

TODO:

- read VKB
- read Alex's Paper

## “Read Those Three Papers”

First, it appears that we care about actually simulating targets with motion modeled per Singer model.

As of 2024-09-30, Python code re-implements: <https://www.mathworks.com/help/fusion/ref/singer.singer.html>

Target  $k$  has state  $s_k = [x\dot{x}\ddot{x}; y\dot{y}\ddot{y}; z\dot{z}\ddot{z}]$ .

$\delta_t$  is a positive time scalar in seconds.

## Where does revisit rate come from?

The revisit rate  $t_r$  is calculated as follows:

$$t_r = 0.4 \left( \frac{R_t \sigma_\theta \sqrt{\Theta}}{\Sigma} \right)^{0.4} \frac{U^{2.4}}{1 + \frac{U^2}{2}}$$

**R\_t**

Range to the target.  $R_t = \sqrt{x^2 + y^2 + z^2}$ .

**sigma\_theta**

**Sigma**

In the slides, Sigma is Manoeuvre std in  $\frac{m}{s^2}$ .

**Theta**

In the slides, Theta is Manoeuvre time in seconds.

**U**

**We set  $U = 0.3$ , why?**

**Where does  $\Sigma$  come from?**

## Thoughts

### van Keuk and Blackman

Van Keuk and Blackman (VKB) [1] state rules for selecting revisit interval time and signal-to-noise (SNR) ratio to minimize the loading of each active tracking task.

In particular, with respect to radar/computer load. My gut instinct is “What do they mean by computational load? Have computers improved since 1986?”

### Notable Assumptions

- Assume that the individual tracks are not distinguished by additional information like estimated threat or any other tasking.

- Swerling I model, does not describe the time correlation of the signal amplitudes  $A_i$  between two consecutive dwells separated by the time  $\tau$ .
- In a phased array,  $\tau$  can be much smaller than a rotating radar.
- $r$  is of no importance for our consideration here. (do everything in u,v polar coordinates uncertainty)

## Numbers

- $\Delta T$  time between consecutive target updates.
- TOT time on target, dwell.
- $B$  is half beamwidth.
- $N$  is time step,  $t_N, t_{N+1}$ , so on.
- $Z(N+1 | N)$  is the predicted target state at time  $t_{N+1}$ .
- $V(N+1 | N)$  is the covariance based on all associated measurements up to  $t_N$ , gives us ellipsoid  $(u, v)$  with major axis  $G$  “Lack of information”.
- $V_0$  maximum allowed inaccuracy of the track.

In their model, when  $G$  hits some  $V_0$ , a track update is commanded. They show a familiar “fast revisits followed by slower” sawtooth.

## VKB Motion Model

Three-dimensional target motion in Cartesian coordinates driven by three mutually independent Markov target Gaussian acceleration processes  $q_i$ . Speed and position follow by integration with respect of time.

They then say that this is completely described by Formula (9) [1]  $E[q_{i(t)}q_{j(t+\tau)}] = \delta_{ij} \cdot \Sigma^2 e^{-\frac{\tau}{\Theta}}$ .

- $\Sigma$  is acceleration standard deviation.
- $\Theta$  is correlation time.

So this autocorrelation function is maximum (equal to  $\Sigma^2$ ) when  $\tau = 0$  and decreases exponentially with increasing  $\tau$ .

VKB then cite an earlier paper by Blackman and another paper by Bar-Shalom and Fortmann.

They end up finding that the “Optimal value is about 0.3 of the half beamwidth”.

## VKB Lost Target

They fire at the centre of a pdf based on  $G$ , then recompute using Bayes’ rule, and then keep firing at the maximum of the pdf. They calculate this once and then just replay the same pattern.

## Singer

The MATLAB code I was referencing cited a 1970 Singer [2] paper.

## My Own Motion Model

After our talk 2024-10-01, I changed my motion model to use the following transition matrix.

In this, the  $t$  is the time until next sensor is available.

$\rho = e^{-\frac{t}{\Theta}}$  based on your notes during this meeting, I’ll find an actual reference in presumably the VKB [1] or Alex [3] paper.

$$\begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \\ y \\ \dot{y} \\ \ddot{y} \\ z \\ \dot{z} \\ \ddot{z} \end{pmatrix} \times \begin{pmatrix} 1 & t & \frac{t^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & t & \frac{t^2}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \rho & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & t & \frac{t^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \rho \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \sqrt{1-\rho^2}\mathcal{N}(0, \Sigma^2) \\ 0 \\ 0 \\ \sqrt{1-\rho^2}\mathcal{N}(0, \Sigma^2) \\ 0 \\ 0 \\ \sqrt{1-\rho^2}\mathcal{N}(0, \Sigma^2) \end{pmatrix}$$

## Singer 2: Back to the Singer

Dimensions are independent,  $9 \times 9$  is just a triple  $3 \times 3$ , so we'll just do one dimension at a time.

- $p = [x, \dot{x}, \ddot{x}]^T$  is some part of the state vector.
- $\theta$  is a target specific time constant sampled  $\sim \mathcal{U}(\Theta)$
- $\alpha = \frac{1}{\theta}$  is the reciprocal of the time constant.
- $\sigma$  is a target specific maneuver constant sampled  $\sim \mathcal{U}(\Sigma)$

You had jotted down the following in our meeting:

$$p(k) = \begin{pmatrix} 1 & t & \frac{t^2}{2} \\ 0 & 1 & t \\ 0 & 0 & e^{-\frac{T}{\theta}} \end{pmatrix} p(k-1) + \begin{pmatrix} 0 \\ 0 \\ \sqrt{1 - \left(e^{-\frac{T}{\theta}}\right)^2} \mathcal{N}(0, \sigma^2) \end{pmatrix}$$

I still couldn't quite figure out, in particular, why the process noise looked like that.

The Singer model in MATLAB is described as follows:

$$p(k) = \begin{pmatrix} 1 & T & \frac{\alpha T - 1 - e^{-\alpha T}}{\alpha^2} \\ 0 & 1 & \frac{1 - e^{-\alpha T}}{\alpha} \\ 0 & 0 & e^{-\alpha T} \end{pmatrix} p(k-1) + w(k)$$

$w(k)$  is singer process noise, and then after pulling some threads I couldn't figure out how they come up with this. They do reference what I assume is the actual Singer [2] paper.

Singer views the world as constant velocity, with turns, evasive maneuvers, and atmospheric turbulence as perturbations on an otherwise constant velocity trajectory. Acceleration is the “maneuver variable”, it is parameterized by variance (magnitude)  $\sigma_m^2$  and the time constant (duration)  $\theta$ .

- $A_{\max}$  is the maximum acceleration (symmetric, so  $a \in [-A_{\max}, A_{\max}]$ ).
- $P_{\max}$  is the probability of selecting  $A_{\max}$
- $P_0$  is the probability of selecting 0 acceleration.
- $\sigma_m^2 = \frac{A_{\max}^2}{3}(1 + 4P_{\max} - P_0)$ , which I do not follow at all.
- I imagined a pmf triangle or maybe pentagon, but a diagram shows a uniform rectangle everywhere with what appear to be spikes, biggest at 0, smaller ones at  $\pm P_{\max}$ ?
- I don't think I care about this spiky distribution? If it even is? I think maybe it's just uniform?

Singer initializes filters with two measurements exactly how you think  $\hat{x} = (y(1) \frac{y(1)-y(0)}{t} 0)^T$  and all noise in  $a$ , not even a measurement noise in the other two?.

Singer throws out some numbers.  $\alpha \approx \frac{1}{60}$  for a lazy turn,  $\alpha \approx \frac{1}{20}$  for an evasive turn, and  $\alpha \approx 1$  for atmospheric turbulence. I don't think I achieve an evasive turn with this though...?

In this  $w(k) = \text{white noise driving function with variance } \begin{pmatrix} 0 \\ 0 \\ 2\alpha\sigma^2 \end{pmatrix}$ . I don't understand this. Why is there a 2?

My  $T$  is always very small. Singer notes that when  $T$  is small the model reduces to Newtonian constant acceleration.

Singer is in the  $10^{-2}$  to  $10^2$  of seconds range, and my  $T$  is in the  $10^{-3}$  to  $10^{-4}$  range, and will only get smaller when adding more sensors due to my "fast forward the universe just until the next sensor is idle". I'm only ever moving everything forward by the shortest remaining dwell time. I am pretty sure I break physics, but frankly don't really care here? Due to my microscopic  $T$ , I don't think that any "Sample and multiply by magnitude" will ever do what I want. I'm going to try to dump back in my "annoying" targets who. I want to prioritize these programmable targets. Due to how difficult I found it to be to find an interesting scenario, and Peter's feedback that he'd like to see some replayable scenes to evaluate algorithms among, I'm basically going to just implement much more *Videogamey* targets that are programmable. I'm just going to put timers in them and they'll just go from 0 to some A after a set time, then maintain that for some time. That instantaneous acceleration may technically break somebody's back, but I'm integrating over such a short  $\tau$  that it'll be lost in the measurement noise even if you were staring right at it. Plus this should let me do even more fun targets that do things like count how often they think they've been measured or something. Model drones. Even Singer says this is only appropriate for airships, submarines, and lumbering 1970s airplanes. We care about missiles, tumbling chaff, and quadcopters. I'm giving up on strictly following Singer for now. If doing some stupider noisy acceleration model is sufficient for VKB [1] then it's good enough for me.

## Bibliography

- [1] G. van Keuk and S. S. Blackman, "On phased-array radar tracking and parameter control," *IEEE Transactions on aerospace and electronic systems*, vol. 29, no. 1, pp. 186–194, 1993.
- [2] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Transactions on Aerospace and electronic systems*, no. 4, pp. 473–483, 1970.
- [3] A. Charlish, K. Woodbridge, and H. Griffiths, "Multi-target tracking control using continuous double auction parameter selection," in *2012 15th International Conference on Information Fusion*, 2012, pp. 1269–1276.