\pi^2(j-1)^2}, \text{ otherwise} \end{cases} \tag{28}$$

and the first column elements are of the same form, and for $i = j$ and $i\in[1,N]$,

$$\left\{\int[\dot{r}\boldsymbol{\xi}(\dot{r})\boldsymbol{\xi}^T(\dot{r})]d\dot{r}\right\}_{ii} = \frac{a_{\dot{r}}+b_{\dot{r}}}{2} \tag{29}$$

In addition, for all i in $[1,N]$,

$$\left\{\int[\boldsymbol{\xi}(s)K\boldsymbol{\psi}^T(s)]ds\right\}_{i1} = 0 \tag{30}$$

i.e, the first column elements are all zero, where s is equal to r or $\dot{r}$. The first row elements for $j\in[2,N]$ are equal to

$$\left\{\int[\boldsymbol{\xi}(s)K\boldsymbol{\psi}^T(s)]ds\right\}_{1j} = \begin{cases} 0, & \text{for } j \text{ odd} \\ \frac{2\sqrt{2}}{\Delta_s}, & \text{otherwise} \end{cases} \tag{31}$$

All diagonal terms, for $i = j$, are zero. Finally, for $i \neq j$ and $i, j \in [2, N]$,

$$\left\{ \int [\boldsymbol{\xi}(s) K \boldsymbol{\psi}^T(s)] ds \right\}_{ij} = \frac{j-1}{\Delta_s} \left[\frac{\cos(i-j)\pi - 1}{i-j} + \frac{\cos(i+j-3)\pi + 1}{i+j-2} \right]$$

and it becomes

$$\left\{ \int [\boldsymbol{\xi}(s) K \boldsymbol{\psi}^T(s)] ds \right\}_{ij} \tag{32}$$
$$= \frac{j-1}{\Delta_s} \begin{cases} 0, \text{ for both } i, j \text{ even or odd} \\ -\dfrac{2}{(i-j)} + \dfrac{2}{(i+j-2)}, \text{ otherwise} \end{cases}$$

The same form of update equations for y or z-axis, i.e., $\mathbf{c_y}(t)$ and $\mathbf{c_z}(t)$, can be derived from the same procedures.

2.5 Update: target position & velocity

The full joint pdf for all three Cartesian axes has to be considered to obtain an update algorithm. Let

$$\begin{aligned} p &= p(x, \dot{x}) p(y, \dot{y}) p(z, \dot{z}) \\ &= [\mathbf{c_x}^T(t) \boldsymbol{\phi}(x, \dot{x})][\mathbf{c_y}^T(t) \boldsymbol{\phi}(y, \dot{y})][\mathbf{c_z}^T(t) \boldsymbol{\phi}(z, \dot{z})] \end{aligned} \tag{33}$$

where the joint pdf is multiplication of all three joint pdfs for three axes, which are assumed to be statistically independent to each other.

Substitute (33) into the Bayesian update equation, (8),

$$\begin{aligned} &\left\{\mathbf{c_x}^T(t_k^+) \left[\boldsymbol{\xi}(x) \otimes \boldsymbol{\xi}(\dot{x})\right]\right\} \left\{\mathbf{c_y}^T(t_k^+) \left[\boldsymbol{\xi}(y) \otimes \boldsymbol{\xi}(\dot{y})\right]\right\} \\ &\left\{\mathbf{c_z}^T(t_k^+) \left[\boldsymbol{\xi}(z) \otimes \boldsymbol{\xi}(\dot{z})\right]\right\} \\ &= \eta_{\mathbf{x}}\ p(\mathbf{y}_k|\mathbf{r}) \left\{\mathbf{c_x}^T(t_k^-) \left[\boldsymbol{\xi}(x) \otimes \boldsymbol{\xi}(\dot{x})\right]\right\} \\ &\left\{\mathbf{c_y}^T(t_k^-) \left[\boldsymbol{\xi}(y) \otimes \boldsymbol{\xi}(\dot{y})\right]\right\} \left\{\mathbf{c_z}^T(t_k^-) \left[\boldsymbol{\xi}(z) \otimes \boldsymbol{\xi}(\dot{z})\right]\right\} \end{aligned} \tag{34}$$

where $\eta_{\mathbf{x}}$ is the normalising constant to be determined, and $\mathbf{r}$ includes all three coordinates, x, y, and z. Integrate both sides by all parameters except x and $\dot{x}$ and the left hand side (LHS) of the equation becomes

$$(\text{LHS}) = \mathbf{c_x}^T(t_k^+) \left[\boldsymbol{\xi}(x) \otimes \boldsymbol{\xi}(\dot{x})\right] \tag{35}$$

as the integrations of the other two brackets are simply equal to 1. In addition, the term inside the second bracket in the right hand side (RHS) of the update equation becomes

$$\mathbf{c_y}^T(t_k^-) \left[\boldsymbol{\xi}(y) \otimes \int_{a_{\dot{x}}}^{b_{\dot{y}}} \boldsymbol{\xi}(\dot{y}) d\dot{y} \right] \tag{36}$$
$$= \mathbf{c_y}^T(t_k^-) \left\{ \boldsymbol{\xi}(y) \otimes \begin{bmatrix} \sqrt{\Delta_{\dot{y}}} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\} = \sqrt{\Delta_{\dot{y}}} \mathbf{c}_y^T(t_k^-) \boldsymbol{\xi}(y)$$

where $\mathbf{c}_y(t_k^-)$ is the N-dimensional vector constructed by every Nth-element from the first to N^2-th elements in $\mathbf{c_y}(t_k^-)$. The similar steps can apply to the third term as follows:

$$\mathbf{c_z}^T(t_k^-) \left[\boldsymbol{\xi}(z) \otimes \int_{a_{\dot{z}}}^{b_{\dot{z}}} \boldsymbol{\xi}(\dot{z}) d\dot{z} \right] = \sqrt{\Delta_{\dot{z}}} \mathbf{c}_z^T(t_k^-) \boldsymbol{\xi}(z) \tag{37}$$

Hence,

$$\begin{aligned} \mathbf{c_x}^T(t_k^+) \left[\boldsymbol{\xi}(x) \otimes \boldsymbol{\xi}(\dot{x})\right] &= \eta \iint \left\{ p(\mathbf{y}_k|\mathbf{r}) [\mathbf{c}_y^T(t_k^-) \boldsymbol{\xi}(y)] \right. \\ &\left. [\mathbf{c}_z^T(t_k^-) \boldsymbol{\xi}(z)] \right\} dydz\ \left\{ \mathbf{c_x}^T(t_k^-) \left[\boldsymbol{\xi}(x) \otimes \boldsymbol{\xi}(\dot{x})\right] \right\} \end{aligned} \tag{38}$$

where $\sqrt{\Delta_{\dot{r}}}$ and $\sqrt{\Delta_{\dot{z}}}$ are absorbed in the normalising constant, $\eta_{\mathbf{x}}$.

Multiply $[\boldsymbol{\xi}(x) \otimes \boldsymbol{\xi}(\dot{x})]^T$ both sides and integrate over the sampling spaces, respectively,

$$\begin{aligned} \mathbf{c_x}^T(t_k^+) &= \iiiint \left\{ p(\mathbf{y}_k|\mathbf{r}) [\mathbf{c}_y^T(t_k^-) \boldsymbol{\xi}(y)] [\mathbf{c}_z^T(t_k^-) \boldsymbol{\xi}(z)] \right\} dydz \\ &\left\{ \mathbf{c_x}^T(t_k^-) \left[\boldsymbol{\xi}(x) \otimes \boldsymbol{\xi}(\dot{x})\right] \left[\boldsymbol{\xi}(x) \otimes \boldsymbol{\xi}(\dot{x})\right]^T \right\} dxd\dot{x}\ \eta_{\mathbf{x}} \end{aligned} \tag{39}$$

Again using the Kronecker product identity, the update equation becomes

$$\mathbf{c_x}(t_k^+) = \eta_{\mathbf{x}} B_{\mathbf{x}} \mathbf{c_x}(t_k^-) \tag{40}$$

where

$$\begin{aligned} B_{\mathbf{x}} = \iiint &\left\{ p(\mathbf{y}_k|\mathbf{r}) [\mathbf{c}_y^T(t_k^-) \boldsymbol{\xi}(y)] [\mathbf{c}_z^T(t_k^-) \boldsymbol{\xi}(z)] \right. \\ &\left. \left[\boldsymbol{\xi}(x) \boldsymbol{\xi}^T(x) \right] \otimes I_N \right\} dxdydz \end{aligned} \tag{41}$$

Note that $B_{\mathbf{x}}$ is symmetric.

The normalising constant is determined by the fact that the pdf must be equal to 1 after it is integrated over the sampling space.

$$\iint \mathbf{c_x}^T(t_k^+) \left[\boldsymbol{\xi}(x) \otimes \boldsymbol{\xi}(\dot{x})\right] dxd\dot{x} \tag{42}$$

$$= \eta_{\mathbf{x}}\ \mathbf{c_x}^T(t_k^-) B_{\mathbf{x}} \iint \left[\boldsymbol{\xi}(x) \otimes \boldsymbol{\xi}(\dot{x})\right] dxd\dot{x} = 1 \tag{43}$$

Then,

$$\eta_{\mathbf{x}} \mathbf{c_x}^T(t_k^-) B_{\mathbf{x}} \left\{ \begin{bmatrix} \sqrt{\Delta_x} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \sqrt{\Delta_{\dot{x}}} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\} = 1 \tag{44}$$

Therefore,

$$\eta_{\mathbf{x}} = \frac{1}{\sqrt{\Delta_x \Delta_{\dot{x}}} \left[\mathbf{c_x}^T(t_k^-) \mathbf{b}_{x1} \right]} \tag{45}$$

where $\mathbf{b}_{x1}$ is the first column of $B_{\mathbf{x}}$.

Similarly, the update equations for $\mathbf{c_y}(t_k^+)$ and $\mathbf{c_z}(t_k^+)$ are obtained as follows:

$$\mathbf{c_y}(t_k^+) = \eta_{\mathbf{y}} B_{\mathbf{y}} \mathbf{c_y}(t_k^-) \tag{46a}$$
$$\mathbf{c_z}(t_k^+) = \eta_{\mathbf{z}} B_{\mathbf{z}} \mathbf{c_z}(t_k^-) \tag{46b}$$

where

$$\begin{aligned} B_{\mathbf{y}} = \iiint &\left\{ p(\mathbf{y}_k|\mathbf{r}) [\mathbf{c}_x^T(t_k^-) \boldsymbol{\xi}(x)] [\mathbf{c}_z^T(t_k^-) \boldsymbol{\xi}(z)] \right. \\ &\left. \left[\boldsymbol{\xi}(y) \boldsymbol{\xi}^T(y) \right] \otimes I_N \right\} dxdydz \end{aligned} \tag{47a}$$

$$\begin{aligned} B_{\mathbf{z}} = \iiint &\left\{ p(\mathbf{y}_k|\mathbf{r}) [\mathbf{c}_x^T(t_k^-) \boldsymbol{\xi}(x)] [\mathbf{c}_y^T(t_k^-) \boldsymbol{\xi}(y)] \right. \\ &\left. \left[\boldsymbol{\xi}(z) \boldsymbol{\xi}^T(z) \right] \otimes I_N \right\} dxdydz \end{aligned} \tag{47b}$$

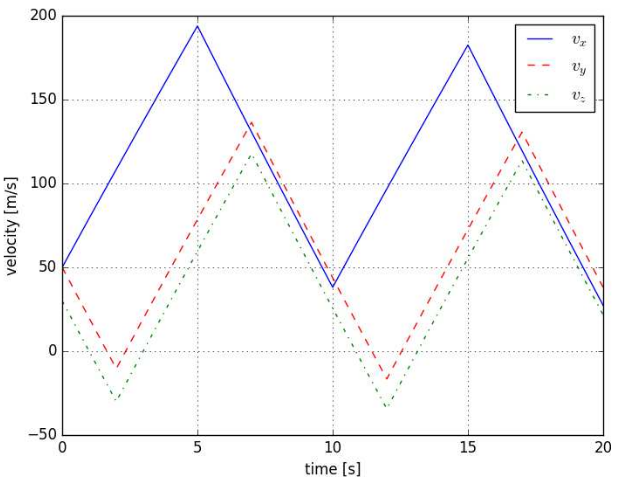

Fig. 2. Velocity history: each segment of the straight line corresponds to a 3g-turn.

and the normalising constant for each is defined by

$$\eta_{\mathbf{y}} = \frac{1}{\sqrt{\Delta_y \Delta_{\dot{y}}} \left[\mathbf{c}_{\mathbf{y}}^T(t_k^-) \mathbf{b}_{y1} \right]} \tag{48a}$$

$$\eta_{\mathbf{z}} = \frac{1}{\sqrt{\Delta_z \Delta_{\dot{z}}} \left[\mathbf{c}_{\mathbf{z}}^T(t_k^-) \mathbf{b}_{z1} \right]} \tag{48b}$$

3. NUMERICAL SIMULATIONS

Initially, a target is at $x = 2.0$km, $y = 0.7$km, and $z = 0$km with the initial velocity is equal to 50m/s, 50m/s and 30m/s in x, y, and z direction, respectively. The drag coefficient, τ, is equal to 0.01. The manoeuvre time constant, α^{-1}, is set to 10s. The variance, σ_v^2, is equal to 0.01 (Park, 1999). The acceleration input, u, is between ± 30 m/s^2.

Figure 2 shows the velocity histories of the target and the corresponding trajectory is shown in Figure 3 indicated by the solid blue line. The manoeuvre is composed of several 3g-turns during 20 seconds.

In the model for the projection filter, u_r is in the range of $\pm$ 33 m/s, i.e., 10% margin to the actual range of acceleration. The variance of the process noise in the model, σ_γ^2 is set to 0.01. The number of basis function for each state, N, is set to 15. Hence, the dimension of $A(u_r)$ for the propagation in the filter is 225 (=15×15). The acceleration input, u_r, in the model is not measured directly. Possibly choose several candidate u_r in the range and run multiple filters at the same time. On the other hand, physically the position and the velocity will be confined in an envelop by two extreme possible position and velocity trajectories, which are resulted from two extreme acceleration input, i.e., $u_r = -33$ or $u_r = 33$, respectively.

The sensor model is assumed to be of the normal distribution as follows:

$$p(\mathbf{y}_k|\mathbf{x}) = \frac{e^{-\frac{1}{2}[\mathbf{y}_k - \mathbf{h}(\mathbf{x})]^T R_k^{-1} [\mathbf{y}_k - \mathbf{h}(\mathbf{x})]}}{\sqrt{(2\pi)^3 |R_k|}},$$

where $R_k = \mathrm{E}(\mathbf{v}_k \mathbf{v}_k^T)$, which is assumed to be constant as follows (Park, 1999):

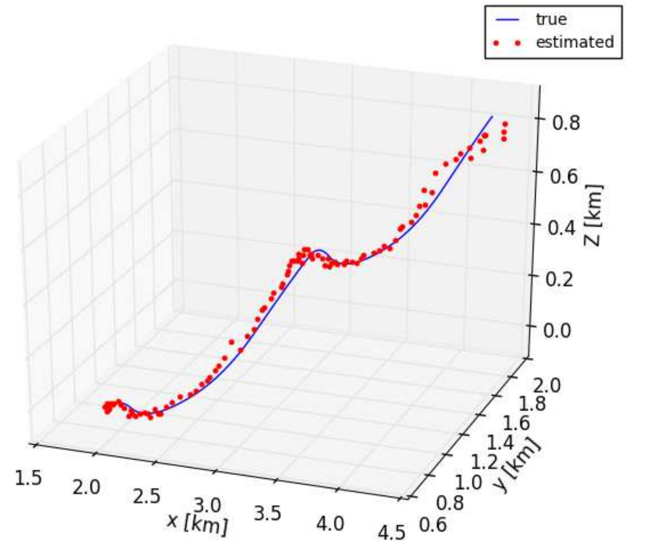

Fig. 3. Target trajectory in the Cartesian coordinates: the true in blue solid line and the estimated in red dots

$$R_k = \mathrm{diag}\left[(7.5\ [\mathrm{m}])^2\ \ (0.002\ [\mathrm{rad}])^2\ \ (0.002\ [\mathrm{rad}])^2 \right] \tag{49}$$

The measurement is available every 0.02s and this is relatively higher frequency compared to the given range of acceleration range. As a result, the position estimation based on the maximum acceleration and the one based on the minimum acceleration does not make any significant difference.

The mean value or the expected value of the position based on the estimation of joint pdf is shown in Figure 3 indicated in the red dots, where the mean value of the position is calculated as follows:

$$\mathrm{E}[x(t_k)] = \int_{a_x}^{b_x} x p(x) dx = \int_{a_x}^{b_x} \mathbf{c}_x(t_k^+) \boldsymbol{\xi}(x) dx \tag{50}$$

where $[a_x, b_x]$ is equal to $[1.0, 6.4]$km. Similarly, for the estimate of $y(t_k)$ and $z(t_k)$ are calculated, where $[a_y, b_y] = [0.35, 2.84]$km and $[a_z, b_z] = [0, 2.25]$km. The position estimation error in the average sense is shown in Figure 4. The maximum error occurs in x-direction, whose magnitude is about 250m.

Similarly, the mean estimation of velocity can be calculated but the velocity estimation error in the mean sense is in general very poor. The estimated joint pdf at $t_k = 20$s for the velocity is shown in Figure 5. Two joint pdfs for the velocity based on two different accelerations are very different. Unlike Kalman filter, which tracks only the first two moments, the projection filter tracks whole joint pdf as shown in Figure 5. Additional information from the joint pdf could be used to optimally select source of the measurements or place the position of sensors in order to improve the accuracy of the pdf.

4. CONCLUSIONS

For a typical target tracking estimation problem, an analytic expression for the propagation part of nonlinear projection filter is derived and higher dimensional expression can be obtained more efficiently without performing any numerical integration. The update part still requires some multi-dimensional integration in real-time. The per-

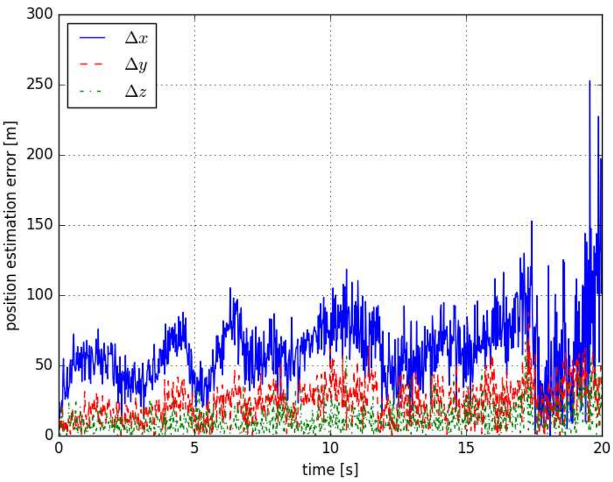

Fig. 4. Position estimation error for each direction

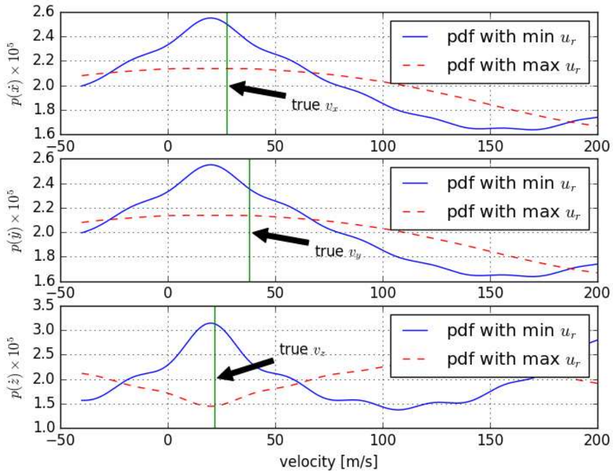

Fig. 5. Estimated velocity joint pdf with maximum or minimum acceleration assumption at $t_k = 20$s

formance of the algorithm is demonstrated with a realistic target tracking scenario where a radar sensor provides the range measurement and an optical sensor provides the azimuth and the elevation angles measurements. Thorough comparisons with existing nonlinear filtering algorithms would be performed in future to reveal possible advantages and disadvantages of the proposed filtering algorithm.

ACKNOWLEDGEMENTS

This research is supported by EPSRC Research Grant, EP/N010523/1, Balancing the impact of city infrastructure engineering on natural systems using robots. The author thank YongKyu Park in KARI (Korea Aerospace Research Institute) in Daejeon, South Korea for introducing the target tracking problem.

REFERENCES

Arulampalam, M.S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2), 174–188. doi:10.1109/78.978374. URL `http://dx.doi.org/10.1109/78.978374`.

Beard, R., Kenney, J., Gunther, J., Lawton, J., and Stirling, W. (1999). Nonlinear Projection Filter Based on Galerkin Approximation. *Journal of Guidance, Control, and Dynamics*, 22(2), 258–266. doi:10.2514/2.4403. URL `http://dx.doi.org/10.2514/2.4403`.

Benavidez, P. and Jamshidi, M. (2011). Mobile robot navigation and target tracking system. In *System of Systems Engineering (SoSE), 2011 6th International Conference on*, 299–304. doi:10.1109/SYSOSE.2011.5966614.

Chen, S. and Huang, W. (2013). Target tracking using particle filter with x-band nautical radar. In *Radar Conference (RADAR), 2013 IEEE*, 1–6. doi:10.1109/RADAR.2013.6585993.

Gray, J. and Murray, W. (1993). A derivation of an analytic expression for the tracking index for the alpha-beta-gamma filter. *Aerospace and Electronic Systems, IEEE Transactions on*, 29(3), 1064–1065. doi:10.1109/7.220956.

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.J. (2002). Particle filters for positioning, navigation, and tracking. *Signal Processing, IEEE Transactions on*, 50(2), 425–437. doi:10.1109/78.978396.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82, 35–45.

Li, X. and Jilkov, V. (2003). Survey of maneuvering target tracking. part i. dynamic models. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(4), 1333–1364. doi:10.1109/TAES.2003.1261132.

Park, Y.K. (1999). *An improvement of target tracking algorithm using multiple model adaptive estimator*. MSc Thesis, Aerospace Engineering, Inha University, Incheon, Republic of Korea, in Korean.

Rogers, S. (1987). Alpha-beta filter with correlated measurement noise. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-23(4), 592–594. doi:10.1109/TAES.1987.310893.

Singer, R.A. (1970). Estimating optimal tracking filter performance for manned maneuvering targets. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-6(4), 473–483. doi:10.1109/TAES.1970.310128.

Single-Liertz, T., Kim, J., and Richardson, R. (2015). Nonlinear projection filter with parallel algorithm and parallel sensors. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, 2432–2437. doi:10.1109/CDC.2015.7402572.

Spingarn, K. and Weidemann, H.L. (1972). Linear regression filtering and prediction for tracking maneuvering aircraft targets. *IEEE Transactions on Aerospace and Electronic Systems*, AES-8(6), 800–810. doi:10.1109/TAES.1972.309612.

Tenne, D. and Singh, T. (2002). Characterizing performance of α-β-γ filters. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(3), 1072–1087.