Optimal Sensor Trajectories in Bearings-Only Tracking

Marcel L. Hernandez
The Advanced Processing Centre
QinetiQ Ltd., St Andrew's Road
Malvern WR14 3PS, UK
Email: mlhernandez@QinetiQ.com

Abstract – *We consider the problem of determining sensor trajectories in the bearings-only tracking of an uncertain target. This work differs from previous research in that we allow the target dynamics to be both uncertain and random; and also consider multi-sensor scenarios. The basis of our technique is to control a measure of estimation error based on the Posterior Cramér-Rao lower bound (PCRLB), and we present both single-step and multi-step planning approaches. We also introduce an efficient search technique that allows us to quickly perform the necessary optimisation(s). Sensor trajectories are shown to be almost identical to those obtained by performing an enumerative search, but the computational load is reduced by several orders of magnitude. Simulation results are presented for 1, 2 and 3 sensor scenarios, and compare the single-step and two-step approaches. The performance (in terms of both the PCRLB and particle filter estimation errors) improves both with two-step planning and as sensor numbers increase, as one would expect.*

Keywords: Posterior Cramér-Rao lower bound, bearings-only tracking, Fisher information matrix, non-linear filtering, resource management, target motion analysis.

1 Introduction

The PCRLB is defined to be the inverse of the Fisher information matrix (FIM) [1] and, in the context of tracking, provides a bound on the optimal achievable accuracy of target state estimation. Traditionally the bound has been used to assess the performance of filtering algorithms (e.g. [2]). However, utilising the sensor resource in order to control the PCRLB has recently been proposed for radar system design [3] and used in automating the spatial deployment and operation of a limited sensor resource [4].

Further, a recent development [5] now allows computationally efficient calculation of the PCRLB for general non-linear estimation problems. The overall result is that the PCRLBs can be calculated quickly, making them suitable for application in a range of online, real-time sensor management scenarios. The sensor management technique [4] has proved to be particularly effective in deploying passive sonobuoys in submarine tracking. Moreover, it has recently been shown to be a viable technique for multistatic radar resource management [6], and for bandwidth limited sensor selection in large networks of unattended ground sensors [7].

Another popular area of sensor management research has been optimal observer manoeuvre in bearings-only tracking (see [8] and references therein). Here the problem is to determine the sensor path that will allow best estimation of the uncertain target state. In [8] the authors consider the problem of tracking a target that can perform manoeuvres but travels with a constant velocity on each leg. Conditions prescribing the optimal observer trajectory based on the CRLB[1] are then determined. Other papers have also looked at the problem of optimal observer manoeuvre using the CRLB. In [9] a constant velocity target was considered and in [10] a fixed target was being localised but with additional constraints imposed on the observer path. In [11] continuous time analysis was performed, again for a constant velocity target.

In the current paper we are again concerned with optimising the motion of bearings-only sensor platforms (e.g. aircraft) in order to best track an uncertain target. We build on previous work [12] to consider multi-step planning (or "fine-step" planning: see [7]), and we again perform the necessary sensor management by controlling a measure of estimation error based on the PCRLB. As in [12] we also divide the observation period into a number of sub-intervals of equal duration, and only allow sensor manoeuvres at discrete time points. However, we present a computationally efficient means of determining the optimal manoeuvres that results in a considerable improvement in running time when compared to an enumerative search technique. We also show that multi-step planning provides better solutions (than single step planning) with both the PCRLB and filter Root Mean Squared Errors (RMSEs) lower.

[1] This is the inverse of the FIM when estimating an uncertain but non-random parameter vector.

There are two primary differences between the current paper and the previous work reported earlier. Firstly, we allow the target dynamics to be both uncertain and random, necessitating utilisation of the PCRLB, which has to account for the uncertain target dynamics and, as a result, is more complicated than the CRLB. The second difference is that we allow multiple observers, although we note that in many of these previous works the extension from the single observer case is straightforward.

The paper is structured as follows. In section 2 we determine the PCRLB for the focal problem. In section 3 we present the sensor management algorithm and define the single-step and multi-step optimisation problems. In section 4 we present the efficient optimisation algorithm. Simulation results are given in section 5 with conclusions following in section 6.

2 Posterior Cramér-Rao Bounds

2.1 Definition

Let $\hat{X}_k$ be an unbiased estimator of a parameter vector X_k, based on some measurement sequence. Then the PCRLB for the error covariance is defined to be the inverse of the FIM, J_k [1], i.e.

$$C_k \triangleq \mathbb{E}\left[\left[\hat{X}_k - X_k\right]\left[\hat{X}_k - X\right]^T\right] \geq J_k^{-1} \quad (1)$$

The inequality in equation (1) means that the difference $C_k - J_k^{-1}$ is a positive semi-definite matrix.

2.2 PCRLBs For Target State Estimation

We consider the following linear dynamical system

$$X_{k+1} = A_k X_k + w_k \quad (2)$$

where X_k is the target state at time k and $\{w_k\}$ is a white noise sequence, with $w_k \sim N(0, Q_k)$. Measurements are available at discrete time epochs. We denote the measurement sequence by Z_k, $k \geq 1$.

Until recently, calculating PCRLBs has proved notoriously difficult. However Tichavsky et al. [5] provide an elegant Riccati like recursion giving the sequence of Fisher information matrices, J_k, $k > 0$, for the general non-linear estimation problem. Moreover, based on the target motion model (2) it can easily be shown (e.g. [2]) that the general recursion of [5] reduces to

$$J_{k+1} = \left(Q_k + A_k J_k^{-1} A_k^T\right)^{-1} + J_Z(k+1) \quad (3)$$

where $J_Z(k+1)$ is the measurement contribution (see the next section). The initial FIM is given by

$$J_0 = \mathbb{E}\left[-\Delta_{X_0}^{X_0} \log p(X_0)\right] \quad (4)$$

where $\Delta_{X_0}^{X_0}$ is a second-order partial derivative operator (see e.g. [5]). If $p(X_0) \sim N(\bar{X}_0, C_0)$ then $J_0 = C_0^{-1}$.

2.3 The Measurement Contribution

Measurements epochs $k = 1, 2, \ldots$ occur at times $t = Tk$, where T is the time between successive measurements. There is a minimum detection range r at each sensor, inside which the target cannot be detected (see [13] for an explanation as to why this is necessary). If the target is outside of this range then a target generated measurement is received. There is no clutter.

Measurements are of bearings only. The measurement at sensor j at time k can then be written as

$$Z_k^j = \tan^{-1}\left(\frac{y_k - y_k^j(S)}{x_k - x_k^j(S)}\right) + \nu_k \quad (5)$$

where

$$X_k^j(S) = (x_k^j(S)\ \dot{x}_k^j(S)\ y_k^j(S)\ \dot{y}_k^j(S))^T \quad (6)$$

$$X_k = (x_k\ \dot{x}_k\ y_k\ \dot{y}_k)^T \quad (7)$$

give the states of sensor j and the target respectively (notation is standard). The error, $\nu_k \sim N(0, \sigma^2)$. It is then straightforward (e.g. [2]) to show that the measurement contribution to the PCRLB is given by

$$J_Z(k) = \frac{1}{\sigma^2}\sum_{j=1}^{N} \mathbb{E}\left[I_r(X_k^j(S); X_k) H_j(X_k)^T H_j(X_k)\right] \quad (8)$$

where

$$H_j(X_k) = \begin{pmatrix} \frac{(y_k - y_k^j(S))}{(x_k - x_k^j(S))^2 + (y_k - y_k^j(S))^2} \\ 0 \\ \frac{-(x_k - x_k^j(S))}{(x_k - x_k^j(S))^2 + (y_k - y_k^j(S))^2} \\ 0 \end{pmatrix}^T \quad (9)$$

$I_r()$ is an indicator function that is equal to unity if the sensor and target are separated by a distance greater than r, and is zero otherwise. N is the number of sensors. The expectation $\mathbb{E}[]$ is with respect to the target state.

3 Sensor Management

3.1 Background

The general framework detailed in [4] is based on utilising the sensor resource in order to control the PCRLB. The PCRLB is dependent on a number of factors: the target state probability density function (PDF), the sensor location(s) and the measurement accuracy. Hence, in our focal application, we can analyse a range of (candidate) sensor trajectories, and in each case determine the sequence of Fisher information matrices, J_k, $k \geq 1$. We are then in a position to control the sensor trajectories in order to minimise a measure based on the resulting PCRLBs (given by J_k^{-1}, $k \geq 1$).

3.2 Performance Measure

In this paper, the path of each sensor is determined entirely from the prior target distribution (at time 0) and (assumed known) target dynamics[2]. The initial FIM is then given by $J_0 = C_0^{-1}$, with J_k, $k \geq 1$ given by the recursion (3).

Each measurement contribution, $J_Z(k)$ is calculated using Monte Carlo integration [14] based on the prior target distribution at time epoch k. This is calculated from the target distribution at time 0 and known dynamics. In figure 2(a) we provide a demonstration (based on a sample of 1000 particles) of how this prior may change during the course of the simulation (see section 5.1 for details of the scenario).

At each time epoch k, the measure of performance is then taken to be the largest target location RMSE bound, b_k, given by:

$$b_k \triangleq \max\left[\sqrt{J_k^{-1}(1,1)}, \sqrt{J_k^{-1}(3,3)}\right] \quad (10)$$

where $J_k^{-1}(i,i)$ denotes the ith diagonal entry of J_k^{-1}. $J_k^{-1}(1,1)$ (respectively $J_k^{-1}(3,3)$) is the mean squared error (MSE) bound on the estimation of x_k (respectively y_k).

3.3 Sensor Flight Paths

3.3.1 Discretising The Decision Process

We will discretise the decision process in the same way as the measurement process. At time epoch k, each sensor can perform a manoeuvre which controls its subsequent motion between epochs k and $k+1$, and most importantly, specifies the position of the sensor at the time of the next measurement. It is assumed that each manoeuvre occurs instantaneously and the sensor then has constant velocity on each leg (i.e. between epochs k and $k+1$)[3].

We let a_k denote the vector prescribing the manoeuvres performed by each sensor at time epoch k. The optimisation problem is then to select a series of actions $\{a_k : k \geq 0\}$, for each sensor, to control the overall measures of performance, $\{b_k : k \geq 1\}$. The optimisation of each sensor trajectory is constrained by its manoeuvrability and maximum possible acceleration/deceleration, together with its minimum and maximum speeds.

3.3.2 Single Step Planning

In single-step planning we will use a step-by-step optimisation approach, in which manoeuvre $a_k^\star$ is chosen to minimise b_{k+1}, i.e.

$$a_k^\star \triangleq \text{argmin}_{a_k}\left[b_{k+1}\right] \quad (11)$$

[2]We note that an alternative approach, given in the original framework [4], creates "renewals" in the sensor management algorithm by generating posterior target state estimates as measurements become available. This approach has been shown to improve the effectiveness of the sensor management algorithm still further [13].

[3]We note that the theory presented in this paper supports a number of motion models, and is not dependent on using this slightly artificial acceleration profile.

This approach was also used to generate the sensor trajectories in [12].

3.3.3 Two-Step Planning

We note that making manoeuvre a_k at time k has an effect on both b_{k+1} and the future performance measures b_r, $r > (k+1)$. The reason for this is that the sensor manoeuvrability/acceleration is constrained. Hence, each manoeuvre will affect the possible trajectory over a number of future legs.

In two-step planning we select the manoeuvre, $a_k^\star(2)$ at time k in order to control both b_{k+1} and the value of the measure at the next measurement epoch, i.e.

$$a_k^\star(2) \triangleq \text{argmin}_{a_k}\left[b_{k+1} + b_{k+2}\right] \quad (12)$$

Now, let $a_{k+1}|a_k$ denote an action at time $k+1$ given the action a_k at time k. We note that in two-step planning on the second (final) step we simply have to optimise b_{k+2}, i.e.

$$\hat{a}_{k+1} \triangleq \text{argmin}_{a_{k+1}|a_k}\left[b_{k+2}(a_k; a_{k+1})\right] \quad (13)$$

where the notation, b_k has been adjusted to note the dependency on the actions at times k and $k+1$. We note that $\hat{a}_{k+1}$ is dependent on (i.e. is a function of) a_k.

Hence we can adopt a two-step dynamic programming approach of first determining $\hat{a}_{k+1}$ and the associated PCRLB measure value:

$$\hat{b}_{k+2}(a_k) \triangleq b_{k+2}(a_k; \hat{a}_{k+1}) \quad (14)$$

and then determining $a_k^\star(2)$ from

$$a_k^\star(2) = \text{argmin}_{a_k}\left[b_{k+1} + \hat{b}_{k+2}(a_k)\right] \quad (15)$$

The two-step planning sensor path is then given by $\{a_k^\star(2) : k \geq 0\}$. We note that calculation of $\hat{a}_{k+1}$ is merely an intermediate step, and $\hat{a}_{k+1}$ does not form part of the final solution.

3.3.4 Generalisation: Multi-Step Planning

In the case R-step planning, $a_k^\star(R)$ is chosen such that

$$a_k^\star(R) = \text{argmin}_{a_k}\left[b_{k+1} + \sum_{j=1}^{R-1} b_{k+1+j}\right] \quad (16)$$

A simple extension of equations (13) – (15) gives us a dynamic programming scheme for the general R-step planning problem. However, this approach suffers a combinatorial explosion as the number of additional time steps increases, because each additional control, $\hat{a}_{k+r}$ ($r = 1, \ldots, R-1$) must be calculated as a function of all previous actions $a_k, \ldots, a_{k+r-1}$. This makes the technique impractical.

Alternatively, we could search for $a_k^\star(R)$, and the additional controls at time epochs $k+1, \ldots, k+R-1$ simultaneously by using an evolutionary search technique, such as a genetic algorithm [15].

4 An Efficient Search Technique

4.1 Approach

In this paper we perform only single-step and two-step planning. We are then faced with the problem of optimising either a single action vector, $a_k^\star$, or using a two stage approach in which we first optimise $\hat{a}_{k+1}$ and then determine $a_k^\star(2)$.

At each time step we determine the manoeuvre of each of the sensors sequentially using an iterative scheme. In this scheme we in turn determine the optimal manoeuvre of each sensor given the "current" manoeuvres of each of the other sensors, and loop around until convergence is achieved[4] (i.e. no sensor deviates its course from that determined on the last iteration).

We now introduce an efficient search algorithm that will enable us to quickly determine the required sensor manoeuvres. We describe how this algorithm works by means of an example. In figure 1(a) we show the first stage of the search. The region divided into 4 quadrants gives the potential locations of a sensor at the time of the next measurement $k+1$, given the current sensor state (at time k) and maximum possible acceleration/manoeuvre. We divide this region into the 4 quadrants by selecting 9 positions as shown. Each position also has a velocity associated with it, specified by the position of the sensor at time k and the manoeuvre, a_k necessary to reach this location at time $k+1$.

For each quadrant we determine the performance measure b_{k+1} at each of its corners[5] and then select the quadrant with the lowest average value. We then divide this quadrant into 4 in the same way (see figure 1(b)) and repeat the process. At each stage we keep a note of the best found solution. The procedure terminates after either a fixed number of divisions, or if the best solution found has not improved for a pre-specified number of iterations. The complete search is shown in figure 1(c).

4.2 Comparison with Enumerative Method

To show the effectiveness of the efficient search algorithm we compared it to an algorithm that uses an enumerative grid search to determine (near) optimal sensor trajectories (see figure 1(d)). In figure 2 we show the results of this comparison when the two techniques are applied to a single-sensor problem in order to find single-step planning solutions. The enumerative technique searches for near-optimal solutions by dividing the region prescribing the potential sensor locations into a grid of 101×101 locations and (at each stage) recording the best solution.

We see from figures 2(b) and figure 2(c) that the sensor paths are almost identical. Unsurprisingly there is also little difference between the RMSE bounds determined using the two approaches (see figure 2(d)). However, the efficient search technique is far more economical than the enumerative method. Indeed, in these simulations we allow a maximum of 6 divisions, meaning we calculate no more than 34 $(9+(5\times5))$ PCRLB values at each stage. This compares to the enumerative technique which calculates more than 10^4 values during each optimisation.

4.3 Generalisation

The technique described in section 4.1 searches a two-dimensional region by selecting 9 points that define the perimeter of 4 quadrants and then focusing attention on one of the quadrants, on which the procedure is repeated. This technique works well for this type of problem, and therefore enables us to determine near-optimal solution for the single-step and two-step planning problems.

However, to apply this technique to problems in which the search space has more than two dimensions requires dividing this higher dimensional space into sub-regions. It is not clear how we should then select the perimeter points.

5 Simulations

5.1 Scenario

The target moves with nearly constant-velocity (CV) [16], with a power spectral density of $l = 1 \times 10^{-8}$. The initial target state has a Gaussian distribution with mean $\bar{X}_0 = (1000\ -40\ 1000\ -40)^T$ and covariance $C_0 = diag(10, 1, 10, 1)$. The time between successive measurement epochs is $T = 5$ seconds. Distances are in metres, and velocities are in metres per second. In figure 2(a) we show 1000 evolutions of the target path.

We track the target for 150 seconds (30 manoeuvres), using either 1, 2 or 3 sensors. The minimum detection range is set at $r = 1000$ metres and the bearing error standard deviation of target generated measurements is $\sigma = 0.01$ radians. The initial sensor states are as follows:

$$X_0^1(S) = (1800\ -40\ 2000\ -40)^T \quad (17)$$

$$X_0^2(S) = (2000\ -40\ 2000\ -40)^T \quad (18)$$

$$X_0^3(S) = (1600\ -40\ 2000\ -40)^T \quad (19)$$

Each sensor has a minimum speed of 50 m/s and a maximum speed of 200 m/s. Sensor platforms can manoeuvre up to $\pi/10$ radians and change their speed by a maximum of 10 m/s at each decision epoch.

5.2 Results

We use the efficient search algorithm to calculate one-step and two-step sensor paths for the 1, 2 and 3 sensor scenarios. Results for the 2 sensor scenario are given in figure 3, with a summary of the overall results given in table 1 and table 2. In each case we also perform tracking, using a Particle Filter (PF) with 10,000 sample points (see [17] for a comprehensive review of PF techniques) and filter RMSEs

[4] Or we reach a maximum number (here = 10) of iterations.

[5] We note that the value of this measure also depends on the "candidate" locations of the other sensors.

are compared to the PCRLB. In the PF we take the Importance Density to be the prior, and we use Sampling Importance Resampling to produce a sample of equally weighted particles that approximates the posterior target PDF.

We see from figures 3(c) – (d) that in the 2 sensor scenario, two-step planning gives improved performance when compared to single-step planning, both in terms of the values of the PCRLB and the filter RMSEs (see also table 1). Unsurprisingly, two-step planning also performs better in the other sensor scenarios, with the PCRLB showing an improvement of up to 28% (1 sensor comparison: see table 2), and filter RMSEs being up to 41% lower (again, see 1 sensor comparison in table 2). We also observe that as the sensor number increases, performance improves both in single-step and two-step planning.

6 Conclusions

We have considered the problem of determining sensor trajectories in bearings-only tracking of an uncertain target. The basis of our technique is to control a measure of estimation error based on the PCRLB, and we present both single-step and multi-step planning approaches. We also introduce an efficient search technique that allows us to quickly perform the necessary optimisation(s). Simulation results show that performance (in terms of both the PCRLB and filter RMSEs) improves both with multi-step planning and as sensor numbers increase, as one would expect.

Acknowledgements

This research was sponsored by the United Kingdom Ministry of Defence Corporate Research Programme CISP.

References

[1] H. Van Trees, "*Detection, Estimation, and Modulation Theory*", Wiley, New York, 1968.

[2] B. Ristic, S. Zollo, and S. Arulampalam, "*Performance Bounds for Manoeuvring Target Tracking Using Asynchronous Multi-Platform Angle-Only Measurements*", Proceedings of 4th International Conference on Information Fusion, Montreal, Québec, Canada, 2001.

[3] A. Nehorai, and M. Hawkes, "*Performance Bounds for Estimating Vector Systems*", IEEE Transactions on Signal Processing, **48**(6), pp. 1737–1749, 2000.

[4] M. L. Hernandez, T. Kirubarajan, and Y. Bar-Shalom, "*Multisensor Resource Deployment Using Posterior Cramér-Rao Bounds*", IEEE Transactions on Aerospace and Electronic Systems, (to appear), 2004.

[5] P. Tichavsky, C. H. Muravchik , and A. Nehorai, "*Posterior Cramér-Rao Bounds for Discrete-Time Nonlinear Filtering*", IEEE Transactions on Signal Processing, **46**(5), pp. 1386–1396, 1998.

[6] P. R. Horridge, and M. L. Hernandez, "*Multistatic Radar Resource Management*", Signal and Data Processing of Small Targets (ed. O. Drummond), Proceedings of SPIE, **5204**, pp. 583–594, 2003.

[7] R. Tharmarasa, T. Kirubarajan, and M. L. Hernandez, "*Large-Scale Optimal Sensor Array Management for Multitarget Tracking*", submitted to: IEEE Transactions on Aerospace and Electronic Systems, 2004.

[8] J. P. Le Cadre, and C. Jauffret, "*Discrete Time Observability and Estimability Analysis for Bearings-Only Target Motion Analysis*", IEEE Transactions on Aerospace and Electronic Systems, **33**(1), pp. 178–201, 1997.

[9] J. P. Le Cadre, "*Optimization of the Observer Motion for Bearings-Only Target Motion Analysis*", Proceedings of the 36th Conference on Decision and Control, pp. 3126–3131, San Diego, CA, USA, 1997.

[10] Y. Oshman, and P. Davidson, "*Optimization of Observer Trajectories for Bearings-Only Target Localization*", IEEE Transactions on Aerospace and Electronic Systems, **35**(3), pp. 892–902, 1999.

[11] J. M. Passerieux, and D. Van Cappel, "*Optimal Observer Maneuver for Bearings-Only Tracking*", IEEE Transactions on Aerospace and Electronic Systems, **34**(3), pp. 777–788, 1998.

[12] M. Smith, C. Angell, M. L. Hernandez, and W. J. Oxford, "*Improved Data Fusion Through Intelligent Sensor Management*", Signal Processing, Sensor Fusion, and Target Recognition XII, Proceedings of SPIE, **5096**, pp. 115–126, 2003.

[13] P. R. Horridge, and M. L. Hernandez, "*Performance Bounds for Angle-Only Filtering with Application to Sensor Network Management*", Proceedings of the 6th International Conference on Information Fusion, Cairns, Queensland, Australia, 2003.

[14] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, "*Numerical Recipes in C*", MIT Press, 1992.

[15] D. Goldberg, "*Genetic Algorithms in Search, Optimisation and Machine Learning*", Addison-Wesley, 1989.

[16] Y. Bar-Shalom, and X. Li, "*Estimation and Tracking: Principles, Techniques and Software*", Artech House, 1993.

[17] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "*A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking*", IEEE Transactions on Signal Processing, **50**(2), pp. 173–188, 2002.

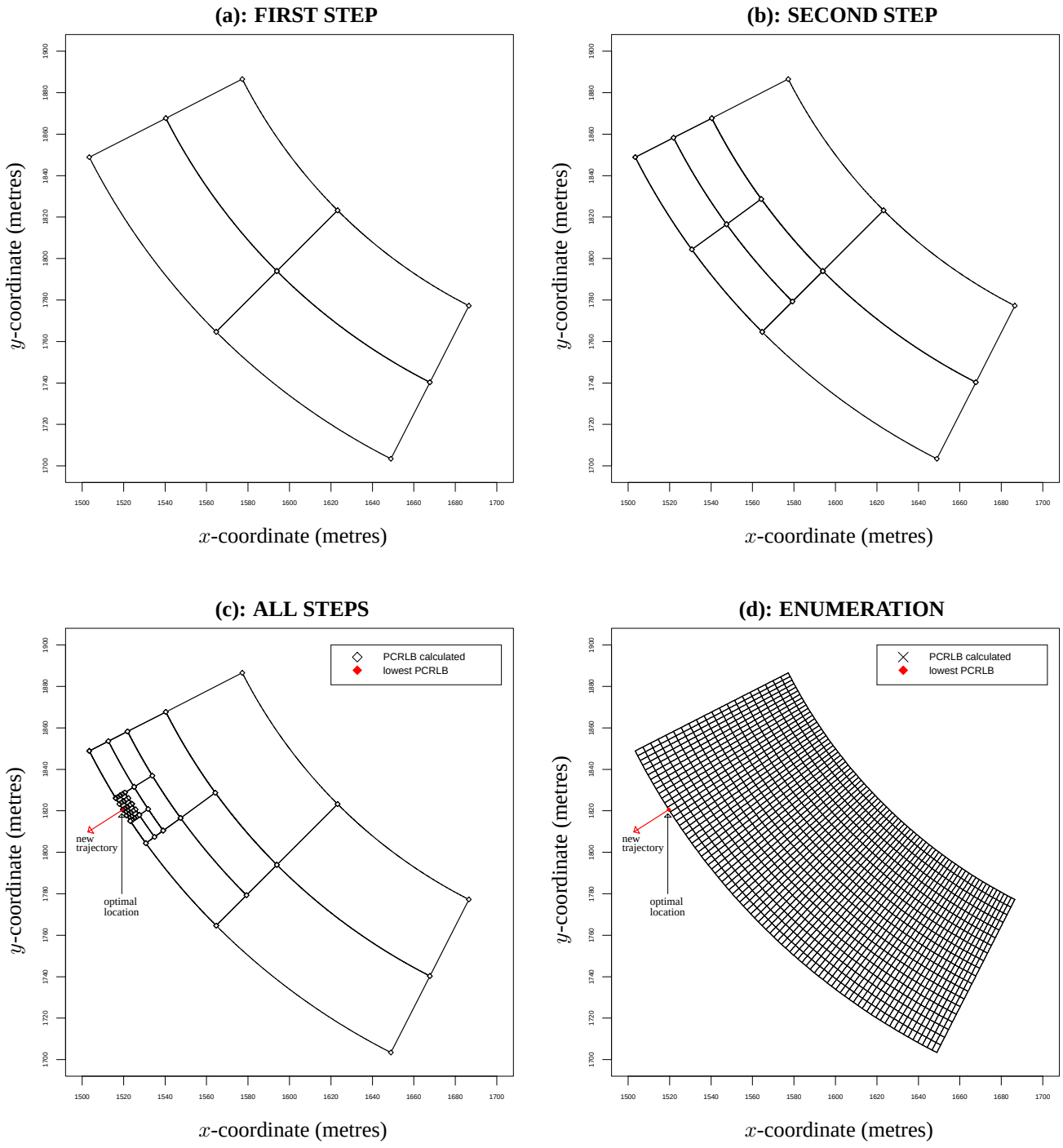

Figure 1: **(a) – (c):** demonstration of the efficient optimisation algorithm, **(d):** grid search (here a grid of 80×20 cells is shown).

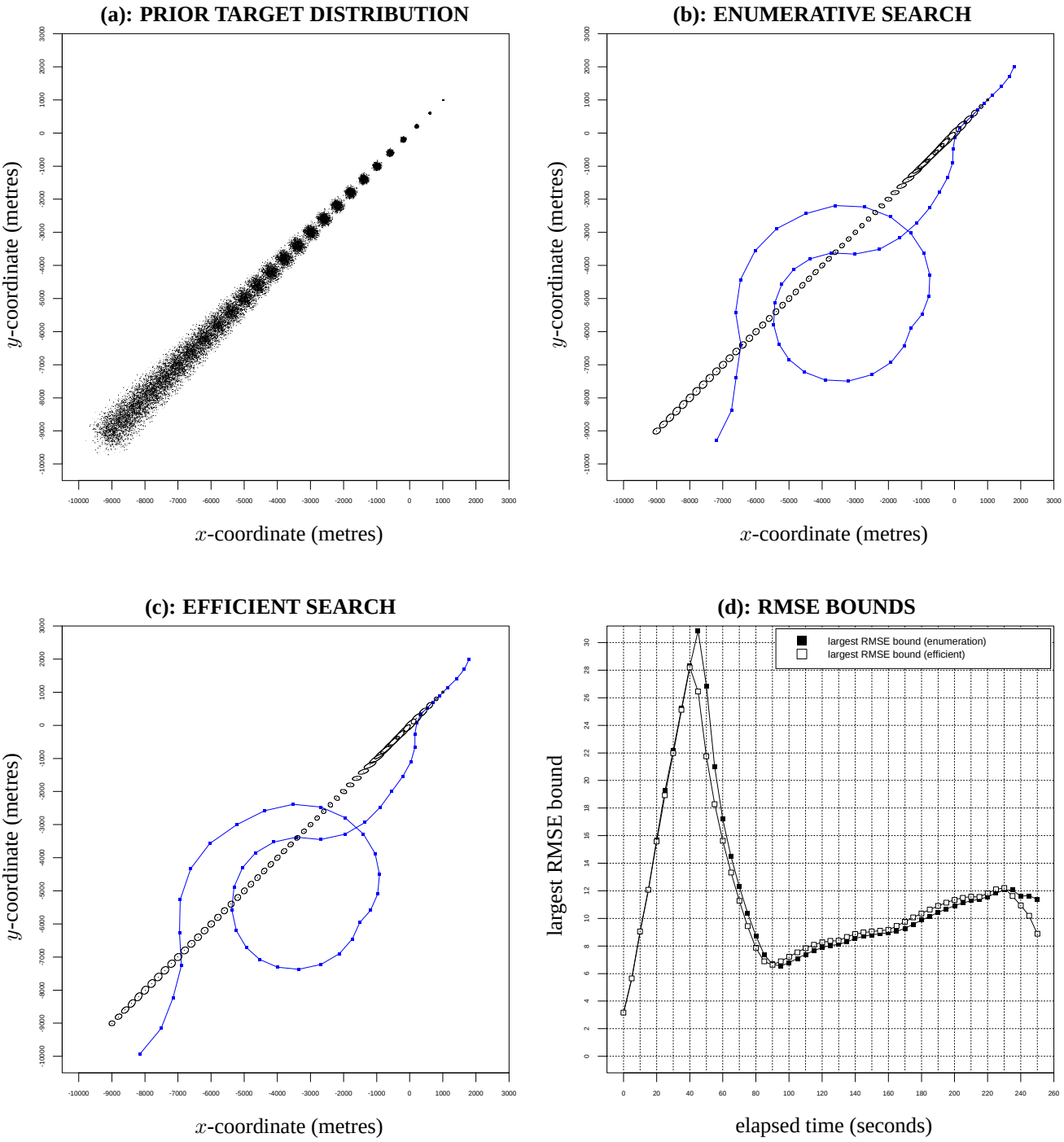

Figure 2: **(a):** the prior target distribution used in calculating the sensor trajectories (here we show the prior at decision/measurement epochs $k = 2i$, $i = 0, \ldots, 25$), **(b) – (c):** the sensor trajectories calculated under the two optimisation schemes, each ellipse gives 10 times the PCRLB, **(d):** a comparison of the RMSE bounds in each case.

Table 1: Performance summary, figures in brackets are filter RMSEs averaged over 100 runs of 30 time steps. Non-bracketed figures are RMSE bounds, averaged over the 30 time steps. All values are in metres.

Direction	1 Sensor		2 Sensors		3 Sensors	
	Single-Step	Two-Step	Single-Step	Two-Step	Single-Step	Two-Step
x-coordinate	12.3 (13.9)	11.9 (13.2)	7.1 (7.6)	6.8 (7.5)	5.9 (6.9)	5.0 (6.1)
y-coordinate	13.9 (18.1)	10.1 (10.7)	7.3 (9.1)	6.7 (7.1)	6.0 (6.8)	5.1 (7.0)

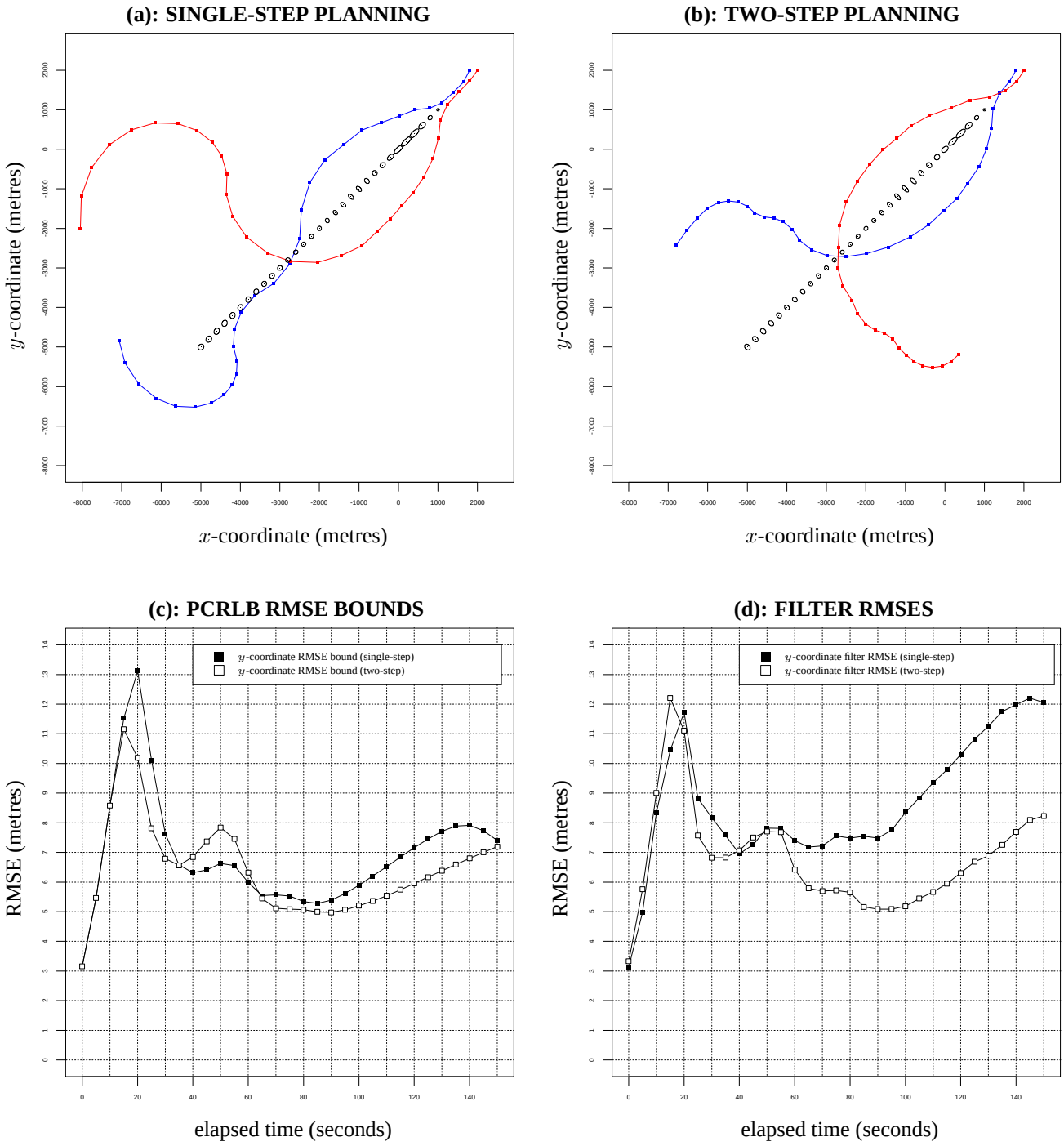

Figure 3: **(a) – (b):** trajectories of 2 sensors with single-step and two-step planning respectively, each ellipse again gives 10 times the PCRLB. Comparisons of the y-coordinate RMSE bounds and filter RMSEs are given in **(c)** and **(d)** respectively.

Table 2: The percentage improvement(s) of two-step planning over single-step planning in each sensor scenario. Bracketed figures relate to the filter RMSEs, with non-bracketed figures showing percentage improvements in the RMSE bounds.

Direction	% Improvement of two-step planning		
	1 Sensor	2 Sensors	3 Sensors
x-coordinate	3.6 (4.5)	5.5 (2.0)	15.1 (11.1)
y-coordinate	27.6 (40.6)	8.1 (21.7)	15.2 (-2.6)