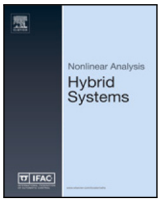
Nonlinear Analysis: Hybrid Systems 55 (2025) 101546

Contents lists available at ScienceDirect

Nonlinear Analysis: Hybrid Systems

journal homepage: www.elsevier.com/locate/nahs



Globally convergent path-aware optimization with mobile robots[☆]

T. Sântejudean [a,b,*], Ş. Ungur [a], R. Herzal [a], I.-C. Morărescu [b,a], V.S. Varma [b,a], L. Buşoniu [a,*]

[a] *Technical University of Cluj–Napoca, Romania*
[b] *Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France*

ARTICLE INFO

Keywords:
Global optimization
Source seeking
Convergence analysis
Mobile robots

ABSTRACT

Consider a mobile robot that must navigate as quickly as possible to the global maxima of a function (e.g. density of seabed litter, pollutant concentration, wireless signal strength) defined over its operating area. This objective function is initially unknown and is assumed to be Lipschitz continuous. The limited velocity of the robot restricts the next samples to neighboring positions, and to avoid wasting time and energy, the robot's path must be adapted as new information becomes available. The paper proposes two methods that use an upper bound on the objective to iteratively change the position targeted by the robot as new samples are acquired. The first method is FTW, which Turns When the best value seen so far of the objective Function is larger than the bound of the current target position. The second is FTWD, an extension of FTW that takes into account the Distance to the target. Convergence guarantees are provided for both methods, and a convergence rate is proven to characterize how fast the FTW suboptimality decreases as the number of samples grows. In a numerical study, FTWD greatly improves performance compared to FTW, outperforms two representative source-seeking baselines, and obtains results similar to a much more computationally intensive method that does not guarantee convergence. The relationship between FTW and FTWD is also confirmed in real-robot experiments, where a TurtleBot3 seeks the darkest point on a 2D grayscale map.

1. Introduction

Consider an autonomous mobile robot that must navigate as quickly as possible to the global maxima of some position-dependent physical quantity (objective function) by sampling it online during its mission. The function, defined over the search area of the robot, is unknown at the beginning of the experiment. An important constraint is the robot's limited velocity, which means it cannot sample the function arbitrarily far. Instead, the next sample is restricted to a neighborhood of the robot's current position. To reduce time and energy consumption [1], the robot should update its trajectory towards the most likely optimum location using all information collected so far, i.e. the pairs of sampled positions and their corresponding values. This setting is formally known as *path-aware global optimization* [2], and has many applications: finding the maximal pollution source [3], the highest odor plume to detect leak sources [4] like gas emission points in landfill sites [5], or the strongest signal strength in sensor networks for robot

[☆] This work was been financially supported from project DECIDE, no. 57/14.11.2022, funded under the PNRR I8 scheme by the Romanian Ministry of Research, Innovation, and Digitisation; and from SeaClear2.0, a project that received funding from the European Climate, Infrastructure and Environment Executive Agency (CINEA) under grant agreement No 101093822.

[*] Corresponding authors.
E-mail addresses: tudor.santejudean@aut.utcluj.ro (T. Sântejudean), ungur.da.david@student.utcluj.ro (Ş. Ungur), herzal.eu.radu@student.utcluj.ro (R. Herzal), constantin.morarescu@univ-lorraine.fr (I.-C. Morărescu), vineeth.satheeskumar-varma@univ-lorraine.fr (V.S. Varma), lucian@busoniu.net (L. Buşoniu).

https://doi.org/10.1016/j.nahs.2024.101546
Received 2 November 2023; Received in revised form 21 March 2024; Accepted 8 September 2024
Available online 13 September 2024

localization and communication purposes [6,7]. Another application of particular importance is the search for marine litter, as in the SeaClear 2.0 project (https://www.seaclear2.eu/). In this example, one robot may map areas with the highest litter density [8,9], and share this mapping information with another robot tasked with collecting the litter [10].

The key objectives in path-aware global optimization are (i) to design methods for a mobile robot to quickly find all global optima of a Lipschitz-continuous, but otherwise arbitrarily complex objective function (nondifferentiable, any amount of local or global optima); and (ii) to guarantee asymptotic convergence and convergence rates.

In this paper, to address objective (i), two methods are presented to solve path-aware global optimization. These methods are related to a branch-and-bound global optimization technique called deterministic optimistic optimization (DOO), which under Lipschitz continuity guarantees convergence at known rates to a global optimum [11,12]. The two methods exploit the Lipschitz continuity of the objective function to recompute at each step an upper bound on it, from all the samples seen so far. The first method drives the robot optimistically towards target positions corresponding to the maxima of the upper bound. At each step, it checks whether the bound at the target position dropped below the best objective value seen so far, and if so, it updates its target to a new best-bound position. Hence, this method is called Turn When the Function is larger than the bound, FTW for short, and was introduced in our preliminary conference version [2]. The second method is novel, and is motivated by the fact that it often makes sense to visit closer points before distant ones, even though their bounds might be slightly smaller. Therefore, this second method modifies the target selection criterion so that large-bound points at smaller Distances to the robot's position are ranked higher, from where the acronym FTWD follows.

A key contribution of this paper is to thoroughly analyze both of these algorithms; such an analysis was missing from [2]. Convergence rates to the global optima are given for FTW, thereby successfully achieving objective (ii). A complexity measure that drives the convergence rate is defined, and a close relationship to the measure defined for the original DOO in [12] is established. The asymptotic analysis discussed earlier largely focuses on FTW because analyzing FTWD would require additional regularity assumptions on the objective function, and therefore reduce generality. Instead, a simpler form of convergence for FTWD (together with FTW) is proven only in the approximate case when the bound is defined on a finite grid.

Nevertheless, a numerical study shows that FTWD performs significantly better than FTW, and almost as well as an algorithm called Path-Aware Optimistic Optimization [13], which is much more computationally intensive and has no convergence guarantees. In the same numerical study, FTWD outperforms two representative, state-of-the-art baselines from source seeking control [14,15] by always finding all global maxima. The convergence rate of FTW is experimentally illustrated. Finally, the simulation results from the comparison between FTW and FTWD are confirmed in a TurtleBot3 experiment, in which the real robot successfully searches for the darkest point of a 2D grayscale surface.

Related work

The closest settings to path-aware optimization are source and extremum seeking, which aim to optimize a process variable (e.g. performance/cost function) of a dynamical system, often through approximate gradient climbing, when only online measurements of the objective are available. These techniques are usually employed to regulate nonlinear plants to reach their optimal operating point [16,17], or – similar to the present paper – applied to mobile robots when searching for the optimum of some physical quantity [18,19]. In the latter case, source/extremum seeking does not usually assume that absolute position coordinates are available (e.g. mobile robots might operate in a GPS-restricted area) [20,21], although exceptions exist [18,19].

The objectives of this paper are optimization-focused and are different from the typical goals in source/extremum seeking, which heavily focus on analyzing practical asymptotic stability [20–23]. Moreover, the shape of the function is significantly more general than in many source/extremum seeking methods [17,20,21,24], in which the function must often be differentiable and radially decreasing around the optimum.

Another major point of differentiation is that the current work provides asymptotic convergence rates, which to the authors' knowledge are very rare in source/extremum seeking, and have only been previously provided under the assumption that the global optimum is unique [21,25]. Two limitations of the proposed approach are that, like in [18,19], absolute robot positions are required; and that the asymptotic analysis disregards the transients, as is not uncommonly done in source seeking, e.g. [16,22].

Additional work related to this paper can be found in coverage problems, where robots monitor an area and build a surveillance path to quickly update a map of this area [26–28]. Another sample-based approach used to examine a physical quantity in minimum time with robot sensors can be found in informative path planning [29]. For instance, agricultural sites can be monitored with a high-resolution depth camera mounted on a UAV [30]. Well-established SLAM methods can be employed if both mapping and robot localization in unknown environments are desired [31,32]. However, none of these methods directly optimize a function defined over the robot's space. In the multiagent framework, a heuristic method searching for the maximum signal strength of an antenna was presented in [33]. The method uses particle swarm optimization to find the global optimum without imposing regularity assumptions on the signal function, although unlike the current work, that paper does not provide a convergence analysis and is not suitable for single-robot optimization.

While the Path-Aware Optimistic Optimization method previously proposed by two of the authors [13] works well in practice, it provides no convergence guarantees, and it is computationally much more expensive than FTW(D).

Next, some background on DOO and preliminary algorithmic development is given in Section 2, followed by the presentation of FTW and FTWD in Section 3. Sections 4 and 5 provide convergence guarantees and rates, particularly for FTW, for which a complexity measure that drives this rate is formally defined. Numerical results and real-robot experiments are presented in Sections 6 and 7 respectively, while Section 8 summarizes the paper and gives possible extensions for future work.

2. Background and preliminaries

This section briefly introduces DOO, which represents the foundation of the proposed methods, and then formally defines the path-aware optimization setting.

2.1. Deterministic optimistic optimization

Consider a compact and connected, n-dimensional state space X. Practical examples of X include polyhedra (not limited to hyper-rectangles or spheres) such as geographical regions. Over this state space, an objective function $f : X \to \mathbb{R}$ is defined. DOO [12] is a global optimization algorithm belonging to the branch-and-bound class that aims to find the optima $x^* \in X^* := \arg\max_{x \in X} f(x)$ of the objective from successive function evaluations. It sequentially splits the search space X into progressively finer partitions, and samples to further expand only the sets associated with the largest upper bound values on f. After a numerical budget has been exhausted, the algorithm approximates the maximum as the location x with the largest f value evaluated so far.

Assumption 1 (*Smoothness*)**.** The objective function f is globally Lipschitz continuous:

$$\|f(x_1) - f(x_2)\| \leq M\|x_1 - x_2\|, \forall x_1, x_2 \in X, \tag{1}$$

where M denotes the Lipschitz constant and $\|\cdot\|$ is the Euclidean norm.

Even though DOO analysis in [12] requires the Lipschitz-continuity assumption in (1) to hold only locally around the maxima $x^* \in \arg\max_{x \in X} f(x)$, the property is needed here to hold globally in order to develop the analysis in the following sections. Note that typically, a global optimization algorithm either requires sampling the entire space, or relies on a smoothness assumption of the objective function [14], like Lipschitz continuity [19,34]. This assumption is necessary because without it, an unexplored region may contain positions where the function experiences an arbitrarily rapid rate of change, potentially concealing a global optimum. This restriction is realistic for many physical phenomena such as the distribution of ocean litter or gas and heat diffusion, which gradually spread over time until reaching a rather smooth steady-state distribution [19].

An alternative approach to the partition splitting in DOO will be used here: the construction of a so-called "saw-tooth" upper bound [12], defined as $B_k : X \to \mathbb{R}$ so that:

$$f(x) \leq B_k(x) := \min_{x_s \in S_k} [f(x_s) + \|x - x_s\|], \ \forall x \in X, \tag{2}$$

where $S_k = \{x_1, \dots, x_k\}$ is the set of states sampled up to step k, with k indexing the last function evaluation; see Fig. 1 for an example. At each iteration, the next target (state to be sampled) is given by the formula:

$$x_t := \arg\max_{x \in X} B_k(x). \tag{3}$$

This method will be called sawtooth DOO. Note that B_k is lowered (refined) implicitly via (2) with each new sample added to S_k.

2.2. Path-aware optimization

Consider a mobile robot characterized by positions $x \in X$ and control inputs $u \in U$, which searches for the global maxima $x^* \in X^*$ of the function f. Next, define the discrete time dynamics $g : X \times U \to X$ so that:

$$x_{k+1} = g(x_k, u_k), \tag{4}$$

where k denotes the time step. Note that the state signal consists of only robot positions and the robot moves with first order dynamics, although this could be generalized like in [13]. In the search for x^*, the robot picks targets $x_t \in X$ with (3). When driving towards these targets, the robot uses dynamics in (4) by applying the control action leading closest to x_t among all available actions:

$$u_k = \arg\min_u \|x_t - g(x_k, u)\|. \tag{5}$$

Most often, due to velocity constraints in dynamics (4), multiple time steps and thus control actions will be required for the robot to reach a given target x_t.

Assumption 2 (*Reachability*)**.** $\exists R > 0$ such that $\forall x \in X$ and $\forall x' \in \mathbb{B}(x, R)$, $\exists u \in U$ such that $g(x, u) = x'$, where $\mathbb{B}(x, R)$ is the n-dimensional (hyper)sphere centered at x and of radius R.

Assumption 2 ensures that the robot can eventually reach all points within the set X. Such an assumption is not uncommon and was made e.g. for the source seeking method of [24]. When applied to the single-integrator dynamics (4), it provides a reasonable approximation of more complex behavior, e.g. second or third order dynamics [35], especially when robots move at relatively low velocities, their control actions are limited to simple maneuvers, or when the operating area is much larger than the robots' size [36,37]. Note that the convergence results of the following sections hold under Assumptions 1 and 2.

Path-aware global optimization [2,13] can be described as follows. The mobile robot aims to locate as quickly as possible all global maxima of objective function f, which per Assumption 1 is Lipschitz continuous with a known Lipschitz constant, but can

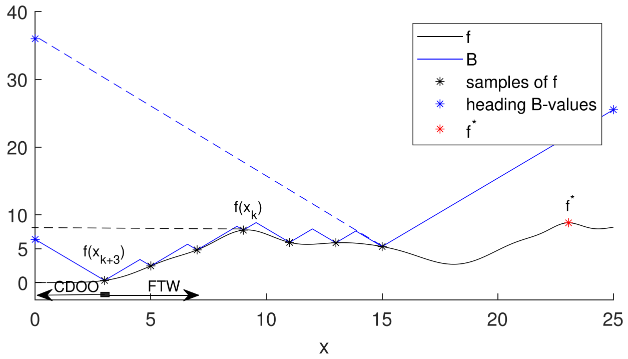


Fig. 1. 1D trajectory example showing the overcommitment issue faced by committed DOO (CDOO). FTW changes direction when its path clearly becomes suboptimal. (For interpretation of the usage of color in this figure, the reader is referred to the web version of this article.)

have properties such as non-differentiability and various local and global optima. The robot lacks prior knowledge of the function f, making the problem model-free optimization. Consequently, the robot learns the function online during a single trajectory, by observing samples at the positions it visits along this trajectory. However, due to constraints on the robot's dynamics (e.g., limited velocity), it can only sample at each step neighboring states instead of being able to sample arbitrarily distant next-step positions.

In this setting, the sampling strategy of sawtooth DOO becomes inappropriate as most often a maximal B-state is not reachable within one robot step, meaning that intermediate steps (samples) are required to reach these targets. The approach in which the robot picks a target according to (3), travels towards this point by taking intermediate steps, and only changes the trajectory once it has been reached, will be called committed DOO. Note that this approach is susceptible to overcommitment: even though the newly acquired samples may suggest the current trajectory has become suboptimal, the method is unable to make trajectory adjustments until the preset target is reached. Fig. 1 provides an example of a committed-DOO trajectory and gives more intuition on the overcommitment issue [2].

3. FTW and FTWD algorithms

Next, the main methods of the paper are presented: FTW and its extension, FTWD. FTW was introduced in the preliminary conference version [2], while FTWD is the main algorithmic novelty of this work.

FTW uses the sawtooth-DOO principle of refining with each new f-sample gathered the upper bound in (2), with the goal of focusing the search towards x^*. To tackle the danger of overcommitment, FTW continuously monitors the upper bound and function values. If after the latest sample the bound of the currently targeted state x_t (the last-chosen maximal-B location) becomes lower than an f-sample previously seen by the robot, the current path is clearly suboptimal. In such a case, the robot updates its target to the current maximum of B. This method will be called FTW, as the robot Turns When a previous Function value becomes larger than the current best upper bound.

Fig. 1 provides some intuition on the difference between committed DOO and FTW. The global maximum of the function is marked with a red star and denoted with f^*, the robot samples are marked with black stars, and the blue line represents the upper bound on f. The robot is heading towards the left endpoint of the search space $X = [0; 25]$, where initially the bound was maximal (marked with a blue star at the end of the blue dotted line). A future turnaround is exemplified: at step $k + 3$ the refined bound at the initial target point will become lower than the current function sample $f(x_k)$, as shown by the horizontal dotted line, and thus FTW will change direction towards the right. In contrast, committed DOO will continue the suboptimal trajectory until reaching the position initially targeted, thus wasting energy and time.

Algorithm 1 summarizes the FTW method. The method starts in position x_1 with no prior knowledge of f ($S_0 \leftarrow \emptyset$), apart from the Lipschitz constant M. At each step k, the robot takes a new sample $f(x_k)$, adds it to S_{k-1}, and updates both the upper bound B_k and $f_k^* = \max\{f(x_s)|x_s \in S_k\}$, the maximal f-value seen so far. The robot updates its current target x_t with (3) as soon as $f_k^* > B_k(x_t)$, or when x_t is reached. The control action in (5) guides the robot towards x_t, resulting in a new robot position x_{k+1}.

The algorithm stops when either the total number of iterations was exhausted or convergence was reached. Convergence is obtained when f_k^* becomes larger than all (or equal to some) B_k values. In that case, the position corresponding to f_k^* is returned as an approximation of the optimum.

A meaningful extension to FTW can be obtained by accounting for the distance to the (locally) maximal bound states while choosing the next target position. To understand why this approach might be preferred, consider the case when two or more, not necessarily global, optima in X are present. If FTW is used, the robot will continue to lower the bound around the first optimum until the next max-bound state is located in the neighborhood of the second optimum. The robot will refine next the bound states around the second optimum, until the first one becomes attractive again, and so on. This leads in practice to oscillating behavior: a trajectory that repeatedly passes through the high bound-regions of different optima, as can be seen on the left of Fig. 2.

Algorithm 1 FTW/FTWD

```
Input: search space $X$, dynamics $g$, Lipschitz constant $M$, maximum number of trajectory steps $N$
measure initial state $x_1$
initialize target $x_t = x_1$ and sample set $S_0 \leftarrow \emptyset$
for each step $k = 1, \dots, N$ do
    sample $f(x_k)$, add $x_k$ to $S_{k-1}$, obtaining $S_k$
    update max $f$-sample $f_k^* = \max_{x_s \in S_k} f(x_s)$ and upper bound $B_k$ (2)
    if $B_k(x_t) \le f_k^*$ then
        update target with:
            FTW: $x_t = \mathrm{argmax}_{x \in X}\, B_k(x)$
                or
            FTWD: $x_t = \mathrm{argmax}_{x \in X}\, D_k(x)$ from (6)
        if $B_k(x_t) \le f_k^*$ then
            convergence occurred, break loop
        end if
    end if
    find $u_k = \mathrm{argmin}_u\, ||x_t - g(x_k, u)||$ and apply $u_k$ to reach $x_{k+1}$
end for
return $\widehat{x^*} = \mathrm{argmax}_{x_s \in S_N}\, f(x_s)$.
```

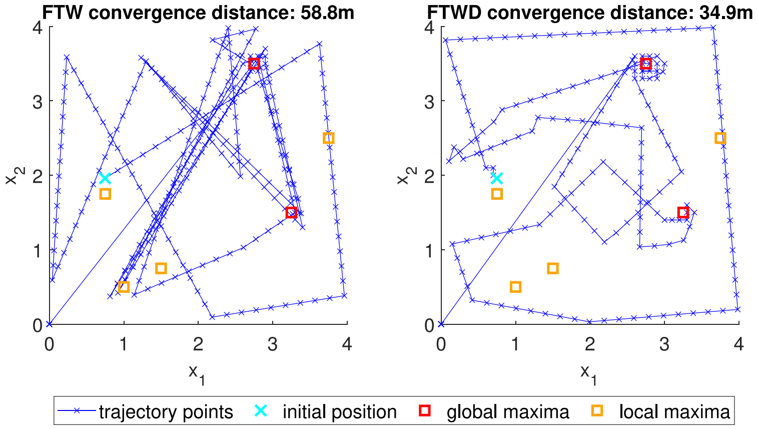


Fig. 2. Left: Illustration of the oscillating behavior of the robot trajectory for a FTW run. Right: The oscillating behavior is reduced for FTWD, so the trajectory length greatly decreases. Results are part of the FTW/FTWD comparison from Section 6.1, and details on the experiment are provided on page Section 6.1. (For interpretation of the usage of color in this figure, the reader is referred to the web version of this article.)

To reduce the oscillations, and get a trajectory similar to the one on the right of Fig. 2, one can consider also the distance (which is used here as a proxy for travel cost) required by the robot to reach the maximal-bound states. To achieve this, define the following quantity:

$$D_k(x) := \frac{B_k(x) - f_k^*}{\|x - x_k\|}, \tag{6}$$

where x_k is the current position of the robot, $B_k(x)$ is the bound in $x \in X \setminus \{x_k\}$, and f_k^* is the maximum sample seen so far (up to step k). The term f_k^* in (6) ensures that the robot will not target states where the bound is below the maximal value sampled so far. Then, the next target point is chosen as:

$$x_t \in \arg\max_{x \in X} D_k(x), \tag{7}$$

instead of the standard choice in (3). The new FTW variant, since it also accounts for the Distance, will be called FTWD, and a simulation comparison to FTW will be made in Section 6.1. Note that like for FTW, convergence of FTWD is reached when $B_k(x) \le f_k^*, \forall x \in X$.

Due to the non-trivial global maximization of the bounds B_k over the entire X, an approximate grid-based version of FTW and FTWD will be implemented in practice. In this case, B_k and D_k are evaluated over an equidistant grid X_g of resolution δ defined over X (the distance between adjacent points across each dimension of X is δ) and the optimum x^* will be found with an accuracy given by the resolution. The convergence for the grid-based methods is reached when $B_k(x) \le f_k^*, \forall\, x \in X_g$. Note that the results of Fig. 2 were generated with the grid-based algorithms.

4. Convergence guarantees

In this section, convergence guarantees are provided for sawtooth DOO, committed DOO and FTW, followed by a convergence proof applicable to the grid-based versions of the same methods, as well as of FTWD. The proof for FTWD is only given for its grid-based version, as in the continuous setting the convergence cannot be guaranteed without making additional assumptions on the function f, which would reduce generality. Such a setting is not in the scope of this paper, but may be studied in future work.

Theorem 1 (*Convergence of Sawtooth DOO and Committed DOO*). *The sequence of best function values obtained up until each step k, f_k^*, will converge to f^* as $k \to \infty$.*

Proof. f_k^* is convergent due to being monotonously increasing and bounded from above ($f_k^* \le f^*$). Suppose that $f_k^* \nrightarrow f^*$ in the limit $k \to \infty$. Thus, $\exists \varepsilon_f > 0$ s.t.:

$$f_k^* \to f^* - \varepsilon_f =: f_\infty^*. \tag{8}$$

Take any global maximum x^*, which by assumption is never sampled. Thus, for any sample x_{k+1}:

$$B_k(x_{k+1}) > B_k(x^*) \ge f^* \tag{9}$$

as otherwise x^* would be sampled. Let $x_s' \in S_k$ and denote by $B_k'(x_{k+1}) = f(x_s') + M \cdot \|x_{k+1} - x_s'\|$ the bound in x_{k+1} given by x_s'. Then, using (2) one can obtain:

$$B_k'(x_{k+1}) = f(x_s') + M \cdot \|x_{k+1} - x_s'\| \ge B_k(x_{k+1}). \tag{10}$$

Using $f_\infty^* \ge f(x_s')$ and (9), the last inequality leads to:

$$f_\infty^* + M \cdot \|x_s' - x_{k+1}\| > f^*. \tag{11}$$

Finally:

$$\|x_s' - x_{k+1}\| > \frac{f^* - f_\infty^*}{M} = \frac{\varepsilon_f}{M} > 0, \ \forall x_s' \in S_k. \tag{12}$$

Inequality (12) implies that at any k, x_{k+1} is placed at a distance greater than $\frac{\varepsilon_f}{M}$ from all states in S_k. This leads to a contradiction, as there is only a finite number of samples that can be taken in X such that the minimum distance between any new sample and those already available in S_k is greater than $\frac{\varepsilon_f}{M}$. Thus, the assumption initially made ($f_k^* \nrightarrow f^*$) is false and the convergence of sawtooth DOO is proven.

In committed DOO the robot pays the traveling cost to x_{k+1} in order to sample this maximal B-state. Thus, the set of samples contains (without being equal to) the set of max B-states recommended by the sampling strategy of sawtooth DOO. Nevertheless, the same proof line exploiting the inequalities (9)–(12) can be applied to prove that $\lim_{k\to\infty} f_k^* \to f^*$. □

Remark 1. The proof also implies that, for any $\varepsilon > 0$, f^* will be found with ε accuracy in a finite number $n_\varepsilon \in \mathbb{Z}_+$ of function evaluations.

Remark 2. The global maximum x^* used in the proof was an arbitrarily chosen element of the set X^*. Thus, the proof holds for any $x^* \in X^*$, and the sawtooth-DOO method will find all global maxima in X^* with arbitrary accuracy. This does not guarantee that a maximum x^* will eventually become a sampled state, but rather that the method is building a dense set of samples around optima.

The convergence guarantee of the FTW method will be given in the sequel. The following Lemma represents an intermediate step towards this.

Lemma 1. *Let $x_1, x_2 \in X$. Then:*

$$B_k(x_1) \le B_k(x_2) + M \cdot \|x_1 - x_2\|. \tag{13}$$

Proof. Let $x_s \in S_k$ be the sample that gives the bound $B_k(x_2)$. Thus:

$$B_k(x_2) = f(x_s) + M \cdot \|x_2 - x_s\|. \tag{14}$$

Using (2) with $x = x_1$:

$$B_k(x_1) \le f(x_s) + M \cdot \|x_1 - x_s\|, \tag{15}$$

and by the triangle inequality:

$$B_k(x_1) \le f(x_s) + M \cdot \|x_1 - x_2\| + M \cdot \|x_2 - x_s\|. \tag{16}$$

Finally, by (14):

$$B_k(x_1) \le B_k(x_2) + M \cdot \|x_1 - x_2\|. \quad □$$

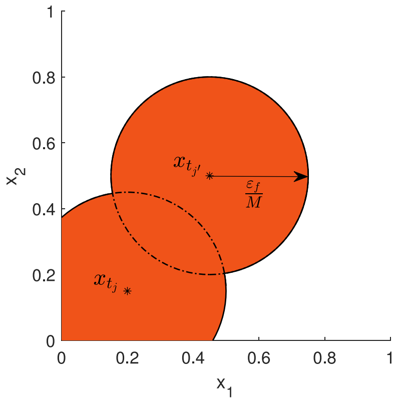


Fig. 3. The shaded area represents the union of two 2-dimensional balls (disks), inside which no future target points are located. Implicitly, all maxima of f will be found outside of the area covered by this inclusion.

Theorem 2 (*Convergence of FTW*). *In the FTW method* $\lim_{k\to\infty} f_k^* \to f^*$.

Proof. Similarly to the proof of Theorem 1, f_k^* is convergent and suppose that $f_k^* \to f^* - \varepsilon_f =: f_\infty^*$, where $\varepsilon_f > 0$. Different from that proof, the set of samples does not necessarily contain the max B-states recommended by the sampling strategy of DOO, due to possible FTW turnarounds when the target's bound becomes lower than f_k^*. Thus, the contradiction in Theorem 1 cannot be obtained in the same way because the max-bound recommendations could still be at a distance larger than $\frac{\varepsilon_f}{M}$ from all samples in S_k.

To obtain the contradiction, it is enough to prove that the total number of FTW turnarounds is finite, i.e., that $\exists k$ after which all max-bound recommendations will eventually become sampled states. Thus, FTW will asymptotically behave as the committed-DOO method, for which the convergence was already proven.

Suppose the contrary: that there exists an infinite number of FTW turnarounds. Denote by $x_{t,j}$ the j^{th} max bound state on which an FTW turnaround was performed, and by k_j the step index of the sampled state at which the turnaround from $x_{t,j}$ was done. Then:

$$B_{k_j}(x_{t,j}) < f_{k_j}^* \tag{17}$$

and a next target $x_{t,j+1}$ is chosen so that:

$$B_{k_j}(x_{t,j+1}) = \max_{x\in X} B_{k_j}(x). \tag{18}$$

Then, by taking x_1 equal to $x \in X$ and $x_2 = x_{t,j}$ in Lemma 1, it follows that:

$$B_{k_j}(x) \le B_{k_j}(x_{t,j}) + M \cdot \|x_{t,j} - x\|. \tag{19}$$

Consider at step k_j the ball $\mathbb{B}\left(x_{t,j}, \frac{\varepsilon_f}{M}\right)$, centered in $x_{t,j}$ and of radius $\frac{\varepsilon_f}{M} > 0$, and take x in this ball. As $\|x_{t,j} - x\| < \frac{\varepsilon_f}{M}$, using (17) in (19) leads to:

$$B_{k_j}(x) < f_{k_j}^* + M \cdot \frac{\varepsilon_f}{M} = f_{k_j}^* + \varepsilon_f < f^*. \tag{20}$$

Finally:

$$B_{k_j}(x) < f^* \le B_{k_j}(x^*), \forall\, x \in \mathbb{B}\left(x_{t,j}, \frac{\varepsilon_f}{M}\right), \tag{21}$$

which means that future max-B states chosen by FTW will not belong to $\mathbb{B}(x_{t,j}, \frac{\varepsilon_f}{M})$. Note that the size of each such ball is bounded from below by a positive quantity, as the radius $\frac{\varepsilon_f}{M} > 0$.

Future max B-states will not belong to the union of the previous balls either, since the bounds in these regions are below $f_{k_{j'}}^*$, $\forall j' > j$. Thus:

$$x_{t,j'} \notin \bigcup_{i=1}^{j} \mathbb{B}\left(x_{t,j}, \frac{\varepsilon_f}{M}\right), \forall j' > j. \tag{22}$$

Note that the balls defined above are not necessarily disjoint. Nevertheless, since $x_{t,j'}$ belongs to the exterior or the frontier of the above union (refer to Fig. 3 for some intuition), and given the assumption that there is an infinite number of turnarounds, leading to an infinite number of such balls, the union in (22) will grow infinitely large. The last statement contradicts the compactness property of the state space X. Thus, the number of FTW turnarounds is finite and the method will asymptotically behave like committed DOO, leading to the property $f_k^* \to f^*$ as in Theorem 1. □

Remark 3. In the proof of Theorem 2, a union of balls was considered to gradually restrict the regions containing future maximal-B states and implicitly x^*. Using Lemma 1, the bounds inside each ball $\mathbb{B}\left(x_s, \frac{f^*-f(x_s)}{M}\right)$ will be below f^*, $\forall x_s \in S_k$, and:

$$x_t \notin \bigcup_{s=1}^{k} \mathbb{B}\left(x_s, \frac{f^* - f(x_s)}{M}\right), \tag{23}$$

where x_t is any maximal-B state to be chosen as target by FTW at steps $k' \geq k$. In what follows, the union in (23) will be denoted by $\cup\mathbb{B}$. All the points outside $\cup\mathbb{B}$ have the bounds greater than f^* and, as the number of trajectory samples increases, the volume covered by $\cup\mathbb{B}$ will also increase. To get an empirical signal on the convergence rate to f^*, the rate at which the volume covered by $\cup\mathbb{B}$ grows will be experimentally studied in Section 6.3.

Theorem 3 (*Convergence of Sawtooth DOO, Committed DOO, FTW and FTWD on a Grid*)**.** *For each one of the methods mentioned, when bounds are evaluated on a grid X_g of resolution δ, f_k^* is convergent and* $\lim_{k\to\infty} f_k^* = f_\infty^* \geq f^* - M\frac{\sqrt{n}\delta}{2}$.

Proof. After a finite number of steps, $\exists x_t \in X_g$, a target chosen to be sampled for the second time. Note that this will happen at worst after sampling all the states on the grid X_g, which are finite in number as X is compact. If x_t is revisited, then $x_t \in S_k$ and:

$$B_k(x) \leq B_k(x_t) = f(x_t) \leq f_k^*, \forall x \in X_g, \tag{24}$$

which represents the convergence criterion of the DOO-based methods on a grid.

Next, let $\tilde{x} \in X_g$ be the closest point to an arbitrarily chosen global maximum x^*. Then:

$$\|x^* - \tilde{x}\| < \frac{\sqrt{n}\delta}{2}, \tag{25}$$

where $\sqrt{n}\delta$ comes from the diagonal of n-dimensional cubes with side length δ, whose centers are given by the points on the grid X_g. Using Lemma 1 for $x_1 = x^*$ and $x_2 = \tilde{x}$, one can obtain:

$$B_k(x^*) \leq B_k(\tilde{x}) + M \cdot \|x^* - \tilde{x}\|, \tag{26}$$

and by combining $f^* \leq B_k(x^*)$, (25) and (26), it follows that:

$$f^* < B(\tilde{x}) + M\frac{\sqrt{n}\delta}{2}. \tag{27}$$

Next, the inequality from the theorem statement will be proven by contradiction. The contrary of that inequality, along with (27), lead to:

$$f_\infty^* < f^* - M\frac{\sqrt{n}\delta}{2} < B(\tilde{x}), \tag{28}$$

which contradicts the convergence criterion ($f_\infty^* \geq B(x), \forall x \in X_g$). □

Remark 4. The remark in Theorem 3 may seem trivial at first since any exhaustive search algorithm could find the optimum on a grid. However, it ensures that the grid-based methods do not get stuck in a loop without sampling the entire space. Finally, even though the proof relies on the worst-case exhaustive search in which the grid is fully sampled, FTW and FTWD are expected to actually do better in practice, as will be shown later in Section 6.1.

5. Convergence rate of FTW

The goal of this section is to study the convergence *rate* of FTW, i.e. to bound the number of samples a robot needs to acquire to reach a given distance from f^*, or equivalently, to characterize the error reached after a given number of samples. As mentioned in the introduction, the convergence rates proven here are asymptotic in nature, i.e. they hold for large numbers of samples.

Consider optimal sphere packing, or similarly the highest coverage density of point lattices, extensively studied in [38,39]. These problems are relevant to the FTW convergence rate, because the optimal packing can give a bound on the largest amount of samples needed to fill X at a given pairwise distance in order to reach a certain near-optimality. While optimal coverage has been solved in 1D-3D [40], 8D [41] and 24D [42], for other dimensions mostly bounds on the best packing densities were found [43].

The following Lemma is a prerequisite to the convergence rate. Note that the multidimensional (hyper)-sphere will be referred to as a *ball*.

Lemma 2. *Denote by N_σ the maximum number of points taken inside or on the frontier of an n-dimensional ball of radius R such that the distance between any two points is greater than or equal to σ. Then:*

$$N_\sigma < \left(1 + \frac{2R}{\sigma}\right)^n. \tag{29}$$

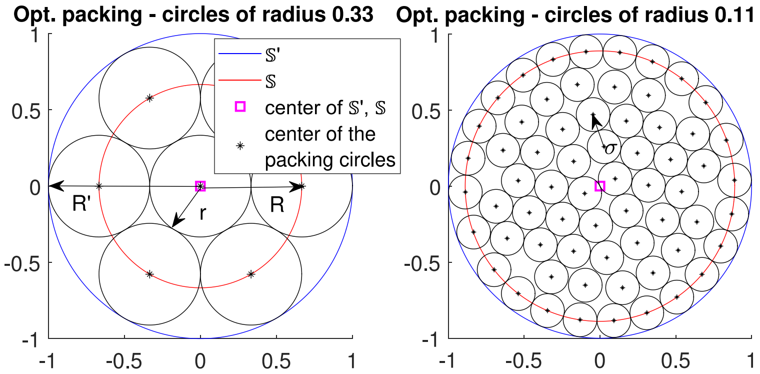

Opt. packing - circles of radius 0.33
Opt. packing - circles of radius 0.11
𝕊'
𝕊
center of 𝕊', 𝕊
center of the packing circles
R'
r
R
σ
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

Fig. 4. Examples of (optimal) packing in 2D with circles of different radius. (For interpretation of the usage of color in this figure, the reader is referred to the web version of this article.)

Proof. N_σ is first placed in a relationship to the ball packing problem. Indeed, to guarantee a minimum distance σ between any two different points, we can take them to be the centers of non-overlapping balls of radius $\frac{\sigma}{2}$. Since the points can be taken also on the frontier of an n-dimensional ball of radius R, they have to be centers of balls packed inside a larger ball of radius $R' = R + \frac{\sigma}{2}$.

Denote by $\mathbb{S}'$ the n-dimensional ball of radius R' and by $N(R', r)$ the maximal number of balls of radius $r \leq R'$ to fit inside $\mathbb{S}'$. Following the reasoning above, $N_\sigma = N(R', \frac{\sigma}{2})$. A simple upper-bound on $N(R', r)$ is obtained by computing the ratio between the Lebesgue measure of the ball of radius R', denoted by $V(R')$, and the Lebesgue measure of the ball of radius r, denoted $V(r)$. Consequently:

$$N(R', r) < \frac{V(R')}{V(r)} = \left(\frac{R'}{r}\right)^n. \tag{30}$$

By replacing $R' = R + r = R + \frac{\sigma}{2}$ in (30) one gets:

$$N_\sigma = N\left(R', \frac{\sigma}{2}\right) < \left(\frac{R + \frac{\sigma}{2}}{\frac{\sigma}{2}}\right)^n = \left(1 + \frac{2R}{\sigma}\right)^n. \quad \square$$

Remark 5. The bound in (29) is by no means tight, and could be tightened by computing N_σ numerically using linear programming techniques such as the ones presented in [43,44]. However, this rough result is enough to derive the complexity measure of FTW later given in Theorem 4.

To give some intuition on Lemma 2, Fig. 4 illustrates two packing examples with balls of different radii. The well-known *honeycomb* obtained by choosing $\sigma = R$ in Lemma 2 is drawn on the left of Fig. 4. The blue and red circles represent the disks (2D balls) $\mathbb{S}'$ and $\mathbb{S}$, and their radii are given as follows: $R' = 1$ is the radius of $\mathbb{S}'$, and $R = 0.66$ of $\mathbb{S}$ (recall that $R' = R + \frac{\sigma}{2}$). The radius of the packing circles is $r = 0.33$. Note that $N_\sigma = 7$, meaning that at most 7 points (circle centers) can be taken in $\mathbb{S}$ such that each of their pairwise distances is at least σ; and the bound on $N_\sigma \leq \left(1 + \frac{2R}{\sigma}\right)^n = 9$ holds true. On the right of Fig. 4, one of the optimal packings with circles of radius for $r = 0.11$ is given, where $R' = 1$, $R = 0.88$ and $\sigma = 0.22$. At most $N_\sigma = 64$ points can be taken in $\mathbb{S}$ such that their pairwise distance does not drop below $\sigma = 0.22$, and thus the bound $N_\sigma \leq \left(1 + \frac{2R}{\sigma}\right)^n = 81$ is verified.

The complexity of FTW will be studied next for a single global optimum $x^* = \arg\max_{x \in X} f(x)$, and then extended to multiple, possibly infinitely many global maxima. Based on the convergence proof of FTW in Theorem 2, $\exists k_\varepsilon$ such that $\forall k > k_\varepsilon$, $\exists \mathbb{B}^*(x_c, R)$ a ball centered in $x_c \in X$, of radius R and containing x^*, such that $\max_{x \in \mathbb{B}^*} B_k(x) \geq f^* > \max_{y \in X \setminus \mathbb{B}^*} B_k(y)$. In other words, while converging to f^* (and implicitly to x^*), the robot will eventually sample only inside a neighborhood of the maximum, contained inside the ball $\mathbb{B}^*(x_c, R)$. In the convergence rate of Theorem 4, it will be important that every point within this ball can be reached in a single step.

Define next the loss criterion $r_k = f^* - f_k^*$, which measures the closeness to optimality reached by FTW after k samples. Obviously, r_k is a monotonously decreasing sequence and $r_k \to 0$ as $f_k^* \to f^*$, per the convergence Theorem 2.

Theorem 4 (*Worst-case Convergence Rate of FTW and Committed DOO for a Single Global Optimum*)**.** *Let $\varepsilon > 0$. After taking at most $\left(1 + \frac{2MR}{\varepsilon}\right)^n$ samples inside $\mathbb{B}^*(x_c, R)$, the loss will be bounded by $r_k < \varepsilon$, and $\exists x_s \in S_k$ such that $\|x_s - x^*\| < \frac{\varepsilon}{M}$.*

Proof. Consider $k = \left(1 + \frac{2MR}{\varepsilon}\right)^n$ and define $\Delta_k := \max_{x \in X} B_k(x) - f_k^*$. Suppose that $\Delta_k \geq \varepsilon$. For convenience, the indexing starts with $k = 1$ as the samples acquired outside the ball $\mathbb{B}^*$ can be omitted for this proof. Note that a contradiction to $\Delta_k \geq \varepsilon$ would prove the bound on the loss, since:

$$r_k = f^* - f_k^* \leq B_k(x^*) - f_k^* \leq \Delta_k < \varepsilon. \tag{31}$$

Since by Assumption 2 all states inside $\mathbb{B}^*$ are reachable from each other within one robot step, the next max-B state targeted by the robot will become an actual sample at the next FTW iteration, i.e. $x_{k+1} = \arg\max_{x\in X} B_k(x)$. Denote by $\tilde{x}$ the closest sample to x_{k+1} taken inside $\mathbb{B}^*$ up to step k. The following inequalities hold true:

$$\Delta_k = \max_{x\in X} B_k(x) - f_k^* \le f(\tilde{x}) + M\|\tilde{x} - x_{k+1}\| - f_k^* \le M\|\tilde{x} - x_{k+1}\|, \tag{32}$$

because $f(\tilde{x}) \le f_k^*$. Using $\Delta_k \ge \varepsilon$, (32) leads to:

$$\|\tilde{x} - x_{k+1}\| \ge \frac{\varepsilon}{M}. \tag{33}$$

The result in (33) implies that the pairwise distance of the samples acquired up to step $k+1$ does not decrease below $\frac{\varepsilon}{M}$. This statement represents a contradiction, as according to Lemma 2, where $\sigma := \frac{\varepsilon}{M}$, the maximum number of samples located at least $\frac{\varepsilon}{M}$ farther away from each other is less than $\left(1 + \frac{2MR}{\varepsilon}\right)^n$. This contradiction ends the first part of the proof, showing that $r_k < \varepsilon$.

To study how close the samples in $\mathbb{B}^*$ get to the optimum x^*, consider the following inequality:

$$\varepsilon > \Delta_k = \max_{x\in X} B_k(x) - f_k^* > B_k(x^*) - f_k^*. \tag{34}$$

Note that $\exists x_s \in S_k$, a sample which gives the bound in x^*:

$$B_k(x^*) = f(x_s) + M \cdot \|x_s - x^*\|. \tag{35}$$

Using (34) and (35), it follows that:

$$\varepsilon > f(x_s) + M \cdot \|x_s - x^*\| - f_k^* \ge M \cdot \|x_s - x^*\|, \tag{36}$$

and then:

$$\|x_s - x^*\| < \frac{\varepsilon}{M}. \tag{37}$$

Lastly, it will be proven that for a large enough number of steps $k > k_\varepsilon$ so that at least one sample is acquired inside $\mathbb{B}^*$, $\exists C \in (0, (4MR)^n]$ such that:

$$\left(1 + \frac{2MR}{\varepsilon}\right)^n < C\varepsilon^{-n} = O(\varepsilon^{-n}). \tag{38}$$

By multiplying the inequality in (38) with ε^n one can obtain:

$$(\varepsilon + 2MR)^n < C. \tag{39}$$

A bound on ε once $\exists x_s \in \mathbb{B}^*$ is:

$$\varepsilon = \max_k \Delta_k = \max_k(\max_{x\in X} B_k(x) - f_k^*) \le M \max_{x\in\mathbb{B}^*} \|x - x_s\| = 2MR, \tag{40}$$

as the maximum distance between any max-B state located inside $\mathbb{B}^*$ and x_s is at most equal to the diameter of the ball, i.e. $2R$. Using (39) and (40):

$$C \in (0, (4MR)^n]. \tag{41}$$

Thus, the $O(\varepsilon^{-n})$ worst-case number of steps/samples for FTW and committed DOO has been obtained. □

The interpretation of Theorem 4 is that FTW and committed DOO need at worst $O(\varepsilon^{-n})$ samples to get $\frac{\varepsilon}{M}$-close to the unique optimum x^*. Note that in practice the algorithm will sample fewer points. To characterize this, a complexity measure of FTW will be given next.

Definition 1. Define the complexity measure of the optimization problem to be the smallest constant $m \in [0, n]$ for which there exists $C > 0$, such that for any $\varepsilon > 0$ the maximum number of samples taken inside $\mathbb{B}^*$ for FTW to reach a loss $r_k < \varepsilon$ is less than $C\varepsilon^{-m}$.

The existence of C and m are verified by Theorem 4 and by the continuity of $C\varepsilon^{-\gamma}$ in γ. Then, by definition, the number of samples to get a loss $r_k < \varepsilon$ in the case of a single x^* is $O(\varepsilon^{-m})$. Next, this result will be generalized to any number of optima.

Theorem 5 (*Convergence Rate of FTW and Committed DOO in the General Case*). *Let $\varepsilon > 0$. The number of samples to reach $\frac{\varepsilon}{M}$-close to all optima of f in X is $O(\varepsilon^{-m})$.*

Proof. To bound the number of steps required to reach $\frac{\varepsilon}{M}$-close to multiple, possibly infinite number of maxima, the near-optimal regions in X can be packed with balls similar to $\mathbb{B}^*$ from above. Again, as FTW finds all global maxima, and following Assumption 2, $\exists k_\varepsilon$ such that $\forall k > k_\varepsilon$, $\exists p$ balls $\mathbb{B}_i^*(x_{c,i}, R_i)$ of radius R_i, containing global maxima, such that $\max_{x\in\cup_{i=1}^p \mathbb{B}_i^*} B_k(x) \ge f^* > \max_{y\in X\setminus\cup_{i=1}^p \mathbb{B}_i^*} B_k(y)$. Note that the number of the balls $\mathbb{B}_i^*$ is finite (as X is compact) and they are not necessarily disjoint. Denote by N_{max} the maximum number of steps required by the robot to travel between any two balls. In the worst-case scenario the robot will sample inside one ball and then immediately move to the next max B-state located in another ball. This will lead to a computational complexity of $O(pN_{max}\varepsilon^{-m}) = O(\varepsilon^{-m})$, with the same rate as in the case of a single x^*.

To cover also the transient regime, the number of trajectory samples until all new target points will be chosen inside $\cup_{i=1}^{p}\mathbb{B}_i^*$ is finite (recall the existence of k_ε in the above proof). Thus, $\exists\overline{C} \geq C$ such that the maximum number of samples across the whole trajectory in order to reach a loss bound $r_k < \varepsilon$ is at most $\overline{C}\varepsilon^{-m}$, i.e. the same complexity $O(\varepsilon^{-m})$. □

The complexity measure m of FTW is closely related to the near-optimality dimension d in [12], which was used to characterize the original DOO. Both measures are related to the maximum number of disjoint, metric balls used to cover a set (ball) of near-optimal states. In what follows, a relation between m and d will be sought.

For a given k, consider the smallest $\varepsilon > 0$ and $\varepsilon' > 0$ such that the loss $r_k = f^* - f_k^* < \varepsilon$ and the suboptimality $r_k' := f^* - f(x_k) < \varepsilon'$, where x_k is the state recommended to be sampled by FTW at step k. Define next the set of ε-optimal states:

$$X_\varepsilon := \{x \in X \mid f(x) > f^* - \varepsilon\}. \tag{42}$$

Since $r_k \leq r_k'$, it follows that $\varepsilon \leq \varepsilon'$ and $X_\varepsilon \subseteq X_{\varepsilon'}$.

From the definition of the near-optimality dimension in [12], there exists $d \geq 0$ and $C' > 0$, such that the total number of disjoint balls of radius $\frac{\varepsilon'}{2M}$ used to cover $X_{\varepsilon'}$ and reach a suboptimality of $r_k' < \varepsilon'$ is at most $C'\varepsilon'^{-d}$. Note that in this case, the measure is called the $\frac{1}{2M}$-near optimality dimension and the constant which ensures that the near-optimal regions are well-shaped in [12] is $\nu = 1$. Using Definition 1, there exists $m \geq 0$ and $C > 0$ such that the number of disjoint balls of the same radius $\frac{\varepsilon'}{2M}$ used to cover X_ε and to reach a loss $r_k < \varepsilon \leq \varepsilon'$ is less than $C\varepsilon'^{-m}$. As $X_\varepsilon \subseteq X_{\varepsilon'}$, fewer balls of the same radius are required to cover the set of ε-optimal states, i.e.:

$$C\varepsilon'^{-m} \leq C'\varepsilon'^{-d}, \tag{43}$$

an inequality that holds for any $\varepsilon' > 0$. The result in (43) can be rewritten as:

$$m \leq d + \frac{\log(C'/C)}{\log(\varepsilon'^{-1})}, \forall \varepsilon' > 0 \tag{44}$$

and by taking the limit $\varepsilon' \to 0$, it finally follows that $m \leq d$. The complexity measure of FTW is therefore at most the near-optimality dimension in DOO.

6. Numerical results

This section numerically studies the FTW and FTWD methods, compares them to path-aware and source seeking baselines, and provides insights into the convergence rate and complexity of FTW. The following experiments will be performed using a simulated robot with the motion dynamics (4) defined as a unicycle:

$$x_{k+1} = x_k + T_s \cdot u_{k,1} \cdot [\cos(u_{k,2}), \sin(u_{k,2})]^T, \tag{45}$$

where state $x_k = [x_{k,1}, x_{k,2}]^T$, sampling period $T_s = 1\,\mathrm{s}$ and control action $u_k = [u_{k,1}, u_{k,2}]^T$. The robot velocity will be taken as $u_{k,1} \in [0, 0.2]\,\mathrm{m/s}$ and the heading $u_{k,2} \in [0, 2\pi)$. To find the maxima, the robot will search inside the space $X = [0,4]\,\mathrm{m} \times [0,4]\,\mathrm{m}$. The bound B will be evaluated over a discretized grid with 41 × 41 points equally spaced by $\delta = 0.1\,\mathrm{m}$ across both dimensions of X.

All the methods in the experiments will use the objective function:

$$f(x) = \max\{\phi_{1,2,3}(x), \psi_{1,2,3}(x)\}, \text{ where } \phi_i(x) = h_i \cdot \exp\left[-\sum_{j=1}^{2}\frac{(x_j - c_{ij})^2}{b_{ij}^2}\right] \text{ and } \psi_i(x) = \lambda_i \cdot [f^* - M'\|x - c_i'\|], \tag{46}$$

i.e. f is the maximum amplitude among 3 cone functions ψ and 3 radial basis functions ϕ (RBFs). The parameters of ψ are as follows: slope coefficients $\lambda_i = [1; 2/3; 1/2]$ and centers $c_i' = [3.25; 1.5], [1; 0.75], [1.5; 0.5]$, where $M' = 312.5$; and of ϕ: widths $b_i = [1.4; 1.4]\lambda_i$, heights $h_i = 255\lambda_i$ and centers: $c_i = [2.75; 3.5], [0.75; 2.5], [3.75; 1.75]$. Fig. 5 includes a contour representation of f.

The Lipschitz constant is set to the maximal absolute value of the derivative of f among all points where it exists, i.e. $M = M' = 312.5$, and the global maximum $f^* = 255$ is attained at two different locations: $x_1^* = [2.75; 3.5]$ and $x_2^* = [3.25; 1.5]$. Moreover, the slope of the cone centered in x_2^* equals the Lipschitz constant and the cone functions are nondifferentiable at their peaks, thus f is overall nondifferentiable. The generality of the function shape is important because as stated in the introduction, most extremum and source seeking techniques assume a single global optimum and global differentiability for f, while FTW(D) methods accommodate multiple global optima, relying on the weaker property of Lipschitz continuity instead. The performance of the proposed methods on the previously described f will be highlighted in the following experiments, particularly in the baseline comparison within Section 6.2.

The first two methods used in the comparison to FTW(D) are from path-aware optimization: committed DOO and Path-Aware Optimistic Optimization (OOPA) [13], and were previously developed by the authors of the present paper. For implementation details and parameter tuning, please refer to [13]. To allow for a fairer comparison, and in order to partly offset the higher computation times of OOPA, its grid resolution will be increased to $\delta = 0.2\,\mathrm{m}$. The execution times of FTW(D) and OOPA will be later given in Section 6.2.

FTW(D) will also be compared against two representative, state-of-the-art source seeking baselines detailed in [14,15]. The method in [14] replaces the often-used gradient with a local spatial average of the objective function computed by taking the mean of f over a circle centered in the robot's center, with its radius determined by the distance r of a forward-mounted sensor. Assuming certain conditions hold (e.g., local extrema do not occur within the robot's averaging radius), the robot is directed away from local extrema towards the global maximum. The tuning of the method remains consistent with [14] (in particular, $r = 0.2\,\mathrm{m}$), while adopting the objective function from (46) and the search area X defined earlier. For clarity and ease of reference, we will refer to this method as the *rotating-robot baseline*, as it implies a rotational movement of the robot's sensor.

The second baseline [15] applies a sliding mode control strategy to steer the robot to the location of an optimum, and thus will be referred to as the *sliding-mode baseline*. The motion dynamics in [15] are of a single-integrator type similar to (4). They involve a constant translational velocity $u_{k,1}$, and an angular velocity given by $\dot{u}_{k,2} = \bar{u}_{k,2} sgn\{\dot{f}(x_k) - v_*\}$, where $|\dot{u}_{k,2}| \leq \bar{u}, \forall k$, and v_* is a controller parameter influencing the size of the optimum's neighborhood in which the robot should converge. The parameters for this baseline were tuned as follows: $u_{k,1} = 0.2\,\mathrm{m/s}$ (same as the maximum velocity of FTW(D)), $\bar{u} = 1.25$ rad/s and $v_* = 5$. Note that the method in [15] assumes the only critical point is given by a single (global) optimum, which is not the case for f. To address this issue, the method is modified such that once the robot converges to the required neighborhood of a maximum, it travels to a randomly chosen location in X before resuming its sliding-mode search strategy. This is done to increase the chance of finding global optima in X. Note that the rotating-robot and sliding-mode baselines were chosen because, after the above modification, they perform a version of global search.

The experiments are reported in the following order. Section 6.1 compares FTW to FTWD, while Section 6.2 gives a comparison of the two with the committed DOO, OOPA and [14,15] baselines. Section 6.3 experimentally studies the convergence of FTW across its whole trajectory, as well as inside a ball containing an x^* in order to give some intuition on the value of m.

6.1. Comparison between FTW and FTWD

Recall first the issue of the oscillating trajectory of FTW, previously presented at the end of Section 3. An example of such a trajectory was given on the left of Fig. 2, where the robot repeatedly moves between multiple maxima that are close in amplitude (the ones marked with the red and orange squares), instead of refining around one maximum and then moving on to the next one. Note that the initial robot position was arbitrarily chosen as $x_0 = [0.74; 1.96]$ for that experiment.

With the improved algorithm FTWD, the robot spends fewer steps oscillating from one maximum to another when started from the same x_0. The plot on the right of Fig. 2 shows that compared to FTW, FTWD requires roughly 40.66% less distance till convergence.[1] Note that tuning parameters remained unchanged between FTW and FTWD.

It is instructive to check the performance of the two methods from more than one initial position. For this, 50 randomly chosen points in X are taken as initial positions of the robot. Fig. 5 shows the trajectory length until reaching convergence for the first 25 runs of the two methods (the number of results plotted is reduced to keep the readability of the figure). FTWD always converges sooner, scoring on average, for all 50 runs, 35.16% less distance compared to FTW.

6.2. Comparison to baselines

The FTW and FTWD methods will be studied next in a baseline comparison against committed DOO, OOPA, as well as the rotating-robot and sliding-mode baselines. The evolution of the best value seen so far, f_k^*, will be reported for all methods, on average over 50 runs, along the first 250 trajectory steps. The initial positions of the robot are chosen as in the experiment of Section 6.1. Fig. 6 shows that FTWD drastically improves the performance of FTW, getting close to the performance of OOPA. This is important since gathering better samples faster translates to earlier convergence to f^* and implicitly to x^*. Recall that OOPA has no convergence guarantees. Due to the finer grid used, FTW(D) reached closer to f^* compared to OOPA, without sacrificing computational time: a step of FTW or FTWD took on average two orders of magnitude less than OOPA (0.01 s vs. 1.85 s). The performance of committed DOO was between FTW and the rotating-robot baseline. While all path-aware methods found both global maxima, this was achieved in only 24% of all runs by the sliding-mode baseline and never by the rotating-robot baseline. The latter method reached one global maximum in 50% of all runs, while the sliding-mode baseline found at least one global maximum in 80% of the runs.

6.3. Convergence rate estimation for FTW

The rate at which the inclusion of the balls in (23), denoted by $\cup\mathbb{B}$, fills the search space will be first studied. Because optima are found in the region outside the covered area, the evolution of the area of $\cup\mathbb{B}$ gives an idea of the convergence rate of the algorithm, especially during its transient regime. For a quantitative study, consider the same 50 initial positions of the robot from the previous sections. The left plot in Fig. 7 reports the average of the area covered at each step for all runs until convergence. Results show a logarithmic increase of the mean area covered by $\cup\mathbb{B}$, approaching the total area of X (16 m^2), which serves as a notable indicator of convergence.

[1] The distance is reported in this work instead of the number of steps until reaching convergence since some of the methods involved have an adaptive step length.

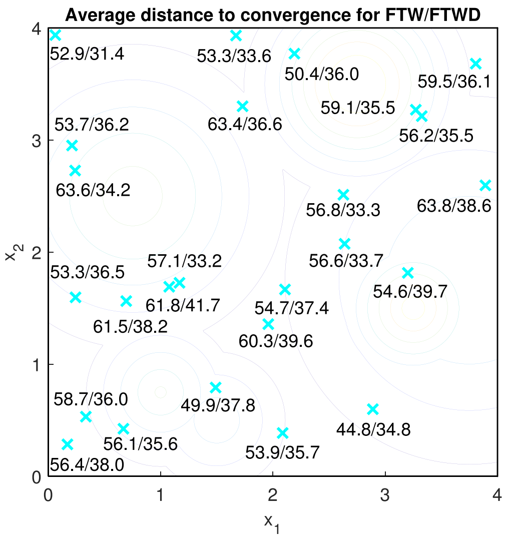


Fig. 5. Results (convergence distance) of 25 runs of FTW and FTWD are separated by a slash and placed near the starting robot positions, drawn with cyan 'x's. (For interpretation of the usage of color in this figure, the reader is referred to the web version of this article.)

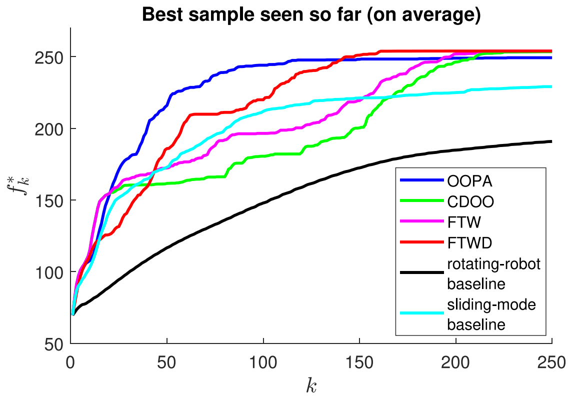


Fig. 6. Comparison between OOPA, committed DOO (CDOO), FTW(D), the rotating-robot and sliding-mode baselines. The experimental results are reported as the best sample seen so far for the first 250 trajectory samples (on average). (For interpretation of the usage of color in this figure, the reader is referred to the web version of this article.)

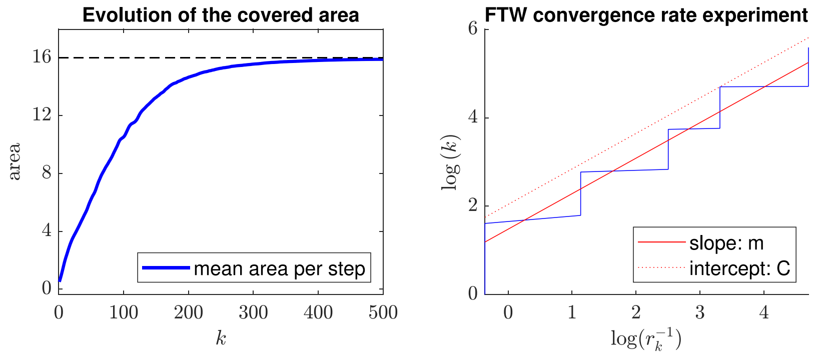


Fig. 7. Left: Result of the convergence rate experiment in terms of area covered by the union of the balls (disks). Recall that this area is an exclusion zone in which no global maximum can be located. Right: Approximating C and m in an FTW convergence rate experiment. (For interpretation of the usage of color in this figure, the reader is referred to the web version of this article.)

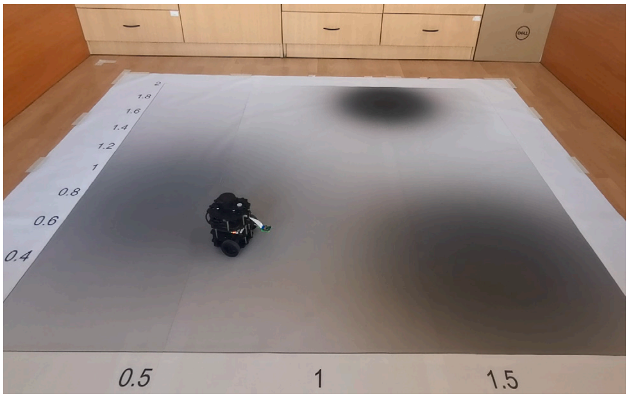


Fig. 8. TurtleBot3 experimental testbed.

Next, the FTW complexity measure is approximated in a numerical experiment. Consider the ball $\mathbb{B}^*(x_c, R)$, where $x_c := x_1^*$ and $R = \frac{T_s \cdot \max\{u_{k,1}\}}{2} = 0.2\,\text{m}$. For a better accuracy while searching for f^*, a different, finer grid of 151 points spaced equally across each dimension in X is considered. The robot is initialized in an arbitrarily chosen position $x_0 = [2.71; 3.46]$.

According to Definition 1, the number of samples k taken inside B^* to reach $r_k < \varepsilon$ is such that $k < C\varepsilon^{-m} < Cr_k^{-m}$. This inequality leads to:

$$\log(k) < \log(C) + m\log(r_k^{-1}). \tag{47}$$

Thus, by fitting the relationship between $\log(r_k^{-1})$ and $\log(k)$ with a line, its intercept helps in approximating the constant C, while the slope approximates the FTW complexity measure m. Note that by studying $\log(k)$ versus $\log(r_k^{-1})$ in a single experiment, only a lower bound m_0 on m can be found, since the constants C and m in Definition 1 ensure that $k < C\varepsilon^{-m}$ for *any* initial position of the robot. Indeed, for experiments initialized in different starting positions, the value of the slope might be higher.

The right plot in Fig. 7 shows the results. The slope of the resulting line can be computed by linear regression, leading to $m_0 \approx 0.8$. An intercept to satisfy $k < C_0 r_k^{-m}$ can be chosen as $\log(C_0) > 2$, i.e. $C_0 > e^2$. Therefore, $m \geq 0.8$.

7. Real-robot experiments with a TurtleBot3 platform

In this section, real-robot experiments will be performed to verify the simulated comparison results between FTW and FTWD previously presented in Fig. 2 of Section 3 and Fig. 5 of Section 6.1. The experiments will be conducted using the ROBOTIS TurtleBot3 platform, and the goal is for the robot to reach the darkest location of a 2D printed grayscale map.

The experimental setup, whose overview is given in Fig. 8, is similar to the setups used in [2,13]. Different from those implementations, the AMCL package was deployed here to increase the localization accuracy of the robot, using LiDAR information from the surrounding walls. The AMCL-based approach has significantly enhanced localization accuracy, reducing the mean error to approximately 5 cm. This represents a notable improvement over the dead reckoning method utilized in [2], which incurred average localization errors exceeding 10 cm.

For the following experiments, the number of trajectory samples is set to $N = 250$ (leading to 50 m travel distance), and the Lipschitz constant is tuned empirically to $M = 300$.

Figs. 9 and 10 present the results of two TurtleBot3 experiments, one performed with FTW and another with FTWD. The oscillating behavior of the FTW trajectory, previously seen in the simulation of Fig. 2, Section 3, can also be observed on the left plot of Fig. 9. Similar to the experiment of Fig. 2 the plot on the right of Fig. 9 shows that FTWD greatly reduces the oscillating behavior, reaching convergence in 26% shorter path than FTW (FTW: 37.5 m vs. FTWD: 27.8 m). In Fig. 10, the magenta and red lines respectively give for each of the two algorithms the best sample acquired up to step k, f_k^*. The dotted lines drawn with the same colors point out the step at which convergence was reached for each method: 190 steps for FTW and 146 steps for FTWD. For clarity, recall the convergence criterion shared by the two methods: $f_k^* \geq B_k(x), \forall x \in X_g$.

In both experiments, the global maximum was found with good accuracy: FTW reached as close as 7 cm to x^*: $\widehat{x^*} = [1.44; 1.76]$ ($\widehat{f^*} = 175.7$), while FTWD approximated x^* with an error of 4 cm: $\widehat{x^*} = [1.41; 1.76]$ ($\widehat{f^*} = 180.4$). Such errors are expected due to localization inaccuracies and hardware/setup limitations. In particular, the rather poor quality of the camera and of the printed plot flattened the real function to the smaller $[80; 180]$ interval, different from the $[0; 255]$ interval from the simulations. That is why the curves stop around 180 on the y-axis in Fig. 10.

Videos corresponding to the TurtleBot3 experiments in Figs. 9 and 10 and to the video still in Fig. 8 are available online at http://rocon.utcluj.ro/files/nahs_ftw_tb3.mp4 and http://rocon.utcluj.ro/files/nahs_ftwd_tb3.mp4, for FTW and FTWD respectively. To shorten the length of the videos such that they fit in 1.5 min or less, each video was accelerated by a factor of approximately 20.

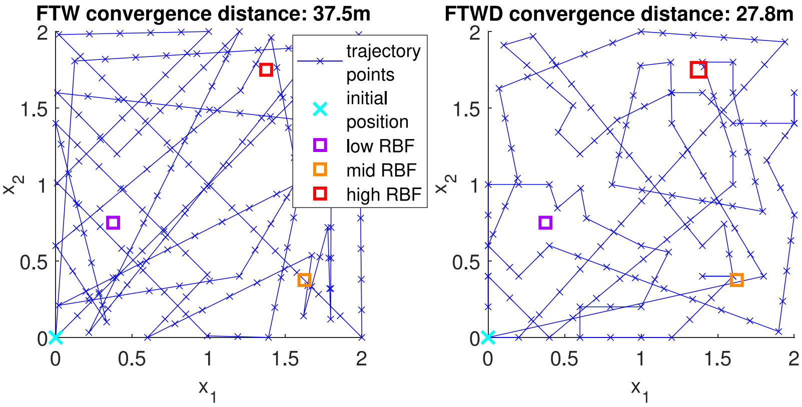


Fig. 9. Left: The oscillating behavior of the FTW trajectory from simulations (Fig. 2) is also present in the TurtleBot3 experiment. Right: The oscillating behavior is reduced for FTWD, like in the simulations. (For interpretation of the usage of color in this figure, the reader is referred to the web version of this article.)

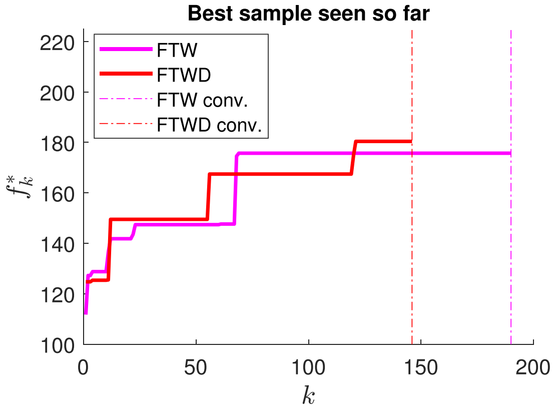


Fig. 10. Experimental results of the TurtleBot3 applying the FTW and FTWD methods. (For interpretation of the usage of color in this figure, the reader is referred to the web version of this article.)

8. Conclusions

This work presented two techniques called FTW and FTWD for mobile robots to find the maximum of a physical quantity defined over their operating area. Convergence guarantees and rates were analytically provided for FTW, whereas the convergence of FTWD was proven for a discrete search space (grid of points). An FTW complexity measure was identified and a relation was found with the already-established near-optimality dimension of DOO. FTWD greatly improved the practical performance of FTW, reaching close to the performance of a much more computationally intensive method that does not have convergence guarantees. In the same baseline study, FTWD outperformed two representative methods from source seeking control by always reaching close to all global maxima. Finally, TurtleBot3 experiments validated the simulation results. Future work includes extensions such as obstacle avoidance and the development of multiagent versions of the techniques.

CRediT authorship contribution statement

T. Sântejudean: Conceptualization, Formal analysis, Investigation, Methodology, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Ş. Ungur:** Software, Validation, Visualization. **R. Herzal:** Software, Validation, Visualization. **I.-C. Morărescu:** Conceptualization, Formal analysis, Methodology, Supervision. **V.S. Varma:** Conceptualization, Formal analysis, Investigation, Methodology, Supervision. **L. Buşoniu:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Yongguo Mei, Yung-Hsiang Lu, Yu Charlie Hu, CS George Lee, Deployment of mobile robots with energy and timing constraints, IEEE Trans. Robot. 22 (3) (2006) 507–522.

[2] Tudor Sântejudean, Lucian Buşoniu, Vineeth Varma, Constantin Morarescu, A simple path-aware optimization method for mobile robots, IFAC-PapersOnLine 55 (8) (2022) 1–6.

[3] Mingrui Jiang, Yu Liao, Xun Guo, Hao Cai, Wenqing Jiang, Zhou Yang, Fei Li, Fei Liu, A comparative experimental study of two multi-robot olfaction methods: towards locating time-varying indoor pollutant sources, Build. Environ. 207 (2022) 108560.

[4] Joseph Bourne, Eric Pardyjak, Kam Leang, Coordinated Bayesian-based bioinspired plume source term estimation and source seeking for mobile robots, IEEE Trans. Robot. 35 (4) (2019) 967–986.

[5] Lingxiao Wang, Shuo Pang, Robotic odor source localization via adaptive bio-inspired navigation using fuzzy inference methods, Robot. Auton. Syst. 147 (2022) 103914.

[6] Johann Borenstein, H.R. Everett, Liqiang Feng, et al., Where am I? Sensors and Methods for Mobile Robot Positioning, Vol. 119, University of Michigan, 1996, p. 27, (120).

[7] Maxim Batalin, Gaurav Sukhatme, Myron Hattig, Mobile robot navigation using a sensor network, in: IEEE International Conference on Robotics and Automation, 2004. Proceedings, ICRA'04. 2004, Vol. 1, IEEE, 2004, pp. 636–641.

[8] Michael Fulton, Jungseok Hong, Md Jahidul Islam, Junaed Sattar, Robotic detection of marine litter using deep visual detection models, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 5752–5758.

[9] Matthias Rosynski, Lucian Buşoniu, A simulator and first reinforcement learning results for underwater mapping, Sensors 22 (14) (2022) 5384.

[10] Marc Gouttefarde, Mariola Rodriguez, Cyril Barrelet, Pierre-Elie Hervé, Vincent Creuze, Jose Gorrotxategi, Arkaitz Oyarzabal, David Culla, Damien Sallé, Olivier Tempier, et al., The robotic seabed cleaning platform: An underwater cable-driven parallel robot for marine litter removal, in: International Conference on Cable-Driven Parallel Robots, Springer, 2023, pp. 430–441.

[11] Rémi Munos, Optimistic optimization of a deterministic function without the knowledge of its smoothness, Adv. Neural Inf. Process. Syst. 24 (2011).

[12] Rémi Munos, From bandits to Monte-Carlo tree search: The optimistic principle applied to optimization and planning, Found. Trends Mach. Learn. 7 (1) (2014) 1–129.

[13] Tudor Sântejudean, Lucian Buşoniu, Online learning control for path-aware global optimization with nonlinear mobile robots, Control Eng. Pract. 126 (2022) 105228.

[14] Raik Suttner, Miroslav Krstic, Nonlocal nonholonomic source seeking despite local extrema, 2023, arXiv preprint arXiv:2303.16027.

[15] Alexey Matveev, Hamid Teimoori, Andrey Savkin, Navigation of a unicycle-like mobile robot for environmental extremum seeking, Automatica 47 (1) (2011) 85–91.

[16] Ying Tan, Dragan Nešić, Iven M.Y. Mareels, Alessandro Astolfi, On global extremum seeking in the presence of local extrema, Automatica 45 (1) (2009) 245–251.

[17] Lina Fu, Ümit Özgüner, Extremum seeking with sliding mode gradient estimation and asymptotic regulation for a class of nonlinear systems, Automatica 47 (12) (2011) 2595–2603.

[18] Sei Zhen Khong, Ying Tan, Chris Manzie, Dragan Nešić, Multi-agent source seeking via discrete-time extremum seeking control, Automatica 50 (9) (2014) 2312–2320.

[19] Marcus Gronemeyer, Mirco Alpen, Joachim Horn, Limited gradient criterion for global source seeking with mobile robots, IFAC-PapersOnLine 53 (2) (2020) 15288–15293.

[20] Mohammadali Ghadiri-Modarres, Mohsen Mojiri, Normalized extremum seeking and its application to nonholonomic source localization, IEEE Trans. Autom. Control 66 (5) (2020) 2281–2288.

[21] Shu-Jun Liu, Miroslav Krstic, Stochastic source seeking for nonholonomic unicycle, Automatica 46 (9) (2010) 1443–1453.

[22] Andrew R. Teel, Dobrivoje Popovic, Solving smooth and nonsmooth multivariable extremum seeking problems by the methods of nonlinear programming, in: Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148), Vol. 3, IEEE, 2001, pp. 2394–2399.

[23] Lina Fu, Umit Ozguner, Variable structure extremum seeking control based on sliding mode gradient estimation for a class of nonlinear systems, in: 2009 American Control Conference, IEEE, 2009, pp. 8–13.

[24] Shun-ichi Azuma, Mahmut Selman Sakar, George J. Pappas, Stochastic source seeking by mobile robots, IEEE Trans. Autom. Control 57 (9) (2012) 2308–2321.

[25] Jennie Cochran, Miroslav Krstic, Nonholonomic source seeking with tuning of angular velocity, IEEE Trans. Autom. Control 54 (4) (2009) 717–731.

[26] Howie Choset, Coverage for robotics - a survey of recent results, Ann. Math. Artif. Intell. 31 (2001) 113–126.

[27] Chomchana Trevai, Jun Ota, Tamio Arai, Multiple mobile robot surveillance in unknown environments, Adv. Robot. 21 (7) (2007) 729–749.

[28] Georgios Fevgas, Thomas Lagkas, Vasileios Argyriou, Panagiotis Sarigiannidis, Coverage path planning methods focusing on energy efficient and cooperative strategies for unmanned aerial vehicles, Sensors 22 (3) (2022) 1235.

[29] Lukas Schmid, Michael Pantic, Raghav Khanna, Lionel Ott, Roland Siegwart, Juan Nieto, An efficient sampling-based method for online informative path planning in unknown environments, IEEE Robot. Autom. Lett. 5 (2) (2020) 1500–1507.

[30] Marija Popović, Teresa Vidal-Calleja, Gregory Hitz, Jen Jen Chung, Inkyu Sa, Roland Siegwart, Juan Nieto, An informative path planning framework for UAV-based terrain monitoring, Auton. Robots 44 (2020) 889–911.

[31] Jakob Engel, Thomas Schöps, Daniel Cremers, LSD-SLAM: Large-scale direct monocular SLAM, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13, Springer, 2014, pp. 834–849.

[32] Andréa Macario Barros, Maugan Michel, Yoann Moline, Gwenolé Corre, Frédérick Carrel, A comprehensive survey of visual SLAM algorithms, Robotics 11 (1) (2022) 24.

[33] Rui Zou, Vijay Kalivarapu, Eliot Winer, James Oliver, Sourabh Bhattacharya, Particle swarm optimization-based source seeking, IEEE Trans. Autom. Sci. Eng. 12 (3) (2015) 865–875.

[34] Shu-Jun Liu, Miroslav Krstic, Stochastic averaging in continuous time and its applications to extremum seeking, IEEE Trans. Autom. Control 55 (10) (2010) 2235–2250.

[35] Abdollah Amirkhani, Amir Hossein Barshooi, Consensus in multi-agent systems: a review, Artif. Intell. Rev. 55 (5) (2022) 3897–3935.

[36] Reza Olfati-Saber, Richard M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Trans. Autom. Control 49 (9) (2004) 1520–1533.

[37] Shiyu Zhao, Zhiyong Sun, Defend the practicality of single-integrator models in multi-robot coordination control, in: 2017 13th IEEE International Conference on Control & Automation, ICCA, IEEE, 2017, pp. 666–671.

[38] John Horton Conway, Neil James Alexander Sloane, Sphere Packings, Lattices and Groups, Vol. 290, Springer Science & Business Media, 2013.

[39] John Leech, Neil James Alexander Sloane, Sphere packing and error-correcting codes, in: Sphere Packings, Lattices and Groups, Springer, 1999, pp. 136–156.

[40] Thomas Hales, Sphere packings, I, in: The Kepler Conjecture, Springer, 2011, pp. 379–431.

[41] Maryna Viazovska, The sphere packing problem in dimension 8, Ann. of Math. (2017) 991–1015.

[42] Henry Cohn, Abhinav Kumar, Stephen Miller, Danylo Radchenko, Maryna Viazovska, The sphere packing problem in dimension 24, Ann. of Math. 185 (3) (2017) 1017–1033.

[43] Yuriy Stoyan, Georgiy Yaskov, Packing congruent hyperspheres into a hypersphere, J. Global Optim. 52 (4) (2012) 855–868.

[44] Nima Afkhami-Jeddi, Henry Cohn, Thomas Hartman, David de Laat, Amirhossein Tajdini, High-dimensional sphere packing and the modular bootstrap, J. High Energy Phys. 2020 (12) (2020) 1–45.