Tracking of a ballistic missile with a-priori information

A. Benavoli[1], *L. Chisci*[1] *and A. Farina*[2]

[1] DSI, Università di Firenze, Firenze, Italy
e-mail: {benavoli,chisci}@dsi.unifi.it
[2] Engineering Division, SELEX - Sistemi Integrati S.p.A., Rome, Italy
e-mail: afarina@selex-si.com

Abstract

The paper addresses the problem of estimating the launch and impact points of a ballistic target from radar measurements. The problem has been faced under different hypotheses on the available prior knowledge. The proposed approach combines a nonlinear batch estimator with a recursive MM (Multiple Model) particle filter in order to attain the estimation goal. Extensive simulations assess the achievable estimation performance.

Keywords: ballistic missile tracking; parameter estimation; state estimation; nonlinear least-squares; particle filter.

1 Introduction

The problem of tracking ballistic objects in the reentry phase has attracted great attention in the literature for both theoretical and practical reasons. From a technical point of view, the problem is to set up a stochastic nonlinear filter due to the nonlinearity of the dynamic state equations of the target. Tracking filters have been conceived for this purpose starting from the seventies [1, 2, 3] until more recent work, e.g. [4, 5, 6, 7]. Practical applications are in the field of surveillance for defense and for safety against the reentry of old satellites. The specific objective of this paper is to estimate in a timely and accurate way the impact, and possibly the launch, point of a tactical ballistic missile using radar measurements. This represents an extremely difficult task for a number of reasons listed hereafter.

1. The radar scan period is long (e.g., 10-12 seconds for a 3D ground-based radar) and the target flight time is so short that there are few available measurements.

2. The dynamics of the target motion are highly nonlinear and time-varying. Further they undergo an abrupt change of model at an unknown time instant, called burnout time, whenever the thrusters are switched off.

3. The target's trajectory is highly sensitive to some parameters, depending on the type of target, which cannot be known a-priori.

4. The time displacements of the measurements with respect to the launch time can be unknown.

In this scenario, it is of paramount importance to devise an ad-hoc estimation procedure which exploits in an optimal way the available data and a-priori knowledge. A-priori information can be available on the launch position, launch time and/or on the burnout time (e.g. provided by an infrared satellite). Further, a-priori information could concern the target's characteristics (e.g. initial mass, mass burn rate, etc.). In this work, the ballistic estimation problem has been tackled and solved under different hypotheses on the available prior knowledge. The problem has been formulated as a nonlinear optimization and a-priori information has been exploited in terms of equality and/or inequality constraints on the variables to be estimated (i.e. parameters and state variables) as well as in the choice of the guess values for the optimum search. The nonlinear optimization has been solved according to a batch approach, provided that it is known when the burnout occurs. Conversely, a recursive filter is used to detect the burnout and is combined with the batch estimator in order to provide estimates of the launch and impact points. It is worth pointing out that, with respect to the existing literature [4, 5, 6, 7] on the tracking of ballistic targets, the contribution of this work has been in two directions:

1. to take into account the time-varying and switching nature of the target's dynamics and still be able to exploit all the available measurements despite of the lack of knowledge on the switching time;

2. to incorporate constraints in the estimation process.

The rest of the paper is organized as follows. Section 2 describes in detail the model of the dynamics of a ballistic target. Section 3 performs a sensitivity analysis in order to quantify how much the various parameters affect the target's trajectory. Section 4 discusses the various types of available a-priori information. Section 5 presents a nonlinear batch estimation algorithm for the target's motion assuming knowledge of the burnout temporal location. Section 6 presents a recursive algorithm, based on the particle filter, for burnout detection. Section 7 analyzes the achievable performance via simulation experiments. Finally, section 8 concludes the paper.

Notation

Throughout the paper the following notation will be adopted.

- Boldface letters denote vectors and matrices.
- Prime denotes transpose.
- $\mathbf{v} \sim \mathcal{N}(\mathbf{m_v}, \mathbf{\Sigma_v})$ means that the random variable $\mathbf{v}$ is normally distributed with mean $\mathbf{m_v}$ and covariance $\mathbf{\Sigma_v}$ and $\mathcal{N}(\cdot; \mathbf{m_v}, \mathbf{\Sigma_v})$ is the relative PDF (*Probability Density Function*).
- $\mathbf{v} \sim \mathcal{U}_{\overline{\mathbf{D}}}$ means that the random variable $\mathbf{v}$ is uniformly distributed in $\overline{\mathbf{D}}$ and $\mathcal{U}_{\overline{\mathbf{D}}}(\cdot)$ is the relative PDF.

2 Ballistic flight modeling

The flight of a ballistic object consists of three phases [8]:

Boost: flight phase between the *launch* and the *burnout*, i.e. the turn-off of thrusters;

Free-Flight: flight phase, outside the earth atmosphere, in which the object follows a Keplerian orbit;

Reentry: flight phase from the reentry into the atmosphere until the impact on the ground.

During these three phases, different forces are acting on the object: gravity, thrust and aerodynamic drag in the *Boost* phase; only gravity during the *Free-Flight*; gravity and aerodynamic drag during the *Reentry*.
In order to get a mathematical model describing the target's motion, it is convenient to adopt the ECEF (Earth Centered Earth Fixed) coordinate system [8]. This reference system is centered in the Earth center, has z axis parallel to the terrestrial rotation axis, x and y axes lying in the equatorial plane with x axis pointing towards the Greenwich meridian. It is also assumed that the Earth is perfectly spherical. This reference system is clearly non inertial, i.e. it rotates along with the Earth at the angular speed $\omega = 7.2921159 \cdot 10^{-5}$ rad/s; for this reason, the apparent forces acting on the target, i.e. centrifugal and Coriolis forces, must be considered. Introducing as state variables the Cartesian position $\mathbf{p} = [p_x, p_y, p_z]'$ and, respectively, velocity coordinates $\mathbf{v} = [v_x, v_y, v_z]'$ in the ECEF system, the state equations describing the target motion in the boost phase are the following:

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{a}_T + \mathbf{a}_D + \mathbf{a}_G + \mathbf{a}_C \end{cases} \tag{2.1}$$

where the total acceleration acting on the target has been decomposed into four terms: thrust acceleration $\mathbf{a}_T$, drag acceleration $\mathbf{a}_D$, gravitational acceleration $\mathbf{a}_G$ and apparent

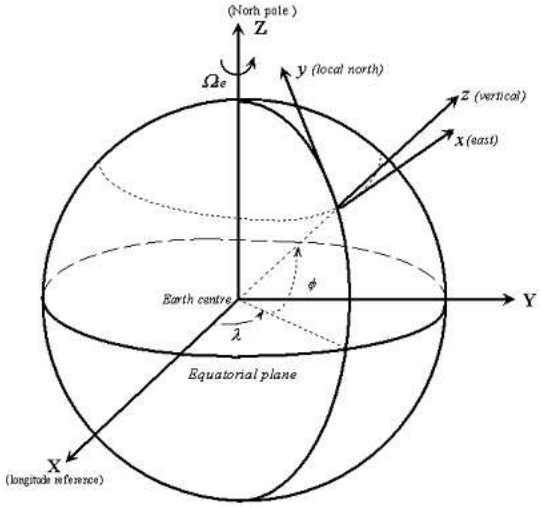

Figure 1: ECEF and ENU coordinate systems

acceleration $\mathbf{a}_C$. The thrust acceleration acts along the target's longitudinal axis[1] and its magnitude is [8]

$$a_T(t) = \frac{g\ I_{sp}\ \dot{m}(t)}{m(t)} \tag{2.2}$$

where $m(t)$ is the target's mass, $g = 9.81\ m\ s^{-2}$ is the gravitational acceleration, I_{sp} is the *specific impulse* (expressed in seconds) and $\dot{m}(t)$ is the mass burn rate. Assuming that the specific impulse is constant and the target mass $m(t)$ decreases linearly w.r.t. time t at a constant rate $\dot{m}(t) = \dot{m}$ (i.e. if $m(0)$ is the target's mass at the launch time, then $m(t) = m(0) - \dot{m}\ t$ for $t > 0$), it follows that the time dependence of the thrust acceleration can be expressed as

$$a_T(t) = \frac{n\ g}{1 - q\ t} \tag{2.3}$$

where

$$\begin{aligned} n &= I_{sp}\ q \text{ is the initial thrust-to-weight ratio;} \\ q &= \frac{\dot{m}}{m(0)} \text{ is the normalized mass burn rate } [s^{-1}]; \end{aligned} \tag{2.4}$$

The drag acceleration acts opposite to the target velocity $\mathbf{v}$ and its magnitude is given by [8]

$$a_D(t) = \frac{c_D(v(t))\ S\ \rho(h(t))\ v^2(t)}{2\ m(t)} \tag{2.5}$$

[1]It has been assumed that the target is in the *gravity turn* guidance phase, viz. the thrust acceleration is parallel to the target's longitudinal axis, i.e. to the velocity vector.

where: $v(t) = \sqrt{v_x^2(t) + v_y^2(t) + v_z^2(t)}$ is the magnitude of the target's velocity at time t; $h(t) = \sqrt{p_x^2(t) + p_y^2(t) + p_z^2(t)} - R_T$ is the target's altitude above the sea level at time t, $R_T = 6378173\ m$ being the Earth's radius; S is the reference area (defined as the target body cross-sectional area perpendicular to the velocity); $c_D(v)$ is the drag coefficient as a function of the velocity magnitude v and $\rho(h)$ represents the air's density as a function of the altitude h:

$$\rho(h) = \rho_0\ e^{-kh}$$

with $\rho_0 = 1.22\ kg/m^3$ and $k = 0.14141 \cdot 10^{-3}\ m^{-1}$. Since the quantity $c_D(v(t))S/m(t)$ turns out to be nearly constant, introducing the ballistic coefficient $\beta \triangleq \dfrac{m(t)}{c_D(v(t))\ S}$, the equation (2.5) can be rewritten in the following way

$$a_D(t) \approx \frac{\rho(h(t))\ v^2(t)}{2\ \beta} \tag{2.6}$$

The gravitational acceleration points from the target to the Earth's center and its magnitude is given by the Newton's law of universal gravitation [8]

$$a_G(t) = \frac{\mu_G}{p^2(t)} \tag{2.7}$$

where $\mu_G = 3.985325 \cdot 10^{14}\ N\ m^2\ kg^{-1}$ and $p(t) = \sqrt{p_x^2(t) + p_y^2(t) + p_z^2(t)}$ is the distance of the target from the Earth's center at time t.

The apparent acceleration is obtained as a sum of two terms [8]:

$$\textbf{Coriolis acceleration:} \quad -2\ \boldsymbol{\omega} \wedge \mathbf{v}(t) \tag{2.8}$$

$$\textbf{centrifugal acceleration:} \quad -\ \boldsymbol{\omega} \wedge (\boldsymbol{\omega} \wedge \mathbf{p}(t)) \tag{2.9}$$

where $\wedge$ indicates vector product and $\boldsymbol{\omega}$ is the Earth's angular velocity vector which, in the ECEF system, turns out to be $\boldsymbol{\omega} = [0, 0, \omega]'$, ω being the constant Earth's angular speed.

Finally, expanding the vector product in (2.8)-(2.9) and gathering the Cartesian components of the thrust, drag, gravitational and apparent accelerations, the state equations

(2.1) take the following componentwise form

$$\begin{cases} \dot{p}_x = v_x \\ \dot{p}_y = v_y \\ \dot{p}_z = v_z \\ \dot{v}_x = \dfrac{g\ n}{(1-q\ t)}\dfrac{v_x}{\sqrt{v_x^2+v_y^2+v_z^2}} - \dfrac{\rho(h)}{2\beta}\sqrt{v_x^2+v_y^2+v_z^2}\ v_x - \dfrac{\mu_G}{\left(\sqrt{p_x^2+p_y^2+p_z^2}\right)^3}\ p_x \\ \qquad + (2\ \omega\ v_y + \omega^2\ p_x) \\ \dot{v}_y = \dfrac{g\ n}{(1-q\ t)}\dfrac{v_y}{\sqrt{v_x^2+v_y^2+v_z^2}} - \dfrac{\rho(h)}{2\beta}\sqrt{v_x^2+v_y^2+v_z^2}\ v_y - \dfrac{\mu_G}{\left(\sqrt{p_x^2+p_y^2+p_z^2}\right)^3}\ p_y \\ \qquad + (-2\ \omega\ v_x + \omega^2\ p_y) \\ \dot{v}_z = \dfrac{g\ n}{(1-q\ t)}\dfrac{v_z}{\sqrt{v_x^2+v_y^2+v_z^2}} - \dfrac{\rho(h)}{2\beta}\sqrt{v_x^2+v_y^2+v_z^2}\ v_z - \dfrac{\mu_G}{\left(\sqrt{p_x^2+p_y^2+p_z^2}\right)^3}\ p_z \end{cases} \tag{2.10}$$

In order to simplify the notation, the time dependence of the state variables and also the dependence of the altitude h from the state variables p_x, p_y, p_z have been left out. The terms appearing in the right-hand-sides of $\dot{v}_x, \dot{v}_y, \dot{v}_z$ represent respectively: (1) the thrust acceleration, (2) the drag acceleration, (3) the gravitational acceleration and (4) the apparent (centrifugal plus Coriolis) acceleration, the latter not being present in $\dot{v}_z$. It must be pointed out that at a given time t_b, called *burnout* time, the thrusters are switched off since either the fuel is over or the desired orbit has been attained. The state equations describing the target's motion after the burnout, i.e. for $t > t_b$, are the following:

$$\begin{cases} \dot{p}_x = v_x \\ \dot{p}_y = v_y \\ \dot{p}_z = v_z \\ \dot{v}_x = -\dfrac{\rho(h)}{2\beta}\sqrt{v_x^2+v_y^2+v_z^2}\ v_x - \dfrac{\mu_G}{\left(\sqrt{p_x^2+p_y^2+p_z^2}\right)^3}\ p_x + (2\ \omega\ v_y + \omega^2\ p_x) \\ \dot{v}_y = -\dfrac{\rho(h)}{2\beta}\sqrt{v_x^2+v_y^2+v_z^2}\ v_y - \dfrac{\mu_G}{\left(\sqrt{p_x^2+p_y^2+p_z^2}\right)^3}\ p_y + (-2\ \omega\ v_x + \omega^2\ p_y) \\ \dot{v}_z = -\dfrac{\rho(h)}{2\beta}\sqrt{v_x^2+v_y^2+v_z^2}\ v_z - \dfrac{\mu_G}{\left(\sqrt{p_x^2+p_y^2+p_z^2}\right)^3}\ p_z \end{cases} \tag{2.11}$$

Notice that in (2.11), the drag acceleration term vanishes exponentially with increasing altitude h and, hence, (2.11) describes both the free-flight and reentry phases. The flying object's trajectory is, therefore, completely specified by the initial state variables $p_x(0)$, $p_y(0)$, $p_z(0)$, $v_x(0)$, $v_y(0)$, $v_z(0)$ as well as from the values of the four parameters β, n, q and t_b.

The ballistic target is observed from a radar station located at given latitude ϕ, longitude λ and altitude h_s. The radar measures - with a sampling time T - the polar coordinates (i.e. range r, azimuth α and elevation ϵ) of the target with respect to the ENU (East North Up) reference system. The ENU system is centered in the radar location and has $\bar{x}$ axis pointing to East, $\bar{y}$ axis pointing to North and $\bar{z}$ axis parallel to the vertical. Measurements are affected by noises that can be modeled as mutually uncorrelated Gaussian white noises with zero means and standard deviations σ_r, σ_α and σ_ϵ respectively for the range, azimuth and elevation. The measurement equation is therefore:

$$\begin{cases} r = \sqrt{\bar{p}_x^2 + \bar{p}_y^2 + \bar{p}_z^2} + \eta_r \\ \alpha = \angle(\bar{p}_y + j\ \bar{p}_x) + \eta_\alpha \\ \epsilon = \angle(\sqrt{\bar{p}_x^2 + \bar{p}_y^2} + j\ \bar{p}_z) + \eta_\epsilon \end{cases} \tag{2.12}$$

where j denotes the imaginary unit and η_r, η_α, η_ϵ are the above mentioned measurement noises. In order to completely specify the measurement equation (2.12), the relationships between ENU and ECEF coordinates must be exploited. The coordinates of the origin of the ENU system (i.e. the radar's position) in the ECEF system are given by:

$$\mathbf{O}_{ENU}^{ECEF} = \begin{bmatrix} (R_T + h_s)\cos(\phi)\cos(\lambda) \\ (R_T + h_s)\cos(\phi)\sin(\lambda) \\ (R_T + h_s)\sin(\phi) \end{bmatrix}. \tag{2.13}$$

Further, the rotation matrix from ECEF to ENU coordinates is:

$$\mathbf{M}(\lambda, \phi) = \begin{bmatrix} -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(\lambda)\sin(\phi) & -\sin(\phi)\sin(\lambda) & \cos(\phi) \\ \cos(\lambda)\cos(\phi) & \cos(\phi)\sin(\lambda) & \sin(\phi) \end{bmatrix} \tag{2.14}$$

Hence, the transformation relating the two coordinate vectors is:

$$\bar{\mathbf{p}} = \mathbf{M}(\lambda, \phi)\left[\ \mathbf{p} - \mathbf{O}_{ENU}^{ECEF}\right] \tag{2.15}$$

where $\bar{\mathbf{p}} = [\bar{p}_x,\ \bar{p}_y,\ \bar{p}_z]'$ and $\mathbf{p} = [p_x,\ p_y,\ p_z]'$ are the position vectors in the ENU and, respectively, ECEF coordinate systems.

Summing up, the overall model of the ballistic flight can be compactly represented by the following state equations:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \boldsymbol{\theta}) \tag{2.16}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\eta} \tag{2.17}$$

where: $\mathbf{x} = [p_x, p_y, p_z, v_x, v_y, v_z]' \in \mathbb{R}^n$ $(n = 6)$ is the target's state vector; $\mathbf{y} = [r, \alpha, \epsilon]' \in \mathbb{R}^m$ $(m = 3)$ is the measurement vector; $\boldsymbol{\theta} = [\beta, n, q, t_b]' \in \mathbb{R}^p$ $(p = 4)$ is a vector of parameters; $\boldsymbol{\eta} = [\eta_r, \eta_\alpha, \eta_\epsilon]' \in \mathbb{R}^m$ is the measurement noise; $\mathbf{f}(\cdot,\cdot,\cdot)$ and $\mathbf{h}(\cdot)$ are nonlinear functions. Notice that the model (2.16)-(2.17), besides being nonlinear both in the state and in the measurement equation, is also time-varying in the state equation. In fact, time-variance arises from the following two facts:

1. the presence of $1 - qt$, which takes into account the linear decrease of the target's mass implied by fuel consumption in the boost stage, in the denominator of the boost acceleration in (2.10);

2. the time-driven change of state equations from (2.10) for $t \leq t_b$ to (2.11) for $t > t_b$

Notice that in order to estimate the motion of the ballistic target (e.g. determine the launch point and/or predict the impact point and time) one has to estimate the states $\mathbf{x}$ (positions and velocities) at a given time instant as well as the parameters $\boldsymbol{\theta}$ that characterize the type of target. For the subsequent estimation developments, it will be assumed that N noisy measurements $\{y(t_i), i = 0, 1, \ldots, N-1\}$ are available at equally spaced time instants

$$t_i = t_0 + iT \quad i = 0, 1, \ldots, N-1$$

where T is the sampling period and $t = 0$ corresponds to the launch time. A further problem is that t_0, i.e. the time elapsing from the launch until the time of the first measurement, may be unknown. Hence, in order to determine the target's trajectory, the objective is to estimate the vector $\boldsymbol{\Theta} = [\mathbf{x}(\tau)',\ \boldsymbol{\theta}', t_0]'$, i.e. the target's state $\mathbf{x}(\tau)$ at some reference time τ (e.g. the launch time $\tau = 0$ or the first measurement time $\tau = t_0$), the target's parameters $\boldsymbol{\theta}$ and the time t_0.

3 Sensitivity analysis

Sensitivity analysis is an important tool to assess the influence of the various parameters on the trajectory of the ballistic object. Considering the dynamical system (2.16)-(2.17), the parametric sensitivity $\mathbf{S}_i^{\mathbf{x}}$ of the state vector $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]'$ w.r.t. the i-th entry θ_i of the parameter vector $\boldsymbol{\theta}$, is defined as:

$$\mathbf{S}_i^{\mathbf{x}} \triangleq \frac{\partial \mathbf{x}}{\partial \theta_i} \qquad i = 1, 2, \ldots, p \tag{3.1}$$

Since $\mathbf{x} = \mathbf{x}(t)$ is the state of a dynamical system, its sensitivity $\mathbf{S}_i^{\mathbf{x}} = \mathbf{S}_i^{\mathbf{x}}(t)$ actually depends on time t. Therefore, exploiting the state equations (2.16)-(2.17) and differentiating (3.1) w.r.t. time t, one obtains the following *sensitivity system* around a given nominal state evolution $\mathbf{x}^o(t)$:

$$\begin{cases} \dot{\mathbf{S}}_i^{\mathbf{x}}(t) &= \left.\dfrac{\partial f(t, \mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^o(t)} \mathbf{S}_i^{\mathbf{x}}(t) \quad + \quad \dfrac{\partial f(t, \mathbf{x}^o(t), \boldsymbol{\theta})}{\partial \theta_i} \\ \mathbf{S}_i^{\mathbf{y}}(t) &= \left.\dfrac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^o(t)} \mathbf{S}_i^{\mathbf{x}}(t) \end{cases} \tag{3.2}$$

that provides a way to calculate the time evolution of the sensitivity $\mathbf{S}_i^{\mathbf{x}}$ for the state and $\mathbf{S}_i^{\mathbf{y}} \triangleq \partial \mathbf{y} / \partial \theta_i$ for the measurement vector. Notice that actually two sets of *ordinary*

differential equations (ODEs) must be solved in order to find the sensitivities according to (3.2):

- first the ODE (2.16), with initial condition $\mathbf{x}^o(0)$, must be solved in order to provide the nominal state evolution $\mathbf{x}^o(t)$;
- next the sensitivity ODE (3.2), with the previously obtained nominal state evolution $\mathbf{x}^0(t)$, must be solved from the initial condition $\mathbf{S}_i^{\mathbf{x}}(0) = \mathbf{0}$ to yield $\mathbf{S}_i^{\mathbf{x}}(t)$ and $\mathbf{S}_i^{\mathbf{y}}(t)$.

To avoid the computation of the Jacobian matrix $\partial f(t, \mathbf{x}, \boldsymbol{\theta})/\partial \mathbf{x}$ in (3.2) and the solution of the sensitivity ODE (3.2), the sensitivity (3.1) can be approximated with the incremental sensitivity

$$\mathbf{S}_i^{\mathbf{x}} = \begin{bmatrix} S_{i1}^{\mathbf{x}} \\ S_{i2}^{\mathbf{x}} \\ \vdots \\ S_{in}^{\mathbf{x}} \end{bmatrix}, \qquad S_{ij}^{\mathbf{x}} \approx \frac{x_j^p - x_j^o}{x_j^o} \frac{\theta_i^o}{\Delta\theta_i} \tag{3.3}$$

where x_j^o is the j-th entry of the nominal state vector corresponding to a nominal value of the parameter θ_i^o and x_j^p is the j-th entry of the perturbed state vector obtained increasing the value of the parameter θ_i^o of the quantity $\Delta\theta_i$. This approximation turns out to be sufficiently accurate if a small parameter perturbation $\Delta\theta_i = \theta_i^p - \theta_i^o$ is chosen. In particular, in this paper, a 1% increment, i.e. $\Delta\theta_i = 0.01\ \theta_i^0$, has been used. From (3.3) it follows that (for $i = 1, 2, \ldots, p$):

$$S_{ij}^{\mathbf{x}}(t) \approx \frac{x_j^p(t) - x_j^o(t)}{x_j^o(t)}\ 100 \qquad \text{for } j = 1, 2, \ldots, n \tag{3.4}$$

$$S_{ik}^{\mathbf{y}}(t) \approx \frac{y_k^p(t) - y_k^o(t)}{y_k^o(t)}\ 100 \qquad \text{for } k = 1, 2, \ldots, m \tag{3.5}$$

where: $S_{ij}^{\mathbf{x}}(t)$ is the sensitivity of the j-th entry $x_j(t)$ of the state vector $\mathbf{x}(t)$ w.r.t the i-th parameter; $S_{ik}^{\mathbf{y}}(t)$ is the sensitivity of the k-th entry $y_k(t)$ of the measurement vector $\mathbf{y}(t)$ w.r.t the i-th parameter; $\mathbf{x}^o(t)$, $\mathbf{y}^o(t)$ and $\mathbf{x}^p(t)$, $\mathbf{y}^p(t)$ are the nominal and, respectively, perturbed responses obtained via solution of the following state equations

$$\mathbf{x}^o(0) = \mathbf{x}^p(0) \tag{3.6}$$

$$\dot{\mathbf{x}}^o(t) = \mathbf{f}(t, \mathbf{x}^o(t), \boldsymbol{\theta}^o) \tag{3.7}$$

$$\dot{\mathbf{x}}^p(t) = \mathbf{f}(t, \mathbf{x}^p(t), \boldsymbol{\theta}^p) = \mathbf{f}(t, \mathbf{x}^p(t), \boldsymbol{\theta}^o + 0.01\ \theta_i^o\ \mathbf{e}_i) \tag{3.8}$$

$$\mathbf{y}^o(t) = \mathbf{h}(\mathbf{x}^o(t)) \tag{3.9}$$

$$\mathbf{y}^p(t) = \mathbf{h}(\mathbf{x}^p(t)) \tag{3.10}$$

where $\mathbf{e}_i$ denotes a vector with 1 in the i-th entry and the remaining entries equal to zero.

In fig.2 the incremental sensitivity of the range component of the measurement vector w.r.t. the 1% perturbation of the parameters is depicted. The nominal value of the

True missile parameters			
n	2.67	t_b	80 s
q	0.00834	t_0	30 s
β	6116	T	12 s
Trajectory			
apogee	202 km	range	669 km
flight time	482 s		

Table 1: Nominal parameters

missile parameters and the characteristics of the missile trajectory can be found in table 1. From the figure 2, it can be seen that the parameters n, q and t_b are characterized by high sensitivity, whereas β has low sensitivity. This means that the influence of β on the trajectory of the ballistic object is negligible w.r.t. the other parameters and therefore β is critical to be estimated as will be explained in the next section. The sensitivity analysis considers the influence of the variation of a single parameter on the trajectory of the ballistic object, however also the mutual interaction among the parameters should be take into account in order to assess the structural identifiability of the parameters. The parameter t_0, which from the sensitivity analysis in fig. 2 is characterized by a medium-low sensitivity, is nevertheless not structurally identifiable (the radar measurements are not able to provide directly or indirectly the launch time). In the next sections possible solutions to manage the low sensitivity of β and the lack of identifiability of t_0 will be discussed.

4 A-priori information

In order to estimate the trajectory of a ballistic object, a-priori information on the target and launch parameters may be available. For example information on launch time, launch position and/or burnout time can be provided by Infrared (IR) sensors mounted on strategically located surveillance satellites. In fact, during the boost phase the fuel exhaust generated by a ballistic missile makes it visible to IR sensors. Conversely, information on the target's parameters n, q and β can be available from the prior knowledge on the missile type. This a-priori information can be incorporated in the estimation algorithm in the form of equality and/or inequality constraints.

Equality constraints can be imposed on launch time t_0, launch position $\mathbf{p}(0)$ and/or burnout time t_b, assuming that the errors deriving from the finite resolution of IR sensors and the uncertainty on the location and synchronization of the satellites can be neglected. On the other hand, inequality constraints, in the form of lower and upper bounds, characterize in a better way the uncertainty on the knowledge of the missile parameters, i.e.

$$n \in [\underline{n},\ \overline{n}], \quad q \in [\underline{q},\ \overline{q}], \quad \beta \in [\underline{\beta},\ \overline{\beta}].$$

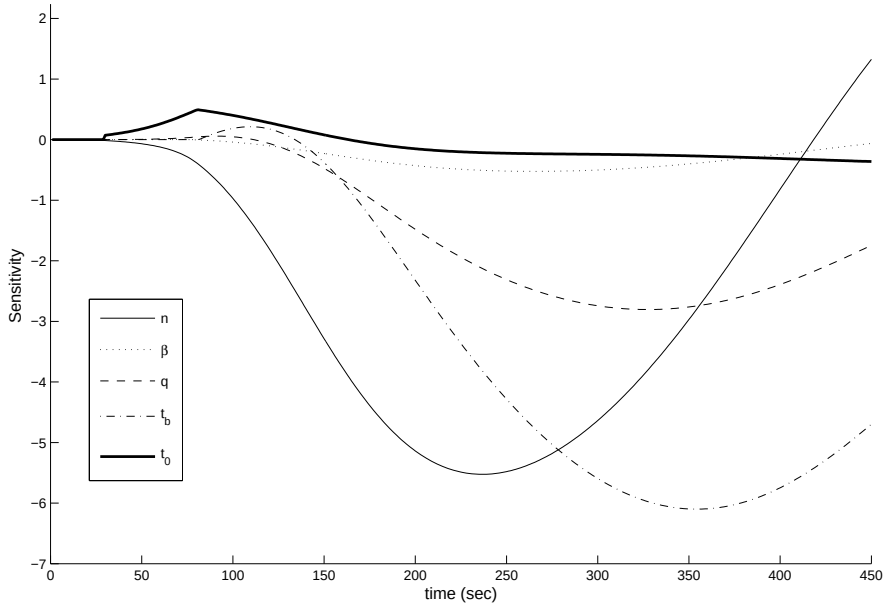

Figure 2: Sensitivity of the range of the measurement vector w.r.t. the variation of 1% of the parameters

These bounds can model two kinds of uncertainty:

1. the presence of different types of missiles;
2. the lack of knowledge of the exact values of the parameters for each missile type.

Taking the first point into account, the intervals should include all possible values of the parameters for different types of missiles. Unfortunately, this solution produces too large intervals and, therefore, a "scarce" a-priori information. Provided that sufficient data and/or information concerning the various types of targets were available, this problem could be overcome by means of a *grid-search* approach operating as follows.

1. For each type of missile, a bounding interval is defined for each parameter using the available knowledge base. In this case, the width of the intervals only accounts for the parameter uncertainty.
2. Based on the available on-line measurements, an estimate is obtained for each type of missile using the relative bounding intervals (*grid-point*).
3. The estimate which best fits the measurements is selected.

In the next section, the estimation problem will be addressed assuming uncertainty on the target's parameters but knowledge of the target type.

5 Motion estimation

The aim is to estimate the motion of the ballistic target using the N noisy measurements provided by the radar and the available a-priori information. Assuming that the measurement noise $\boldsymbol{\eta}$ in (2.17) is a white, zero-mean, Gaussian sequence with known covariance matrix $\mathbf{R}$, the measurements $\{\mathbf{y}(t_i), i = 0, 1, \ldots, N-1\}$ turn out to be mutually independent. Hence, the conditional PDF of the measurement vector $\mathbf{Y} = [\mathbf{y}'(t_0), \mathbf{y}'(t_1), \ldots, \mathbf{y}'(t_{N-1})]'$ w.r.t. the vector $\boldsymbol{\Theta}$ to be estimated is given by [10]-[11]

$$p(\mathbf{Y}|\boldsymbol{\Theta}) = \prod_{i=0}^{N-1} p(\mathbf{y}(t_i)|\boldsymbol{\Theta}) \tag{5.1}$$

$$= \prod_{i=0}^{N-1} \frac{1}{\sqrt{det(2\pi\mathbf{R})}} \exp\left(-\frac{\|\mathbf{y}(t_i) - \mathbf{h}(\boldsymbol{\varphi}(t_i, \tau, \boldsymbol{\Theta}))\|^2_{\mathbf{R}^{-1}}}{2}\right) \tag{5.2}$$

where $\mathbf{h}(\cdot)$ is the measurement function in (2.17) and $\boldsymbol{\varphi}(t_i, \tau, \boldsymbol{\Theta})$ is the solution of the ODE (2.16) at time t_i initialized from time τ. Notice that such ODE's solution depends on $\boldsymbol{\Theta} = [\mathbf{x}(\tau)',\ \boldsymbol{\theta}']'$, i.e. the initial state at time τ and the parameters. The PDF (5.1) is the likelihood function of $\boldsymbol{\Theta}$.

Following a Bayesian approach, the a-priori information on the parameters can be incorporated in the estimation process by considering $\boldsymbol{\Theta}$ as a random variable with prior density $p(\boldsymbol{\Theta})$. In this case the conditional PDF of $\boldsymbol{\Theta}$ w.r.t. $\mathbf{Y}$ is given by the Bayes' rule

$$p(\boldsymbol{\Theta}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\boldsymbol{\Theta})\, p(\boldsymbol{\Theta})}{p(\mathbf{Y})} \tag{5.3}$$

An estimate of $\boldsymbol{\Theta}$ can be provided, for instance, by the *Maximum A-Posteriori* (MAP) estimator:

$$\widehat{\boldsymbol{\Theta}} \stackrel{\triangle}{=} \arg\max_{\boldsymbol{\Theta}} p(\boldsymbol{\Theta}|\mathbf{Y}) = \arg\max_{\boldsymbol{\Theta}}\ p(\mathbf{Y}|\boldsymbol{\Theta})\ \ p(\boldsymbol{\Theta}) \tag{5.4}$$

The a-priori information on the launch variables (time, position and velocity), burnout time and missile parameters can be managed in the following two ways:

- the quantities subject to equality constraints are considered known and eliminated from the vector $\boldsymbol{\Theta}$;
- the quantities subject to inequality constraints are estimated solving (5.4) and using an *uniform prior density* $p(\boldsymbol{\Theta}) = \mathcal{U}_{\overline{\boldsymbol{\Theta}}}(\boldsymbol{\Theta})$, with support equal to the region $\overline{\boldsymbol{\Theta}}$ defined by the inequality constraints, to incorporate the a-priori information in the estimation process.

With this choice, the MAP estimator becomes

$$\widehat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}\in\overline{\boldsymbol{\Theta}}} \; p(\mathbf{Y}|\boldsymbol{\Theta}) \tag{5.5}$$

where the vector $\boldsymbol{\Theta}$ is re-defined by eliminating all known parameters.

Since $\arg\max p(\cdot) = \arg\max \log p(\cdot) = \arg\min[-\log p(\cdot)]$, taking the logarithm of (5.2) the maximization (5.5) yields the constrained non linear least squares (CNLLS) problem

$$\widehat{\boldsymbol{\Theta}} \;=\; \arg\min_{\boldsymbol{\Theta}\in\overline{\boldsymbol{\Theta}}} \Gamma(\boldsymbol{\Theta}) = \arg\min_{\boldsymbol{\Theta}\in\overline{\boldsymbol{\Theta}}} \sum_{i=0}^{N-1} \|\mathbf{y}(t_i) - \mathbf{h}(\boldsymbol{\varphi}(t_i,\tau,\boldsymbol{\Theta}))\|^2_{\mathbf{R}^{-1}} \tag{5.6}$$

Solving this CNLLS problem is an hard task for a number of reasons:

- ill-conditioning;
- local minima;
- model switch.

Ill-conditioning is related to identifiability and, therefore, depends on the sensitivity of the estimated parameters. In fact, it is easy to see that

$$\frac{\partial\Gamma}{\partial\boldsymbol{\Theta}_j} = -2 \sum_{i=0}^{N-1} [\mathbf{S}^{\mathbf{y}}_j]' \; \mathbf{R}^{-1} \; (\mathbf{y}(t_i) - \mathbf{h}(\boldsymbol{\varphi}(t_i,\tau,\boldsymbol{\Theta})) \tag{5.7}$$

Hence, the derivative of the error functional Γ w.r.t. the j-th entry of the vector $\boldsymbol{\Theta}$ to be estimated, turns out to be proportional to the measurements sensitivity $\mathbf{S}^{\mathbf{y}}_j$ w.r.t. $\boldsymbol{\Theta}_j$, so that it is harder to estimate parameters with low sensitivity than parameters with high sensitivity.

Local minima are due to the non-convexity of the error functional and their presence yields the optimization algorithm sensitive to the initial value (guess parameter vector) used to start the optimization process.

Another difficulty is the switch between the boost model and the post-boost model during the burnout, which affects both the optimization process and the numerical solution of the ODE (2.16).

To overcome these difficulties possible solutions are:

1. to eliminate the ill-conditioned parameters β (low sensitivity) and t_0 (not identifiable) from $\boldsymbol{\Theta}$;
2. to use the a-priori information in order to choose a guess value for $\boldsymbol{\Theta}$;

3. to eliminate the switch parameter t_b from $\boldsymbol{\Theta}$ and devise alternative strategies to estimate it.

Hence, the new parameter vector to be estimated, obtained by eliminating the ill-conditioned parameters and the burnout time, turns out to be $\boldsymbol{\Theta} = [\mathbf{x}'(\tau), n, q]'$ and is obtained via the minimization of (5.6). Conversely the parameter β is fixed to a nominal value $\beta_0 = (\overline{\beta}+\underline{\beta})/2$. Finally, the parameters t_0 and t_b will be estimated with different strategies that will be described later (subsection 5.1 for t_0 and section 5.2 for t_b).

5.1 Guess value choice

The possibility of the optimization algorithm being trapped in a local minimum makes crucial the choice of the initial guess vector used to start the optimization process. A good choice can be obtained exploiting the radar measurements and the a-priori information in the following way.

(1) Missile parameters - For the parameters n and q subject to inequality constraints, guess values $\check{n}$ and $\check{q}$ can be chosen as the centers of the respective bounding intervals, i.e.

$$\check{n} = (\underline{n} + \overline{n})/2, \quad \check{q} = (\underline{q} + \overline{q})/2. \tag{5.8}$$

(2) Burnout time - For the burnout time t_b, an equality constraint is assumed if an IR sensor is available to detect the burnout. Conversely, if an IR sensor is not available, we will assume the knowledge of the sampling interval in which the burnout occurs. A method to locate this interval will be presented in section 6 whereas the algorithm, which exploits this information in the estimation process, will be presented in the next subsection.

(3) Missile state - The choice of the guess state $\mathbf{x}(\tau)$ depends on the available a-priori information.

Launch position known - If equality constraints on launch position $\mathbf{p}(0)$ are available, then the position components of the initial state are clearly known. To determine the velocity vector $\mathbf{v}(0)$, its magnitude v_m as well as the azimuth and elevation components v_α and, respectively, v_ϵ must be initialized. Notice that the magnitude of the velocity vector is actually zero at the launch and therefore all components of $\mathbf{v}(0)$ are zero. Nevertheless, this does not allow to exploit the available information on the azimuth and elevation angles. This problem can be overcome assuming that the magnitude of the initial velocity is suitably small but non zero, e.g. $v_m = 0.1\ m/s$. Since the flight plane of the ballistic missile is constant, the azimuth v_α can be estimated using the radar measurements in the following way:

$$v_\alpha \approx -\left[\arctan(s_1) + (1 - sign(s_1))\frac{\pi}{2} + \frac{\pi}{2}\right] + (\lambda_p - \lambda) \tag{5.9}$$

where λ_p is the longitude of the launch point and s_1 is the slope coefficient of the line $y = s_0 + s_1\, x$ which best fits the radar measurements in ENU Cartesian coordinates $(\bar{p}_{x,i}, \bar{p}_{y,i})$ for $i = 1, \ldots, N$. This coefficient is obtained from the least squares solution:

$$\begin{bmatrix} s_0 \\ s_1 \end{bmatrix} = (\mathbf{H}\ \mathbf{H}')^{-1}\ \mathbf{H}'\ \mathbf{Y} \tag{5.10}$$

with

$$\mathbf{H} = \begin{bmatrix} 1 & \bar{p}_{x,1} \\ 1 & \bar{p}_{x,2} \\ \vdots & \vdots \\ 1 & \bar{p}_{x,N} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \bar{p}_{y,1} \\ \bar{p}_{y,2} \\ \vdots \\ \bar{p}_{y,N} \end{bmatrix} \tag{5.11}$$

For the elevation angle the knowledge of lower and upper bounds, $\underline{v}_\epsilon$ and $\overline{v}_\epsilon$, is assumed and $v_\epsilon = (\underline{v}_\epsilon + \overline{v}_\epsilon)/2$ (a typical choice is $\underline{v}_\epsilon = 80°$ and $\underline{v}_\epsilon = 90°$). Hence, the guess value for the launch velocity is:

$$\begin{cases} v_x(0) = v_m\ cos(v_\epsilon)\ sin(v_\alpha) \\ v_y(0) = v_m\ cos(v_\epsilon)\ cos(v_\alpha) \\ v_z(0) = v_m\ sin(v_\epsilon) \end{cases} \tag{5.12}$$

Launch position unknown - In this case, the radar measurements must be used to determine a good guess state. Since the first measurement is available only t_0 seconds after the launch, a guess state can be found for $\mathbf{x}(\tau)$ only by fixing $\tau = t_0$. In this case the objective becomes to estimate the missile state at the first measurement time. This is a nuisance objective, the real goal being the estimate of the launch and impact points. However, this choice allows to generate a good guess state from the radar measurements in the following way:

$$\begin{cases} p_x(t_0) = r(t_0)cos(\alpha(t_0))cos(\epsilon(t_0)) \\ p_y(t_0) = r(t_0)sin(\alpha(t_0))cos(\epsilon(t_0)) \\ p_z(t_0) = r(t_0)sin(\epsilon(t_0)) \\ v_x(t_0) = \dfrac{x(t_0+T) - x(t_0)}{T} - a_x\dfrac{T}{2} \\ v_y(t_0) = \dfrac{y(t_0+T) - y(t_0)}{T} - a_y\dfrac{T}{2} \\ v_z(t_0) = \dfrac{z(t_0+T) - z(t_0)}{T} - a_z\dfrac{T}{2} \end{cases} \tag{5.13}$$

where a_x (respectively a_y and a_z) is the acceleration along the ECEF x-axis (respectively y and z). Accelerations are calculated as follows:

$$a_x = \left[\frac{x(t_0 + 2\ T) - x(t_0 + T)}{T} - \frac{x(t_0 + T) - x(t_0)}{T} \right] \frac{1}{T}$$

and similarly for a_y, a_z.

(4) Launch time - For the parameter t_0, the choice of a guess value is critical, since t_0 is not structurally identifiable. If an IR sensor detects the launch, an equality constraint on t_0 is available and, therefore, this parameter is known. In this case the lack of identifiability is overcome by eliminating t_0 from the parameter vector to be estimated. Conversely, whenever t_0 is unknown, since it is not structurally identifiable, the estimation process does not provide a good estimate neither of t_0 nor of the other parameters in the vector $\mathbf{\Theta}$. Three possible solutions for this problem are outlined hereafter.

1. Modify the *boost* model (2.10), replacing the parameters n and q defined in (2.4) by
$$n(t_0) \triangleq I_{sp}\ \frac{\dot{m}}{m(t_0)} \text{ and } q(t_0) \triangleq \frac{\dot{m}}{m(t_0)} \tag{5.14}$$
In this case, the time dependence on t_0 is eliminated by re-defining the parameters n and q w.r.t. the first measurement time. In fact the new parameters $n(t_0)$ and $q(t_0)$ are the same as n and q but at time t_0 instead of time 0.
2. Use the following time-invariant model for the *boost* phase [9]:
$$\begin{cases} \dot{p}_x = v_x \\ \dot{p}_y = v_y \\ \dot{p}_z = v_z \\ \dot{v}_x = a \dfrac{v_x}{\sqrt{v_x^2 + v_y^2 + v_z^2}} - \dfrac{\mu_G}{\left(\sqrt{p_x^2 + p_y^2 + p_z^2}\right)^3}\ p_x + 2\ \omega\ v_y + \omega^2\ p_x \\ \dot{v}_y = a \dfrac{v_y}{\sqrt{v_x^2 + v_y^2 + v_z^2}} - \dfrac{\mu_G}{\left(\sqrt{p_x^2 + p_y^2 + p_z^2}\right)^3}\ p_y - 2\ \omega\ v_x + \omega^2\ p_y \\ \dot{v}_z = a \dfrac{v_z}{\sqrt{v_x^2 + v_y^2 + v_z^2}} - \dfrac{\mu_G}{\left(\sqrt{p_x^2 + p_y^2 + p_z^2}\right)^3}\ p_z \\ \dot{a} = a\ b \\ \dot{b} = b^2 \end{cases} \tag{5.15}$$
where the new state variables are: the thrust+drag acceleration $a(t) \triangleq a_T(t) + a_D(t)$ and the burn mass rate normalized by the missile mass at time t, i.e. $b(t) \triangleq \dfrac{\dot{m}}{m(t)}$. Notice that this model is time-invariant and, therefore, does not depend on t_0.

3. Calculate the estimate using only the measurements collected during the post-boost phase. In this way, since the post-boost model is time-invariant, the dependence on t_0 is eliminated; however, in this case, only the impact point can be estimated (to estimate the launch position, whenever this is unknown, the *boost* model is necessary).

At any rate, the choice of the initial (guess) values for the parameters $n(t_0)$ and $q(t_0)$ as well as for the states is related to t_0. Therefore, if t_0 is unknown, the a-priori information on missile parameters and launch position cannot be used. For this reason, only the third approach will be considered in the sequel.

5.2 Model switch

The model switch due to the burnout can be efficiently managed by splitting the optimization task (5.6), that uses all the measurements, into two sequential sub-tasks that use only the measurements in the boost and, respectively, in the post-boost phase. This requires to know which measurements actually belong to the two phases. Let us assume that this information is available and that N_1 out of the N measurements are in the *boost* phase while the remaining N_2 are in the *post-boost* phase (see fig.3). Clearly, $N = N_1+N_2$. Further, in order to be able to estimate t_b, it must be $N_1 \geq 1$ and $N_2 \geq 1$ from which it follows that

$$\underline{t}_b \in [\underline{t}_b, \overline{t}_b], \text{ where } \underline{t}_b = t_{N_1-1},\ \overline{t}_b = t_{N_1} = \underline{t}_b + T. \tag{5.16}$$

Otherwise if $N_1 = 0$ (respectively $N_2 = 0$) it is only known that $t_b < t_0$ (respectively $t_b > t_{N-1}$). Under the a-priori knowledge of N_1 and N_2, the following **motion estimation algorithm** is suggested.

Motion Estimation Algorithm

1. Initialization:
   - Set $\check{n}$ and $\check{q}$ as in (5.8).
   - Fix $\beta_0 = \left(\underline{\beta} + \overline{\beta}\right)/2$.
   - If $\mathbf{p}(0)$ is known, set $\tau = 0$ and $\check{\mathbf{x}}(0) = [\mathbf{p}'(0), \mathbf{v}'(0)]'$ with $\mathbf{v}(0)$ given by (5.9)-(5.12);
   - If $\mathbf{p}(0)$ is unknown, set $\tau = t_0$ and $\check{\mathbf{x}}(t_0) = [\mathbf{p}'(t_0), \mathbf{v}'(t_0)]'$ with $\mathbf{p}(t_0)$ and $\mathbf{v}(t_0)$ given by (5.13);
2. If $N_1 = 0$, set $\overline{t}_b = t_0$ and go to step 7.

3. The N_1 boost measurements are used to estimate $\mathbf{x}(\tau)$, n and q by solving

$$\widehat{\mathbf{\Theta}} \;=\; \left[\begin{array}{c} \hat{\mathbf{x}}(\tau) \\ \hat{n} \\ \hat{q} \end{array}\right] = \arg \min_{\substack{\mathbf{x}(\tau) \\ \underline{n} \le n \le \overline{n} \\ \underline{q} \le q \le \overline{q}}} \sum_{i=0}^{N_1-1} \left\| \mathbf{y}(t_i) - \mathbf{h}\left(\boldsymbol{\varphi}(t_i, \tau, \mathbf{x}(\tau), n, q, \beta_0)\right)\right\|^2_{\mathbf{R}^{-1}} . \tag{5.17}$$

using $\check{\mathbf{x}}(\tau), \check{n}, \check{q}$ as guess values for $\mathbf{x}(\tau), n, q$ in the optimum search.
Then, the estimate $\hat{\mathbf{x}}(\underline{t}_b)$ is obtained by propagating the estimated state $\hat{\mathbf{x}}(\tau)$ up to time $\underline{t}_b$, i.e.

$$\hat{\mathbf{x}}(\underline{t}_b) \;=\; \boldsymbol{\varphi}(\underline{t}_b, \tau, \hat{\mathbf{x}}(\tau), \hat{n}, \hat{q}, \beta_0) \tag{5.18}$$

Then, if the launch position is unknown, $\hat{\mathbf{x}}(0)$ is obtained from $\hat{\mathbf{x}}(\tau)$ via back-integration of the boost model differential equations (2.10).

4. If $t_b = \underline{t}_b = \overline{t}_b$ is known, set $\hat{t}_b = t_b$ and

   - if $N_2 \neq 0$ go to step 7;
   - if $N_2 = 0$ go to step 8.

5. Choose an integer $L > 1$, fix $\delta T \stackrel{\triangle}{=} T/L$ and consider L "guess" burnout times $t_{bi} \stackrel{\triangle}{=} \underline{t}_b + i\ \delta T$, for $i = 1, 2, \ldots, L$. For each t_{bi}, based on the estimated state $\hat{\mathbf{x}}(\underline{t}_b)$, output predictions $\hat{\mathbf{y}}_i(t_j)$ are computed at the post-boost measurement instants t_j, $j = N_1, \ldots, N-1$. Finally, the following optimal value is selected

$$\hat{i} \;=\; \arg \min_{1 \le i \le L} \sum_{j=N_1}^{N-1} \|\mathbf{y}(t_j) - \hat{\mathbf{y}}_i(t_j)\|^2_{R^{-1}} \tag{5.19}$$

and the corresponding estimate of t_b is set equal to

$$\hat{t}_b = \underline{t_b} + \hat{i}\ \delta T \tag{5.20}$$

6. The estimated state $\hat{\mathbf{x}}(\underline{t}_b)$, given by (5.18), is propagated up to time $\overline{t}_b$, i.e.

$$\check{\mathbf{x}}(\overline{t}_b) \;=\; \boldsymbol{\varphi}\left(\overline{t}_b, \underline{t}_b, \hat{\mathbf{x}}(\underline{t}_b), \hat{n}, \hat{q}, \hat{t}_b, \beta_0\right) \tag{5.21}$$

7. The N_2 post-boost measurements are used to estimate $\mathbf{x}(\overline{t}_b)$ by solving

$$\hat{\mathbf{x}}(\overline{t}_b) \;=\; \arg \min_{\mathbf{x}(\overline{t}_b)} \sum_{i=N_1}^{N-1} \left\| \mathbf{y}(t_i) - \mathbf{h}\left(\boldsymbol{\varphi}(t_i, \overline{t}_b, \mathbf{x}(\overline{t}_b, \beta_0))\right)\right\|^2_{\mathbf{R}^{-1}} \tag{5.22}$$

using $\check{\mathbf{x}}(\overline{t}_b)$, given by (5.21), as guess value for $\mathbf{x}(\overline{t}_b)$ in the optimum search.

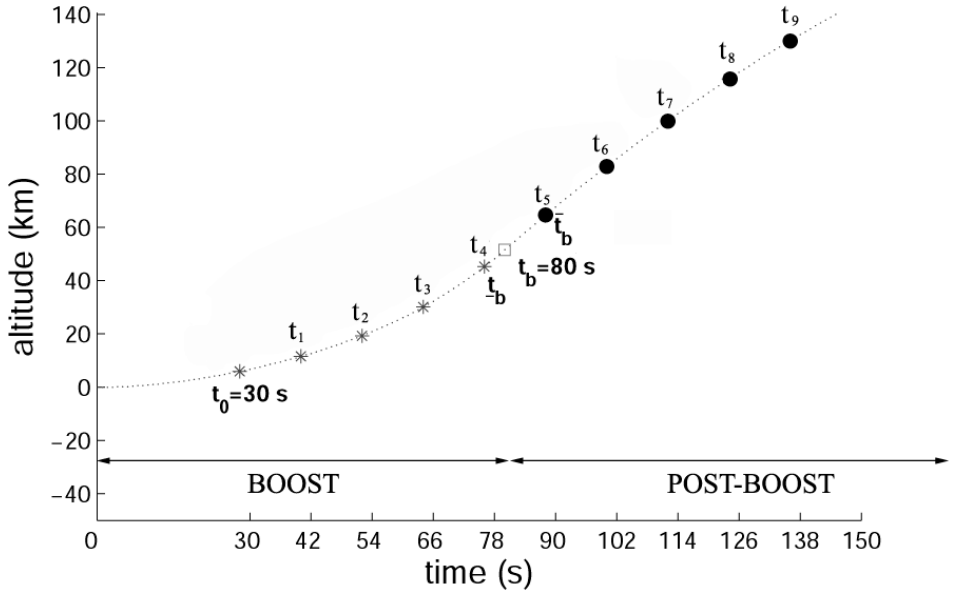

Figure 3: Missile altitude at the instants of the radar measurements

8. The estimate $\hat{\mathbf{x}}(\bar{t}_b)$ is propagated up to the impact time (that is the time in which the missile hits the earth's surface) to estimate the impact point.

Notice that if $N_2 = 0$, the burnout time t_b must be known in order to correctly propagate the state up to the final (impact) time and thus estimate the impact point. Conversely, if $N_1 = 0$ both t_0 and t_b are not needed to get the impact point, while there is no way to estimate the launch point. In this case, without loss of generality one can set $t_0 = 0$ so that the time origin is fixed at the time of the first measurement.

It must be highlighted that if the model transition is not explicitly taken into account and a full CNLLS optimization (including t_b as additional parameter) is carried out, a non negligible probability of estimation failure (parameter divergence) can be experienced.

6 Burnout detection

To apply the previous motion estimation algorithm, the interval $[\underline{t}_b, \bar{t}_b]$ in which the burnout occurs must be known. If this interval is not provided by the a-priori information, it needs to be estimated from the available data. Therefore, exploiting the radar measurements and the a-priori information on (n, q, β, t_0), the idea is to locate $[\underline{t}_b, \bar{t}_b]$ by detecting the burnout model switch. For this purpose, a *Multiple Model (MM)* recursive filter has been implemented using two *Particle Filters (PF)* tailored to the boost and,

respectively, post-boost model [6, 7]. This choice is motivated by the following considerations:

1. the MM recursive approach allows to detect the model switch comparing, scan-by-scan, the performance of the two filters (i.e. the magnitude of the prediction error of each model);

2. the particle filter allows to manage the model non-linearities as well as to include, in the filtering process, the available a-priori information on missile parameters.

In order to estimate the parameters with the MM recursive filter, the following assumptions have been made. The launch time t_0 has been considered known (equality constraints). Conversely, the parameters subject to inequality constraints, n and q, have been included in the boost filter state (assuming a random walk evolution of these parameters) and estimated as state variables. Finally, the parameter β, characterized by low sensitivity, has been fixed to a nominal value chosen according to a-priori information. Hence, the boost filter state is $\mathbf{x} = [p_x, p_y, p_z, v_x, v_y, v_z, n, q]'$, and the boost model (2.10) is augmented with $\dot{n} = 0$ and $\dot{q} = 0$. The post-boost model and the measurement equation are as in (2.11) and, respectively, (2.12). The Sampling Importance Resampling (SIR) algorithm [12]-[13] has been chosen to implement the PF, as shown in appendix A. The burnout detection algorithm based on boost and post-boost PFs is depicted in fig. 4. As it can be seen from the figure, the burnout detection algorithm is based on the comparison among the sums of particle weights of the two PFs. In fact, since each weight represents the likelihood of the corresponding particle, the model characterized on average by the highest weights is the most likely. Further, as can also be noted from fig. 4, the detection algorithm operates as follows.

1. It is assumed that at the outset the true model is the boost one ($N_1 > 0$).

2. The model comparison is carried out by means of:

$$\sum_{i=1}^{M} w_i^{pb} > \kappa \sum_{i=1}^{M} w_i^{b} \tag{6.1}$$

   where: M is the number of particles; w_i^b and w_i^{pb} ($i = 1, 2, \ldots, M$) are the weights of the boost and, respectively, post-boost PFs; $\kappa > 1$ is a scalar suitably selected so as to prevent the false alarms caused by measurements and/or process disturbances.

3. At each scan, both PFs are re-initialized with the particles of the filter that was the most likely at the previous scan.

It is worth pointing out that the adoption of the PF and of the statistical test (6.1), averaged over a large population of particles, offers a higher degree of robustness (i.e.

lower probability of false burnout detections) with respect to using the single prediction error provided by, e.g., an Extended Kalman Filter. On the other hand, as will be discussed in the next section, the computational load of the resulting burnout detection algorithm is still compatible with the prescribed timing constraints.

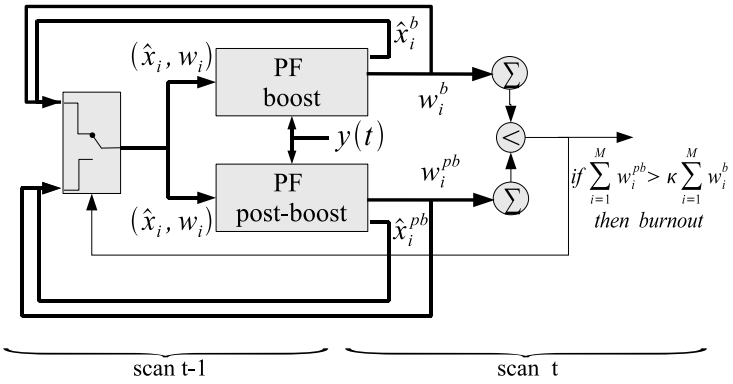

Figure 4: Burnout detection algorithm

7 Simulation results

Simulation experiments have been carried out in order to assess the estimation accuracy of the ballistic target's launch and impact points. The parameters concerning the characteristics and trajectory of the specific missile considered in the simulations, are reported in table 1 and 2. Figs. 5 and 6 show the time behavior of the target's altitude and, respectively, the radar position with respect to the launch and impact points.

Noise std. dev.			
range	100 m	azimuth	0.23°
elev.	0.20°		

Table 2: Simulation parameters

The experiments have been performed via Monte Carlo simulations with $F = 100$ independent trials obtained by varying the measurement noise realization and the bounds on the missile parameters around their true values n, q, β according to

$$\begin{aligned} &n(1-2\delta) \leq \underline{n} \leq n,\ q(1-2\delta) \leq \underline{q} \leq q,\ \beta(1-2\delta) \leq \underline{\beta} \leq \beta \\ &\overline{n} = \underline{n} + 2\delta n,\ \overline{q} = \underline{q} + 2\delta q,\ \overline{\beta} = \underline{\beta} + 2\delta\beta \end{aligned} \tag{7.1}$$

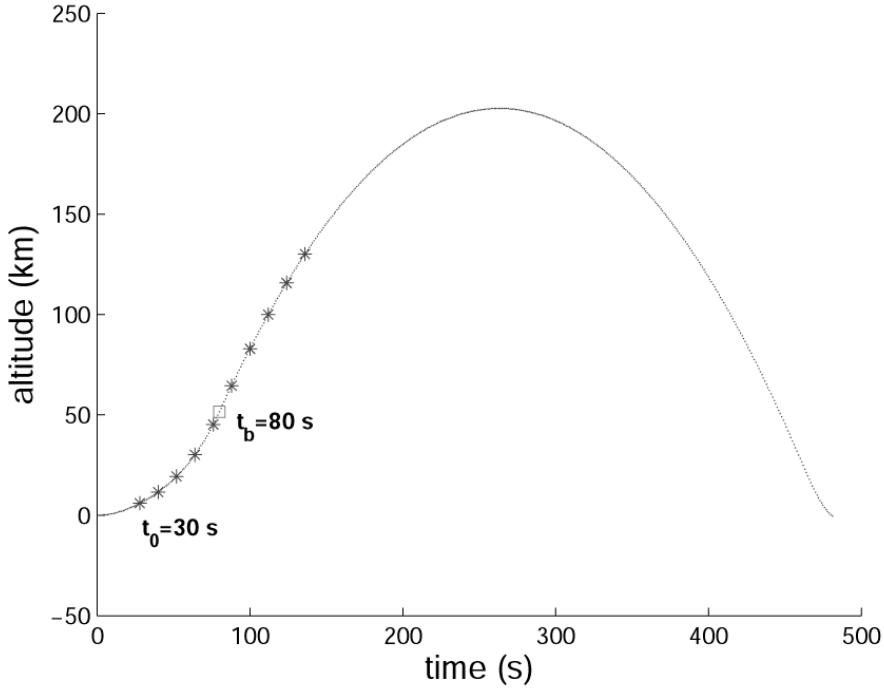

Figure 5: Target trajectory

where $\delta \in [0, 1]$ is a given relative perturbation. As performance metrics, the estimation *root mean square errors (RMSE)* on the missile parameters as well as on the launch/impact points have been considered. More specifically, denoting by $\mathbf{a}$ the generic variable to be estimated (true value) and by $\hat{\mathbf{a}}_i$ its estimate obtained in the i^{th} Monte Carlo trial, the computed metric is:

$$\mathbf{RMSE} = \sqrt{\frac{\sum_i^F (\mathbf{a} - \hat{\mathbf{a}}_i)' (\mathbf{a} - \hat{\mathbf{a}}_i)}{F}}$$

In particular, in the subsequent performance analysis, $\mathbf{a}$ will be either a scalar target parameter (n, q or t_b) or a position vector (launch or impact point). We shall first evaluate the performance of the motion estimation algorithm assuming to know either t_b or the measurements locations w.r.t. t_b, i.e. N_1 and N_2. Subsequently, the performance of the burnout detection algorithm will be examined. Finally, the joint performance of the motion estimation and burnout detection algorithms will be evaluated in the case in which the burnout time is completely unknown.

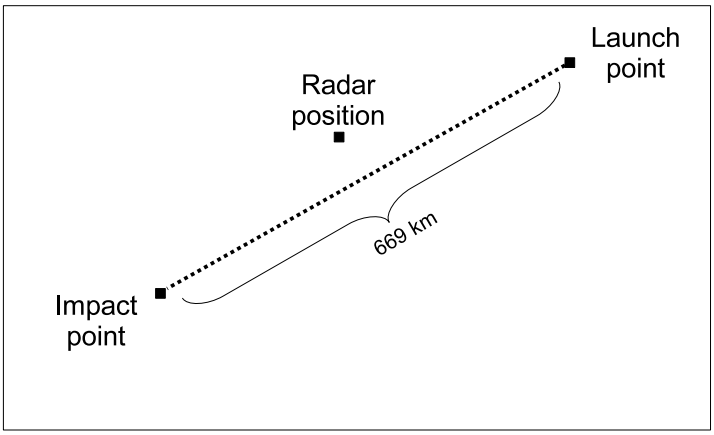

Figure 6: Projection of the missile's trajectory on the earth surface

7.1 Evaluation of the motion estimation algorithm

Five experiments, with different types of a-priori knowledge and time locations of the measurements, have been considered. These experiments are detailed in table 3. In particular, for each experiment the following items are specified: number and location of measurements, known variables, variables subject to inequality constraints, guess values used to initialize the motion estimation algorithm, complexity of the optimization problems. The number and location of measurements is specified by the notation $N = N_1 + N_2$ which indicates that N_1 out of the N measurements are in the boost phase and the remaining N_2 in the post-boost phase. Similarly, the complexity is expressed via the notation $n_p + n_{pb}$ which means that n_p variables are involved in the boost optimization (5.17) and n_{pb} in the post-boost optimization (5.22).

Notice that in the first two experiments, all the measurements are located in the boost phase, i.e. $N_1 = 5$ and $N_2 = 0$; the launch position is assumed known in experiment 1 and unknown in experiment 2. Conversely, in experiment 3 all the measurements are located in the post-boost phase i.e. $N_1 = 0$ and $N_2 > 0$. Finally, in the last two experiments $N_1 = 5$ boost measurement and $N_2 > 0$ post-boost measurements are available; the launch position is assumed known in experiment 4 and unknown in experiment 5. The time of the first measurement has been set to $t_0 = 83\ s$ for experiment 3 and $t_0 = 30\ s$ for the other experiments. The simulation results concerning all the experiments are reported in table 4 for different values of δ (the relative parameter uncertainty) and N_2 (number of post-boost measurements). Notice that the RMSE is expressed in km for the launch/impact points and as percentage of the true value for the parameters n, q and t_b. The examination of these results suggests the following considerations.

- The lack of knowledge on the launch point implies a performance degradation (see

the results of experiments 1 and 4 compared to experiments 2 and, respectively, 5).

- For given N, the best accuracy on the impact point is obtained whenever all measurements are located in the post-boost phase, i.e. for $N = N_2$ (compare the results of experiment 3 with the others). As a matter of fact, for given N_2, the exploitation of additional boost measurements can in certain cases even worsen the accuracy on the impact point (compare the results of experiment 3 with $N = 0 + 5$ and experiment 5 with $N = 5 + 5$). This counter-intuitive result is perhaps due to the estimation error on t_b which, though small, adversely affects the estimation of the impact point.

- The procedure adopted to estimate the burnout time t_b, given the bounds $\underline{t}_b$ and $\overline{t}_b$, always provides good results with errors below 1 s.

- The estimation errors on the missile parameters n and q are quite similar in the various experiments, since such estimates depend only on the boost measurements and these experiments are characterized by the same number $N_1 = 5$ of such measurements.

- In all experiments, tighter constraints ($\delta = 0.02$) clearly imply a better estimation accuracy.

7.2 Evaluation of the burnout detection algorithm

Whenever a-priori information is not available on t_b, the burnout detection algorithm must be applied to estimate the interval $[\underline{t}_b, \overline{t}_b]$ in which the burnout occurs. Using $M = 800$ particles and setting $\kappa = 1.35$ in (6.1), for both the boost and post-boost PFs, applying the detection scheme illustrated in fig. 4 and carrying out two different simulations ($t_b = 80$ s and $t_b = 88$ s), the results shown in table 5 have been obtained. From the table it can be seen that, in the first simulation ($t_b = 80$ s), the algorithm provides the estimate of the correct interval $[78, 90]$ in the 74% of the Monte Carlo runs, whereas the algorithm estimates the successive interval in the 24% of the runs. In the second simulation ($t_b = 88$ s) the correct interval $[78, 90]$ is chosen only in the 9% of the runs and the successive interval in the 89%. This different performance is clearly due to the different location of true value t_b w.r.t. the sampling interval extrema 78 and 90.

From the table it can be noticed that, summing the detection percentages, the obtained result is not 100%. In fact, in the two simulations ($t_b = 80$ s and $t_b = 88$ s) there are 2% of the trials in which the algorithm detects the burnout in intervals preceeding the correct one. This is due to divergence of PFs, caused by bad initialization of particles and/or measurement errors. Summing up, in the $t_b = 80$ case the burnout detection algorithm yields the correct estimate of the burnout interval $[\underline{t}_b, \overline{t}_b]$ in the 74% of the runs; conversely, in the $t_b = 88$ case, the algorithm yields the correct estimate only in

the 9% of the runs. To overcome this problem, a possible solution is to modify the step 5 of the motion estimation algorithm (see section 5.2) so as to manage burnout intervals of width $2T$. Therefore, if $\underline{t}_b$ and $\bar{t}_b$ are the estimates provided by the burnout detection algorithm, then the motion estimation algorithm is applied to the interval $[\underline{t}_b - T, \bar{t}_b]$. This solution allows to use the true interval, that is the interval in which the burnout really occurs, with a 98% percentage in both cases $t_b = 80$ and $t_b = 88$.

7.3 Joint motion estimation and burnout detection

In this section, the motion estimation and the burnout detection algorithms have been combined to estimate the launch and impact points whenever the burnout time is completely unknown. The interaction scheme of the two algorithms is depicted in figure 7. First, the burnout detection algorithm is used to estimate the time interval $[\underline{t}_b - T, \bar{t}_b]$ in which the burnout occurs. Then, this interval is used in the motion estimation algorithm to estimate the launch and impact points.

The experiment 4 in table 3 has been re-executed applying the combined algorithm. The obtained results are reported in table 6. Comparing these results with the results in table 4, it can be seen that the combined algorithm provides nearly the same accuracy on the impact point that has been obtained assuming the prior-knowledge of the correct sampling interval where the burnout is located. Therefore, whenever no a-priori information on the burnout time is available, the burnout detection algorithm is an efficient solution to get this information from the data.

Comments

- CNLLS optimization has been implemented using the SNOPT solver [15], an SQP (Sequential Quadratic Programming) algorithm for large-scale constrained optimization.

- The simulations have been run on a PC with AMD Athlon 64, 2.2 GHz processor and 512 Mb RAM. On this computer the batch estimation process requires on average 30 s, while the execution of the burnout detection algorithm requires 7 s per scan (less than the sampling period $T = 12$ s). Hence the proposed approach is feasible for real-time implementation.

- In the simulations, two different values of δ (the relative parameter uncertainty) have been considered: $\delta = 0.02$ (tighter constraints) and, respectively, $\delta = 0.1$ (looser constraints). For increasing δ the performance of the motion estimation algorithm clearly decreases, since the parameter region to be explored by the optimization algorithm increases with δ. To improve performance, a possible solution could be a grid search approach, as it has been discussed in section 4.

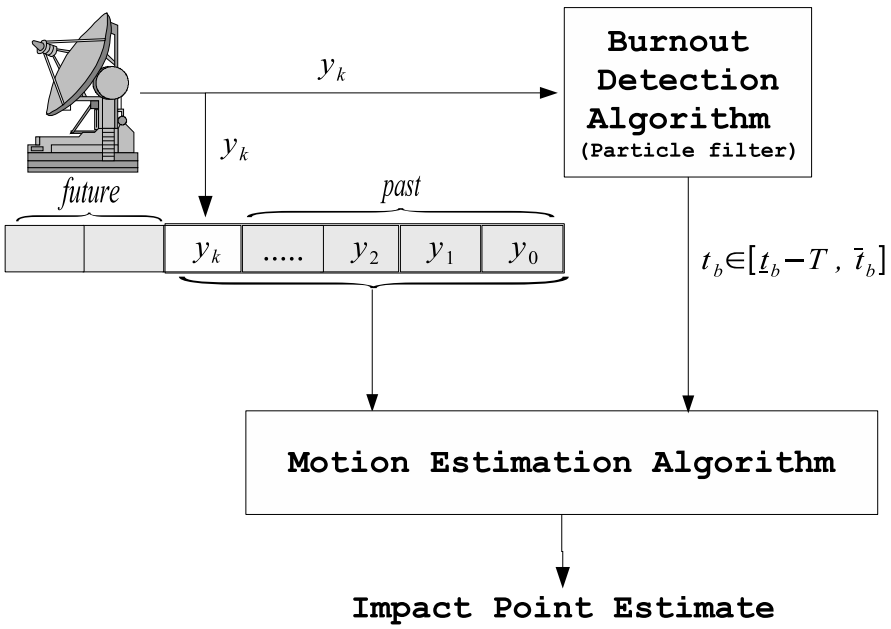

Figure 7: Joint motion estimation and burnout detection

8 Conclusions

The problem of estimating launch and impact points of a ballistic target is made critical by the scarcity of available measurements as well as by the change in the target's dynamics at burnout, i.e. when the thrusters are switched off. The switching nature of the target's dynamics and the lack of knowledge on the switching time have been overcome by means of a two-stage procedure in which a recursive filter is used to detect the model switch (burnout) and a batch optimization is used to estimate the target's launch and impact points. Notice that a batch approach allows to exploit in a better way the few available measurements for more accurate launch and impact point estimates. On the other hand, a recursive estimator, which processes the available measurements in real time, yields a prompter detection. Moreover the scarcity of data is also compensated for by a careful exploitation of a-priori information in terms of constraints and/or by a suitable choice of the guess values in the optimization. Future work will be devoted to consider the case in which the detection probability is less than 1 and the false alarm probability is greater than 0 [14]. Moreover, while the present work assumes that there is uncertainty on the target parameters but the type of the target is correctly identified, the case of unknown target type will be addressed in the future by using knowledge-based techniques. A further objective will be the derivation of a Cramer Rao lower bound for the ballistic estimation

problem.

A Burnout detection algorithm

To apply the SIR algorithm to both the boost and post-boost models, two steps must be accomplished. The boost and post-boost models must be discretized. The discretized models have been obtained by solving numerically, via the Runge-Kutta method, the ODEs (2.10) and (2.11) at the sampling instants $t = t_0 + i\ T$ for $i = 0, \ldots, N-1$. Let the discretized model be

$$\begin{cases} \mathbf{x}(t+1) &= \mathbf{f}(t, \mathbf{x}(t)) + \mathbf{w}(t) \\ \mathbf{y}(t) &= \mathbf{h}(\mathbf{x}(t)) + \boldsymbol{\eta}(t) \end{cases} \tag{A.1}$$

with $\mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$, $\boldsymbol{\eta}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ and

$$\mathbf{R} = diag\left\{\sigma_r^2, \sigma_\alpha^2, \sigma_\epsilon^2\right\}, \quad \mathbf{Q} = diag\left\{\mathbf{Q}_c;\ \mathbf{Q}_d\right\} = diag\left\{\sigma_{p_x}^2, \sigma_{p_y}^2, \sigma_{p_z}^2, \sigma_{v_x}^2, \sigma_{v_y}^2, \sigma_{v_z}^2;\ \sigma_n^2, \sigma_q^2\right\}$$

where σ_j is the standard deviation of the j-th entry of the state or measurement vector. The process noise $\mathbf{w}(t)$ has been introduced to consider the discretization errors and the uncertainty on the initial condition. Further, it is assumed that

- the process noise vectors $\mathbf{w}(t)$ form an i.i.d. sequence;
- the measurement noise vectors $\boldsymbol{\eta}(t)$ form an i.i.d. sequence;
- $\{\mathbf{w}(t)\}$, $\{\boldsymbol{\eta}(t)\}$ and $\mathbf{x}(0)$ are all mutually independent.

Notice that the process covariance matrix $\mathbf{Q}$ has been partitioned in two sub-matrices: $\mathbf{Q}_c$ is the covariance of the kinematic components $\mathbf{c} = [p_x, p_y, p_z, v_x, v_y, v_z]'$ and $\mathbf{Q}_d$ is the covariance of the parametric components $\mathbf{d} = [n,\ q]'$. The state vector has been partitioned accordingly as $\mathbf{x}(t) = [\mathbf{c}'(t), \mathbf{d}'(t)]'$. This partitioning is necessary for two reasons: first the $\mathbf{d}$ components are only present in the boost model, secondly the a-priori information is available only for $\mathbf{d}$. Therefore, the SIR algorithm is [12]-[13]:

(0) Initialization - For $i = 1, \ldots, M$:

- $\overline{w}_i(0) = \dfrac{1}{M}$
- **Sampling** - $\mathbf{x}_i(0) = [\mathbf{c}_i'(0), \mathbf{d}_i'(0)]'$: $\mathbf{c}_i(0) \sim \pi_c(\mathbf{c}(0))$ and $\mathbf{d}_i(0) \sim \pi_d(\mathbf{d}(0))$
- **Constraint checking** - If $\mathbf{d}_i(0) \notin \overline{\mathbf{D}}$ go back to **sampling**.

(1) For $t > 1$ and for $i = 1, \ldots, M$:

- **Sampling** - $\mathbf{x}_i(t) = [\mathbf{c}_i'(t), \mathbf{d}_i'(t)]'$: $\mathbf{c}_i(t) \sim \pi_c(\mathbf{c}(t)|\mathbf{c}_i(t-1), \mathbf{y}(t))$ and $\mathbf{d}_i(t) \sim \pi_d(\mathbf{d}(t))$;

- **Constraint checking** - If $\mathbf{d}_i(0) \notin \overline{\mathbf{D}}$ go back to **sampling**;
- **Weight updating** - $w_i(t) = p(\mathbf{y}(t)|\mathbf{x}_i(t-1))$

(2) Weight normalization - $\overline{w}_i(t) = \dfrac{w_i(t)}{\sum_{j=1}^{M} w_j(t)}$

(3) Estimate calculation - $\widehat{\mathbf{x}}(t) = \sum_{i=1}^{M} \overline{w}_i(t)\ \mathbf{x}_i(t)$

(4) Resampling - Resample $\mathbf{x}_i(t)$ for $i = 1, \ldots, M$, set $t = t + 1$ and go back to (1).

where $\mathbf{x}_i(t)$ and $w_i(t)$ are the state value and, respectively, the weight of the i^{th} particle; $\widehat{\mathbf{x}}(t)$ is the estimated state; $\overline{\mathbf{D}}$ is the constrained region defined by a-priori information on the state variables n and q (i.e. $\underline{n} \leq n \leq \overline{n}$, $\underline{q} \leq q \leq \overline{q}$) and

$$\begin{aligned} \pi_c(\mathbf{c}(0)) &= \mathcal{N}(\mathbf{c}(0), \mathbf{Q}_c) \\ \pi_c(\mathbf{c}(t)|\mathbf{c}(t-1), \mathbf{y}(t)) &= \mathcal{N}(\mathbf{m}(t), \mathbf{\Sigma}(t)) \\ \pi_d(\mathbf{d}(t)) &= \mathcal{U}_{\overline{\mathbf{D}}} \\ p(\mathbf{y}(t)|\mathbf{x}(t-1)) &= \mathcal{N}(\mathbf{y}(t); \mathbf{h}(\mathbf{f}(t-1, \mathbf{x}(t-1))), \mathbf{Q}(t) + \mathbf{H}'(t)\ \mathbf{R}(t)\ \mathbf{H}(t)) \end{aligned} \tag{A.2}$$

with

$$\begin{aligned} \mathbf{H}(t) &\triangleq [\mathbf{H}_c(t),\ \mathbf{H}_d(t)] = \left[\frac{\partial \mathbf{h}(\mathbf{x}(t))}{\partial \mathbf{c}(t)},\ \frac{\partial \mathbf{h}(\mathbf{x}(t))}{\partial \mathbf{d}(t)}\right] \\ \mathbf{\Sigma}^{-1}(t) &\triangleq \mathbf{Q}_c^{-1}(t) + \mathbf{H}_c'(t)\ \mathbf{R}^{-1}(t)\ \mathbf{H}_c(t) \\ \mathbf{m}(t) &\triangleq \mathbf{\Sigma}(t)\ \mathbf{Q}_c^{-1}(t)\ \mathbf{f}_c(t, \mathbf{x}(t)) + \mathbf{H}_c'(t)\ \mathbf{R}^{-1}(t)\ \mathbf{H}_c(t) \\ \mathbf{f}(t, \mathbf{x}(t)) &\triangleq [\mathbf{f}_c'(t, \mathbf{x}(t)), \mathbf{f}_d'(t, \mathbf{x}(t))]' \end{aligned} \tag{A.3}$$

The initial condition $\mathbf{x}(0) = [\mathbf{c}'(0), \mathbf{d}'(0)]'$ can been obtained using the measurements for the position and velocity components, as in (5.13), and the centers of the bounding intervals for n and q. Notice that the $\mathbf{d}$ components are not present in the post-boost PF.

References

[1] A.H. Jazwinski (1971).*Stochastic processes and filtering theory*, Academic Press, New York.

[2] R. Mehra (1971). A comparison of several nonlinear filters for re-entry vehicle tracking. *IEEE Trans. on Automatic Control*, **AC-16**, n. 4, pp. 307-319.

[3] C.B. Chang, R.H. Whiting, M. Athans (1977). On the state and parameter estimation for maneuvering reentry vehicles. *IEEE Trans. on Automatic Control*, **AC-22**, n. 1, pp. 99-105.

[4] P. Costa (1994). Adaptive model architecture and extended Kalman-Bucy filters. *IEEE Trans. on Aerospace and Electronic Systems*, **AES-30**, n. 2, pp. 525-533.

[5] R. Hutchins, A. San Jose (1998). IMM tracking of a theater ballistic missile during boost phase. In *Proc. of SPIE Conf. on Signal and Data Processing of Small Targets (SPIE vol. 3373)*, Orlando, FL, pp. 528-537.

[6] A. Farina, B. Ristic, D. Benvenuti (2002). Tracking a ballistic target: comparison of several nonlinear filters. *IEEE Trans. on Aerospace and Electronic Systems*, **AES-38**, n. 3, pp. 854-867.

[7] A. Farina, B. Ristic, D. Benvenuti (2002). Estimation accuracy of a landing point of a ballistic target. In *Proc. Fusion 2002 Conf.*, Annapolis, USA, 2002.

[8] R. Bate, et al. (1971). *Fundamentals of Astrodynamics*, Dover, New York.

[9] X. R. Li, V. P. Jilkov (2001). A survey of maneuvering target tracking: ballistic target models. In *Proc. of SPIE Conf. on Signal and Data Processing of Small Targets*, San Diego, CA, USA, 2001.

[10] S.M. Kay (1993). *Fundamentals of statistical signal processing: estimation theory*, Prentice Hall, London

[11] Y. Bar-Shalom, X. R. Li, T. Kirubarajan (2001). *Estimation with applications to tracking and navigation*, John Wiley & Sons, New York.

[12] B. Ristic, S. Arulampalam, N. Gordon (2004). *Beyond the Kalman Filter: particle filters for tracking applications*, Artech House, Boston.

[13] A. Doucet, J. F. G. De Freitas, N. J. Gordon (2001). *Sequential Monte Carlo methods in practice*, Springer-Verlag, New York.

[14] M. Hernandez, A. Farina, B. Ristic (2006). A PCRLB for tracking in cluttered environments: a measurement sequence conditioning approach. To appear in : *IEEE Trans. on Aerospace and Electronic Systems*.

[15] P.E. Gill, W. Murray, M. Saunders (2005). SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Review*, pp. 99-131.

Experiment 1	**Known launch point, boost measurements**
	measurements $N = 5 + 0$;
	known vars. launch position $\mathbf{p}(0)$, t_0, t_b;
	constrained vars. $n \in [\underline{n},\ \overline{n}]$, $q \in [\underline{q},\ \overline{q}]$
	guess values $v_x(0), v_y(0), v_z(0)$ as in (5.9)-(5.12); n_0 and q_0 as in (5.8)
	complexity $5 + 0$
	estimation objective impact point, n, q
Experiment 2	**Unknown launch point, boost measurements**
	measurements $N = 5 + 0$;
	known vars. t_0, t_b;
	constrained vars. $n \in [\underline{n},\ \overline{n}]$, $q \in [\underline{q},\ \overline{q}]$
	guess values $\mathbf{p}(t_0), \mathbf{v}(t_0)$ as in (5.13); n_0 and q_0 as in (5.8)
	complexity $8 + 0$
	estimation objective launch and impact point, n, q
Experiment 3	**Post-boost measurements**
	measurements $N = 0 + 5$;
	known vars. none
	constrained vars. none
	guess values $p_x(t_0), p_y(t_0), p_z(t_0), v_x(t_0), v_y(t_0), v_z(t_0)$ as in (5.13);
	complexity $0 + 6$
	estimation objective impact point
Experiment 4	**Known launch point, boost and post-boost measurements**
	measurements $N = 5 + N_2$;
	known vars. launch position $\mathbf{p}(0)$, t_0;
	constrained vars. $n \in [\underline{n},\ \overline{n}]$, $q \in [\underline{q},\ \overline{q}]$, $t_b \in [\underline{t}_b, \overline{t}_b]$
	guess values $v_x(0), v_y(0), v_z(0)$ as in (5.9)-(5.12); n_0 and q_0 as in (5.8)
	complexity $6 + 6$
	estimation objective impact point, t_b, n, q
Experiment 5	**Unknown launch point, boost and post-boost measurements**
	measurements $N = 5 + N_2$;
	known vars. t_0;
	constrained vars. $n \in [\underline{n},\ \overline{n}]$, $q \in [\underline{q},\ \overline{q}]$, $t_b \in [\underline{t}_b, \overline{t}_b]$
	guess values $\mathbf{p}(t_0), \mathbf{v}(t_0)$ as in (5.13), n_0 and q_0 as in (5.8);
	complexity $9 + 6$
	estimation objective launch and impact point, t_b, n, q

Table 3: Experiments for the motion estimation

		$\delta = 0.02$	$\delta = 0.1$
Experiment 1, $N = 5 + 0$	**impact point**	18.63 km	23.79 km
	parameter n	0.55 %	1.19 %
	parameter q	1.02 %	1.56 %
Experiment 2, $N = 5 + 0$	**impact point**	27.42 km	36.11 km
	parameter n	1.52 %	1.71 %
	parameter q	1.32 %	1.37 %
	launch point	1.7 km	2.2 km
Experiment 3, $N = 0 + 5$	**impact point**	7.69 km	8.43 km
Experiment 3, $N = 0 + 10$	**impact point**	2.01 km	2.13 km
Experiment 4, $N = 5 + 5$	**impact point**	8.12 km	15.64 km
	parameter n	0.52 %	1.08 %
	parameter q	1.12 %	1.59 %
	parameter t_b	0.52 s	0.66 s
Experiment 4, $N = 5 + 10$	**impact point**	1.8 km	2.05 km
	parameter n	0.53 %	1.09 %
	parameter q	1.03 %	1.61 %
	parameter t_b	0.58 s	0.71 s
Experiment 5, $N = 5 + 5$	**impact point**	8.69 km	16.69 km
	parameter n	1.47 %	1.77 %
	parameter q	1.36 %	1.66 %
	parameter t_b	0.63 s	0.84 s
	launch point	1.61 km	1.96

Table 4: Simulation results for the motion estimation algorithm

	$t_b = 80$ s	$t_b = 88$ s
sampling interval	detection percentage	detection percentage
[78, 90]	74%	9%
[90, 102]	24%	89%

Table 5: Simulation results for the burnout detection algorithm.

		$\delta = 0.02$	$\delta = 0.1$
Experiment 4, $N = 5 + 5$	**impact point**	9.32 km	16.85 km
	parameter n	0.67 %	1.23 %
	parameter q	1.16 %	1.74 %
	parameter t_b	0.56 s	0.65 s
Experiment 4, $N = 5 + 10$	**impact point**	1.91 km	2.13 km
	parameter n	0.65 %	1.21 %
	parameter q	1.23 %	1.84 %
	parameter t_b	0.59 s	0.69 s

Table 6: Simulation results for the combined algorithm