2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Optimized Data Selection for Nonlinear Filtering

Jean-Baptiste Soubaras[1], Clément Chahbazian[2]

***Abstract*— Nonlinear estimation is ubiquitous in control and signal processing. It aims to estimate the most probable state of a system and its range of uncertainty by fusing data from multiple sensors over time using a filter. Although many efficient filters exist in the literature, their computational cost may increase when the set of data to process is significant. Besides, some data can be redundant or bring little information to the estimation of the state. In that case, their processing is costly and does not contribute to the filter's estimation. This paper introduces a semi-heuristic method to select a relevant subset of observations from a more extensive set of available data, using a cost function based on the approximation of the Cramer-Rao Lower Bound. This approach is adopted on an angles-only optimal navigation scenario where an extensive signal set is available. The results show that the filter achieves close-to-optimal accuracy for a lower computational cost.**

I. Introduction

Data fusion algorithms are critical to the functioning of autonomous systems. They define numerical methods that estimate kinematic variables, environment maps, or structural parameters commonly used for control, path planning, and decision-making. In the case of a nonlinear estimation problem, a data fusion algorithm typically estimates hidden variables in real-time by merging the time-integrated prediction of a dynamical model with the measurements of aiding sensors. Nonlinear state estimation is still an active field of research. Among a vast ecosystem of methods, the most popular approaches are based on Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter (PF) [1]–[3].

However, new applications often involve strongly nonlinear sensors producing large data sets that must be processed in real-time. Such an amount of data can lead to a prohibitive computational cost for embedded systems where the hardware resources are limited.

Recent works address this problem by adding a pre-processing step that selects the relevant observation for the filter. Previous works on tracking applications investigated this idea in [4]. Besides, a recent study showed that a well-chosen combination of measurements leads to a suitable estimation accuracy on a navigation problem [5]. Both works involve the Cramer-Rao Lower Bound (CRLB), defined as the asymptotic minimum variance of an unbiased estimator. In practice, the CRLB characterizes the quantity of information from a set of observations given prior knowledge of the estimated state. Moreover, [5] introduced a cross-entropy algorithm to perform the combinatorial optimization required to select the relevant observations. This approach obtained close-to-optimal accuracy with a small subset of measurements. However, cross-entropy algorithms can be computationally intensive for real-time embedded applications. Besides, computing the Cramer-Rao Lower Bound online is challenging in practice. Therefore, [5] performs with offline computations where the system's trajectory is known, and computing resources are not limited.

The approach developed in this paper addresses this problem by using a scalable combinatorial optimization method in which the cost function uses an approximation of the Cramer-Rao Lower Bound. The rationale of this approach is gradually detailed and compared with optimal solutions.

Section II provides information theory concepts and states the estimation problem with the cost function to be optimized. Then, Section III presents the main contribution of this paper. It details the proposed filtering architecture and focuses on two optimization algorithms: Brute Force which is a reference providing optimal selections, and Variable Neighborhood Search (VNS), a meta-heuristic suited to large-dimension problems. Section IV explains the simulated navigation scenario, and Section V presents the numerical results, which investigate the validity and benefits of the approach. Eventually, Section VI concludes the paper.

II. Problem Statement

This section introduces the problem addressed in this paper. It describes the Extended Kalman Filter and the Cramer-Rao Lower Bound, which defines the dynamic decision.

A. State Estimation

Let the discrete-time hidden state $\{x_k\}_{k\in\mathbb{N}} \in \mathbb{R}^d$ be a Markov process according to a set of observations $\{y_k\}_{k\in\mathbb{N}} \in \mathbb{R}^m$:

$$\begin{cases} x_{k+1} = f(x_k, u_{k+1}, n^q_{k+1}), \\ y_{k+1} = h(x_{k+1}, n^r_{k+1}), \end{cases} \tag{1}$$

where $\{y_k\}_{k\in\mathbb{N}}$ are mutually independent measurements, u_k is a control input vector, (n^q_{k+1}, n^r_{k+1}) are centered noise vectors, and (f, h) two possibly nonlinear smooth functions. Extended Kalman Filter (EKF), based on Kalman Filter [6], is a popular method to estimate the state's probability density function when the noises are mutually independent and Gaussian:

$n^q_k \sim \mathcal{N}(0, Q_k)$, $n^r_{k+1} \sim \mathcal{N}(0, R_{k+1})$ and $\mathbb{E}\left[n^q_k {n^r_k}^T\right] = 0$.

[1] Jean-Baptiste Soubaras is with ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France jean-baptiste.soubaras@ensta-paris.fr

[2] Clément Chahbazian is with MBDA France, dpt. of Guidance Control and Navigation , Le Plessis Robinson, France clement.chahbazian@mbda-systems.com

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

To that extent, EKF tracks the conditional expectation $\hat{x}$ and covariance matrix P with a propagation step:

$$\begin{aligned} \hat{x}_{k+1|k} &= f(\hat{x}_{k|k}, u_{k+1}), \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + Q_{k+1}, \end{aligned} \tag{2}$$

and an update step:

$$\begin{aligned} K_{k+1} &= P_{k+1|k} H_{k+1}^T S_{k+1}^{-1}, \\ S_{k+1} &= H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}, \\ P_{k+1|k+1} &= (I - K_{k+1} H_{k+1}) P_{k+1|k}, \\ \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1} \left(y_{k+1} - h(\hat{x}_{k+1|k}) \right), \end{aligned} \tag{3}$$

where F_k and H_{k+1} are the Jacobian matrices of f and h respectively computed at $\hat{x}_{k|k}$ and $\hat{x}_{k+1|k}$:

$$F_k = \left. \frac{\partial f(x)}{\partial x} \right|_{x=\hat{x}_{k|k}} ; \; H_{k+1} = \left. \frac{\partial h(x)}{\partial x} \right|_{x=\hat{x}_{k+1|k}} . \tag{4}$$

B. Cramer-Rao Lower Bound

Let $x_k \in \mathbb{R}^d$ be a random vector and $\hat{x}_{k|k}$ be an unbiased estimator of x_k from the observation vector $y_k \in \mathbb{R}^p$. The posterior Cramer-Rao Lower Bound (CRLB) [7] is a positive semi-definite matrix defined as the inverse of the Fisher information matrix J_k and verifies the inequality:

$$P_{k|k} \geq J_k^{-1}, \tag{5}$$

in the sense that the eigenvalues of $P_{k|k} - J_k^{-1}$ are positive, and J_k is computed from the joint probability density function $p(x_k, y_k)$:

$$J_k \triangleq -\mathbb{E} \left[\frac{\partial^2 \log p(x_k, y_k)}{\partial x_k^2} \right]. \tag{6}$$

Assume that x and y verify the discrete-time system (1). The approach proposed in [8] enables a recursive computation of the Fisher information matrix at time k denoted J_k:

$$J_{k+1} = D_k^{22} - D_k^{12} \left(J_k + D_k^{11} \right)^{-1} D_k^{21}, \tag{7}$$

where:

$$\begin{cases} D_k^{11} &= -\mathbb{E} \left[F_k^T Q_k^{-1} F_k \right], \\ D_k^{12} &= -\mathbb{E} \left[F_k^T Q_k^{-1} \right] = {D_k^{21}}^T, \\ D_k^{22} &= -Q_k^{-1} - \mathbb{E} \left[H_{k+1}^T R_{k+1}^{-1} H_{k+1} \right], \end{cases} \tag{8}$$

with F_k and H_{k+1} defined in (4).

C. Dynamic Decision Problem

Let $\mathcal{Y}_k = \{y_k^1, \cdots, y_k^N\}$ be a set of N available observations from different measurement sources and $\mathcal{S}_k \subset \mathcal{Y}_k$ be a subset of $K \leq N$ selected vectors. The subset $\mathcal{S}_k$ is characterized by the selection vector $\mathfrak{s}_k \in \{0,1\}^N$, which is a binary sequence of the selected observations. If $y_k^i \in \mathcal{S}_k$, then $\mathfrak{s}_k^i = 1$. Otherwise, if $y_k^i \notin \mathcal{S}_k : \mathfrak{s}_k^i = 0$.
The decision problem addressed in this paper is to find at every time step k the subset $\mathcal{S}_k \subset \mathcal{Y}_k$ of K observations, which brings the best contribution to an estimation filter. Once the observations are selected, the measurement model of (1) is modified accordingly to the selected observations:

$$\begin{cases} x_{k+1} = f(x_k, u_{k+1}, n_k^q), \\ \tilde{y}_{k+1} = \tilde{h}(x_{k+1}, n_{k+1}^r), \end{cases} \tag{9}$$

where $\tilde{y}$ and $\tilde{h}$ are taken only for the corresponding non-zero values of $\mathfrak{s}_k$.

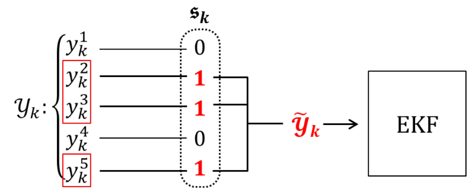

Fig. 1: Illustration of the filtering with selected measurements. In this example, the selection vector is $\mathfrak{s}_k = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \end{bmatrix}$.

D. Optimization Criterion

The criterion to be optimized should reflect the quantity of information brought by a selection. Hence, the approach described in this paper is based on an approximation of the CRLB conditionally to the selection vectors $\mathfrak{s}_k$. The Fisher Information Matrix associated to a selection is denoted $J_k(\mathfrak{s}_k)$, and computed with the recursive process (7) where H_{k+1} is replaced with $\tilde{H}_{k+1}$ defined from (9):

$$\tilde{H}_{k+1} = \left. \frac{\partial \tilde{h}(x)}{\partial x} \right|_{x=\hat{x}_{k+1|k}} . \tag{10}$$

The cost function ϕ for a selection at a time k is the determinant of approximated CRLB, which is positive since CRLB is a semi-definite positive matrix:

$$\phi(\mathfrak{s}_k) = \det(J_k(\mathfrak{s}_k))^{-1}. \tag{11}$$

The choice of using the determinant of CRLB in the cost function is motivated by its link with the Fisher Information Matrix, which translates as the maximum information available. Besides, the determinant accounts for the non-diagonal and diagonal terms of CRLB without need for weighting according to each variable since it is multiplicative. Moreover, it is homogeneous of degree d.
Other approaches use the trace of CRLB as cost function [4] [9]. Although this approach is mathematically relevant, it implies summing variables of different types and overlooks the impact of the correlation of the state variables.
Considering a discrete finite sequence of T steps, with N measurement sources at disposition and $K \leq N$ measures to be selected, the problem to solve is formalized as:

$$\begin{cases} \arg \min\limits_{\mathfrak{s}_1, \cdots, \mathfrak{s}_T} & \sum\limits_{k=0}^{T} \det(J_k(\mathfrak{s}_k))^{-1}, \\ \text{s.t. } \forall k \leq T, & \|\mathfrak{s}_k\|_1 \leq K. \end{cases} \tag{12}$$

III. Measurement Selection and Filtering

This section represents the main contribution of this paper. It aims to solve the dynamic decision problem with a pragmatic combinatorial optimization method. First, the decision problem (12) is developed with the Bellman equation and discussed with different levels of assumptions. Then, two algorithms are discussed, leading to the resolution method proposed in this paper.

A. Dynamic programming

The Bellman dynamic programming equation introduced in [10] subdivides this problem into several simpler ones:

$$\forall k \leq T : \begin{cases} \mathfrak{s}_k & = \arg\min_{\mathfrak{s}} B_k(\mathfrak{s}), \\ B_k(\mathfrak{s}) & = \det(J_k(\mathfrak{s})^{-1}) + \min_{\mathfrak{w}} B_{k+1}(\mathfrak{w}). \end{cases} \tag{13}$$

According to Bellman, solving (13) is equivalent to solving (12). However, the term $\min_{\mathfrak{w}} B_{k+1}(\mathfrak{w})$ depends on future states and observations, which are unknown at time k. Hence, this paper considers a simplified approach consisting in minimizing only the first term - i.e., considering that if $\mathfrak{s}_k$ minimizes $\det(J_k)^{-1}$, it minimizes $\min_{\mathfrak{w}} B_{k+1}(\mathfrak{w})$. This assumption prioritizes the present selection over the future one, which is reasonable when the future trajectory is unknown.

This assumption rewrites the problem (12) as, for all $k \leq T$:

$$\begin{cases} \mathfrak{s}_k = \arg\min_{\mathfrak{s}} \det(J_k(\mathfrak{s}))^{-1}, \\ \text{s.t. } \|\mathfrak{s}\|_1 \leq K. \end{cases} \tag{14}$$

B. Solving algorithms

The sequel presents two methods that solve the Bellman equation.

1) Brute Force: The Brute Force method solves the optimization problem by computing the objective value for each possible selection and returns the best one. At each selection step, there are $\binom{N}{K}$ subsets of size K to be processed. Thus, the computational complexity of the Brute Force approach for a selection is $O(KN^K)$.

This method is suitable for small problem instances, and it is optimal. However, it becomes computationally intractable for extensive instances due to its complexity. In that case, the meta-heuristic presented in the sequel is more adapted, given its complexity.

Algorithm 1: Brute Force

Result: $\mathfrak{s}_k$
Selection:
- Compute the cost (11) for each selection vector.
- Return the selection vector with the lowest cost.

2) Variable Neighborhood Search: The Variable Neighborhood Search is a combinatorial optimization algorithm introduced in [11], which consists of improving a feasible selection by local research. This algorithm needs an initial feasible selection and a definition of a family of neighborhoods depending on a size parameter l. The neighborhoods considered in this paper will be for a size $l \in [\![0, K]\!]$:

$$V_l(\mathfrak{s}_k) = \left\{\mathfrak{w}_k \in \{0,1\}^N ; \|\mathfrak{w}_k - \mathfrak{s}_k\|_1 = 2l\right\}, \tag{15}$$

that is to say, the selection vectors of K sources with exactly $K - l$ of them differing from $\mathfrak{s}_k$.

The parameters of VNS are:

- l_{max}: the maximum size for a neighborhood;
- $\forall l \in [\![1, l_{max}]\!]$, M_l: the maximum number of selections generated for each size l.

At each iteration, the algorithm disposes of a reference selection $\mathfrak{s}_k^{ref}$ (initially the one given as input) and a current size l, initiated at 1. It generates a random selection in the neighborhood of $\mathfrak{s}_k^{ref}$ by exchanging l selected sources with l unselected sources: $\mathfrak{w} \sim \mathcal{U}(V_l(\mathfrak{s}_k^{ref}))$. Then, the selection with the smallest cost becomes the new reference.

If the reference selection has stayed the same for M_l iterations, the size of the neighborhood is incremented $l \leftarrow l + 1$, making the algorithm more exploratory. The process stops when l reaches $l_{max} + 1$; the output is the last reference selection.

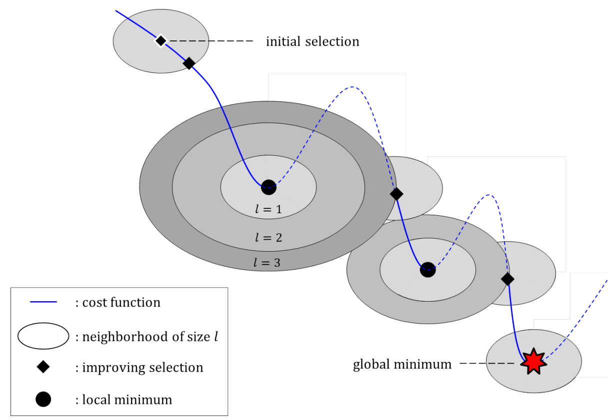

Fig. 2: Illustration of the Variable Neighborhood Search, iteratively evolving towards improving selections. Increasing the size of the neighborhood allows to escape terminating on a local minimum. The cost function is represented as continuous here for a better visualization.

Algorithm 2: Variable Neighborhood Search

Result: $\mathfrak{s}_k$
Input: initial selection $\mathfrak{s}$
Parameters: $l_{max}; M_1, \cdots, M_{l_{max}}$
Selection:
$l := 1$; # *current size of the neighborhood*
$j := 0$ # *number of iterations at current size*
While $l < l_{max}$:
  Randomly sample $\mathfrak{w} \in V_l(\mathfrak{s})$
  If $f(\mathfrak{w}) < f(\mathfrak{s})$:
    $\mathfrak{s} \leftarrow \mathfrak{w}$; $l \leftarrow 1$; $j \leftarrow 0$
  Else:
    $j \leftarrow j + 1$
    If $j > M_l$:
      $l \leftarrow l + 1$; $j \leftarrow 1$
$\mathfrak{s}_k := \mathfrak{s}$

The VNS algorithm improves the input selection iteratively, and its performance depends on the quality of the initial selection, which can be obtained with an appropriate heuristic. A suitable choice is to take the output of the previous selection assuming the state has had little change between selections. The first selection at $k = 0$ can be obtained with any heuristic based on prior knowledge and sensor models of the system. Otherwise, using Brute Force determines the optimal initial selection, which is acceptable for the computational load as it occurs only once. The computational complexity of one selection made by VNS is $O(NK)$.

C. Selection-Based Filtering

This paper introduces a measurement selection step between an estimation filter's propagation and update steps. Algorithm 3 details the proposed approach based on the Extended Kalman Filter.

Algorithm 3: Extended Kalman Filter with Selection

Result: $(\hat{x}_{k|k}, P_{k|k})$, $k \in [1, N]$

Propagation step:

$\hat{x}_{k+1|k} = f(\hat{x}_{k|k})$

$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_{k+1}$

Selection step:

Get $\mathfrak{s}_{k+1}$ from Algorithm 1 or 2

Compute $\tilde{y}_{k+1}$, $\tilde{h}$ and $\tilde{H}_{k+1}$ from (9)(10)

Update step:

$K_{k+1} = P_{k+1|k} \tilde{H}_{k+1}^T \left(\tilde{H}_{k+1} P_{k+1|k} \tilde{H}_{k+1}^T + R_{k+1} \right)^{-1}$

$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \left(\tilde{y}_{k+1} - \tilde{h}(\hat{x}_{k+1|k}) \right)$

$P_{k+1|k} = (I - K_{k+1} \tilde{H}_{k+1}) P_{k+1|k}$

However, the selection step requires the Cramer-Rao Lower Bound to build the cost function for the optimization algorithm. The online estimation of the CRLB can be costly and difficult to implement in practice. Hence, the key approximation made in Algorithm 3 is to use the estimated covariance matrix P instead of the CRLB. This approximation holds, assuming that estimation errors remain small, which is often the case for EKF.

Besides, the filtering process's complexity is expected to decrease with the selection step performed with VNS. Indeed, the EKF update step has a cubic complexity with the dimension of the measurement vector [12].

IV. Application to Angles-Only Navigation

A. Context

This testing scenario describes the optimal navigation of an Unmanned Aerial Vehicle (UAV) which trajectory is shown in Figure 3. The ground frame $[e]$ has a fixed origin with respect to the Earth, and its axes point East, North, and Up. Also, $[e]$ is assumed to be inertial in this study. The frame attached to the vehicle is denoted $[b]$, and its axes points forward, rightward, and downward. The notations and conventions of this section related to navigation variables are detailed in [12].

This navigation problem is to use an EKF to estimate the UAV's position x_{eb}^e, velocity v_{eb}^e, and attitude Euler angles θ_b^e given a kinematics model and the measurements of an antenna carried by the UAV. This sensor observes the angles-of-arrival of signals emitted from a set of $N = 30$ known landmarks, which positions are denoted $[p_n^e]_{n \in [1, N_l]}$.

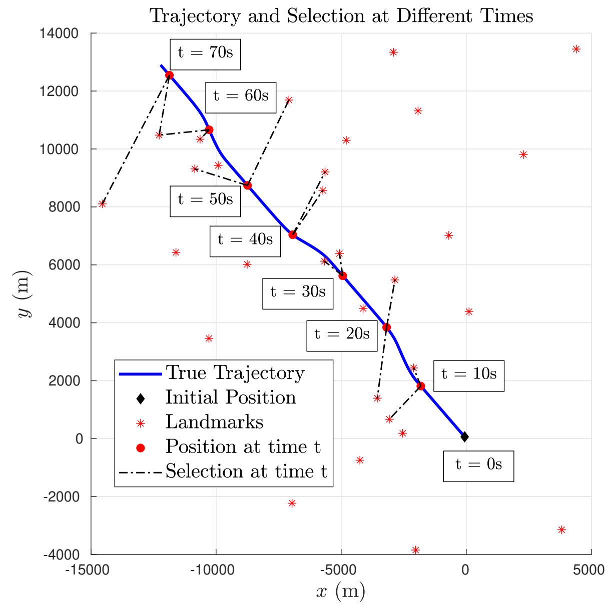

Fig. 3: True trajectory in the (x, y) plan with the position of the landmarks and several selections.

The measurement model is y_k where the measurement model for the n^{th} beacon is:

$$y^n = \begin{bmatrix} \text{arctan2}\left(\Delta_y^n, \Delta_x^n\right) \\ \text{arctan2}\left(\Delta_z^n, \rho^n\right) \end{bmatrix}. \tag{16}$$

with:

$$\begin{cases} \rho^n &= \sqrt{(\Delta_x^n)^2 + (\Delta_y^n)^2}, \\ \Delta^n &= C_e^b (p_n^e - x_{eb}^e), \end{cases} \tag{17}$$

and $(\Delta_x^n, \Delta_y^n, \Delta_z^n)$ denote the components of Δ_n^b. Note that Δ^n is the relative distance between a landmark and the UAV resolved in the body frame $[b]$, C_e^b is the rotation matrix obtained from the attitude Euler angles, and arctan2 is such that $\forall (x, y) \neq (0, 0)$:

$$\text{arctan2}(y, x) = \begin{cases} \text{sign}(y) \arctan\left|\frac{y}{x}\right| & x \leq 0, \\ \text{sign}(y)\frac{\pi}{2} & x = 0, \\ \text{sign}(y)(\pi - \arctan\left|\frac{y}{x}\right|) & x \leq 0. \end{cases} \tag{18}$$

B. Measurement model Jacobian matrix

The Jacobian matrix for the full measurement model is calculated with respect to every beacon:

$$H = \begin{bmatrix} H^{1^T} & \cdots & H^{N^T} \end{bmatrix}^T, \tag{19}$$

where, the Jacobian matrix for the n^{th} beacon is:

$$H^n = \Psi^n \begin{bmatrix} [\Delta^n]_\times & 0_3 & -I_3 \end{bmatrix}, \tag{20}$$

with Ψ_n such that:

$$\Psi^n = \begin{bmatrix} -\frac{\Delta_y^n}{(\rho^n)^2} & \frac{\Delta_x^n}{(\rho^n)^2} & 0 \\ \frac{\Delta_x^n \Delta_z^n}{\rho^n ||\Delta^n||^2} & \frac{\Delta_y^n \Delta_z^n}{\rho^n ||\Delta^n||^2} & \frac{-\rho^n}{||\Delta^n||^2} \end{bmatrix}. \tag{21}$$

Sensor Parameters	
Sensors rate (Hz)	10 Hz
IMU noise (1σ)	Gyrometer: $2°/h$, Accelerometer: $10^{-3} m/s^2$
AOA noise (1σ)	Azimuth: $0.6°$, Elevation: $0.6°$
Model Parameters	
Init. error (1σ)	Attitude: $0.6°$, Velocity: $1 m/s$, Position: $50m$
Model noise (1σ)	Azimuth: $0.6°$, Elevation: $0.6°$
Selection Parameters	
Observations	$N = 30$ available landmarks
Selection	$K = 2$ selected landmarks
Monte Carlo	100 Simulations

TABLE I
Simulation and filters parameters.

C. Landmarks Selection Problem

In the simulation scenario illustrated in Figure 3, the vehicle observes the angles of arrival from $N = 30$ beacons. Hence, the dimension of the measurement vector y is $2N = 60$, which can lead to prohibitive computational costs in a filtering process. This problem is addressed with the developments from Section III by selecting a subset of landmarks at every time step and only using their signals to update an EKF.

Hence, Algorithm 3 is applied for the state estimation and landmarks selection with the parameters of Table I.

V. Simulations Methodology and Results

This section details the numerical results for the Section IV scenario. It focuses on the Cramer-Rao Lower Bound and Algorithm 3, both tested with three selection approaches:

- With all the measurements available ;
- With two measurements selected from Brute Force ;
- With two measurements selected from VNS.

These tests focus on the accuracy of the estimation filter executed with the selection step performed by the VNS method. Given the low complexity of VNS, this approach is expected to be suitable for real-time implementation. The Brute Force method gives optimal selections, which will be used to assess the accuracy of VNS, and estimation using all the available measurements is used as a lower bound to evaluate the highest possible accuracy.

A. Instantaneous Cost Function

Figure 4 displays the plots of the instantaneous cost function from (11). As expected, the cost function is minimal when the 30 landmarks are considered. Besides, VNS and Brute Force cost functions are close, meaning that VNS selections are close-to-optimal with respect to Brute Force. Hence, VNS is relevant to the selection step and provides accurate solutions to the optimization problem.

B. Cramer-Rao Lower Bound

Figure 5 presents the Cramer-Rao Lower Bound plots. During the simulation, the bounds corresponding to VNS and Brute Force selections are close to the ones computed with all the measurements available. Therefore,

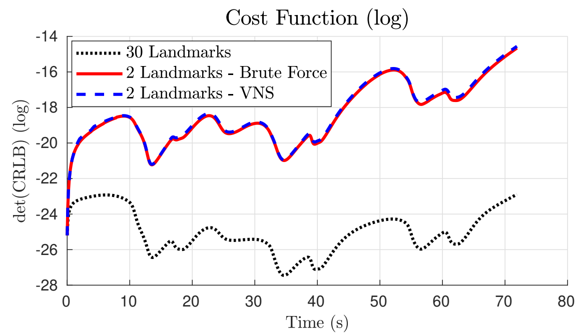

Fig. 4: Illustration of the evolution of the optimization criterion in the calculus of CRLB. Brute Force and VNS show similar accuracy.

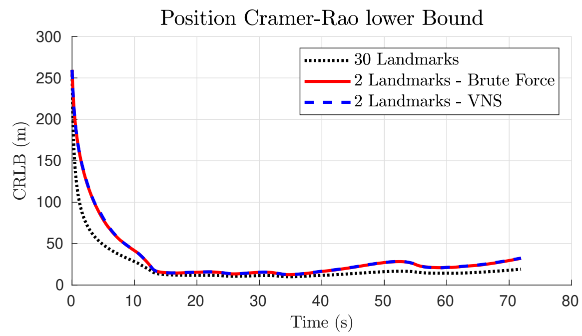

Fig. 5: Cramer-Rao Lower Bound (CRLB) for the L_2 norm of the position vector for 30 landmarks, two landmarks selected with Brute Force and two landmarks selected with VNS. The small subset of selected observations provides most of the information. Besides, VNS and Brute Force lead to similar results. Hence, in this scenario, VNS returns close-to-optimal selections with respect to Brute Force.

in this scenario, the selection has a limited impact on the information received by the filter.

Besides, given the strong convergence CRLB in every case, the filtering process based on EKF is expected to perform well, as it receives sufficient information from the selected measurements.

C. Filtering and Selection

The results of the selection and filtering process from Algorithm 3 are displayed in Figure 6. The Root Mean Square Error (RMSE) plots show that the different selection steps lead to comparable accuracy for Brute Force and VNS. This result was expected since their CRLB displayed in Figures 5 and their instantaneous cost functions displayed in Figure 4 are close. Hence, in this scenario, VNS is suitable to perform the selection step since it is accurate compared to Brute Force, leading to good filtering process accuracy. This result is confirmed by the proximity of the plot of the RMSE with the CRLB calculated for 30 landmarks.

Besides, Figure 3 displays a few of the selections computed by Brute Force. It illustrates that the selection can be counter-intuitive due to non-diagonal terms in the CRLB and prior uncertainties on the state.

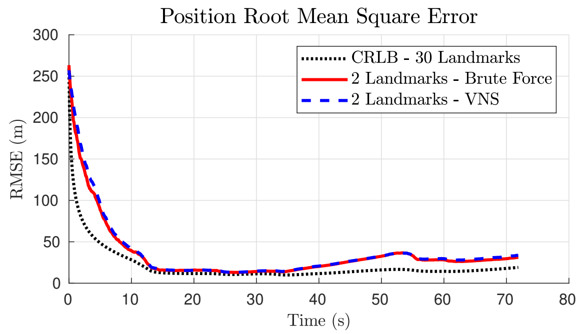

Fig. 6: Root Mean Square Error (RMSE) for the L_2 norm of the position vector estimated with Extended Kalman Filter for 30 landmarks, two landmarks selected with Brute Force and two landmarks selected with VNS. The filters running on the selected subset of measurements present good accuracy compared to the bound calculated for the 30 landmarks.

Selection Algorithm	**VNS**	**Brute Force**	**None**
Avg. Execution Time	$\mathbf{1.53}$ ms	4.81 ms	4.44 s

TABLE II: Average duration of a single selection for different methods. A time step lasts 100 ms between two IMU measurements.

The RMSE of each method is consistent with the CRLB, showing that the efficient data selection allows similar accuracy. Besides, the selection leads to a lower computational cost, as detailed in Table II. This result was expected given the computational complexity of the selection algorithms discussed in Section III and the complexity of EKF without the selection step. It can be seen that the Brute Force is not suitable for a real-time implementation since its execution time is higher than the time step and would result in the creation of a delay. On the opposite, the VNS runs faster and allows finding the selection in a reasonable time. It is then suitable for real-time implementation. Moreover, since for K fixed, the Brute Force computational complexity is $O(N^K)$ and the VNS is $O(N)$, the computation time of the Brute Force would increase drastically with K. In contrast, the VNS would be expected to run for an adequate duration.

VI. Conclusion

This article introduces a method to select a relevant measurement subset among an extensive amount of data for the update step of an estimation filter. This approach is based on the Bellman dynamic programming equation. It involves an optimization problem in which the cost function is defined from an approximation of the Cramer-Rao Lower Bound. Two resolution algorithms were studied: Brute Force, which provides optimal selections, and VNS, which is a metaheuristic. These methods were applied to an angles-only navigation problem addressed with an Extended Kalman Filter. The simulation results show that the VNS performs faster than the Brute Force approach with similar results. Besides, adding a selection step with VNS in Extended Kalman Filter maintains the estimation accuracy. Therefore, this paper provides a generic method to alleviate the computation when dealing with an intractably large data set. By solving or simplifying adequately the Bellman equation, this approach can be applied on a large variety of estimation problems.

In future works, alternative metrics for the cost function will be investigated. Besides, a resolution of the partial or complete Bellman equation will also be explored in problems where future states are partially predictable at any given time. Moreover, other types of filters (e.g., Unscented Kalman Filter and Particle Filter) can be tested on more realistic scenarios.

Acknowledgment

The authors are thankful to MBDA France, ONERA the French Aerospace Lab and AID the Innovation and Defense Agency for their support on this research work.

References

[1] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Defense, Security, and Sensing*, 1997.
[2] M. S. Arulampalam, S. Maskell, N. J. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, pp. 174–188, 2002.
[3] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.
[4] Z. Wang, X. Shen, P. Wang, and Y. Zhu, "The cramér–rao bounds and sensor selection for nonlinear systems with uncertain observations," *Sensors*, vol. 18, no. 4, 2018.
[5] C. Musso, F. Dambreville, and C. Chahbazian, "Filtering and sensor optimization applied to angle-only navigation," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, 2021, pp. 1–8.
[6] R. E. Kalman, "A new approach to linear filtering and prediction problems" transaction of the asme journal of basic," 1960.
[7] H. Cramér, *Mathematical methods of statistics*. Princeton University Press, 1999.
[8] P. Tichavský, C. Muravchik, and A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal Process.*, vol. 46, pp. 1386–1396, 1998.
[9] J. P. Helferty and D. R. Mudgett, "Optimal observer trajectories for bearings only tracking by minimizing the trace of the cramer-rao lower bound," *Proceedings of 32nd IEEE Conference on Decision and Control*, pp. 936–939 vol.1, 1993.
[10] R. Bellman, *Dynamic Programming*, 1957.
[11] N. Mladenović and P. Hansen, *Computers and Operation Research*, 1997.
[12] P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*. Artech, 2013.