Robotics and Autonomous Systems 61 (2013) 920–931

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

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

# A novel efficient algorithm for mobile robot localization

Mohsen Mirkhani [a], Rana Forsati [b,*], Alireza Mohammad Shahri [c], Alireza Moayedikia [d]

[a] Department of Electrical and Computer Engineering, Islamic Azad University, Qazvin Branch, Qazvin, Iran
[b] Department of Computer Engineering, Islamic Azad University, Karaj Branch, Karaj, Iran
[c] Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran
[d] Department of Computing, Asia Pacific University, Kuala Lumpur, Malaysia

## HIGHLIGHTS

- Coverage of some previous works.
- Localization is viewed as an optimization problem and is solved by a harmony search.
- Proposing a hybrid method and proposing 2 methods in total.

## ARTICLE INFO

*Article history:*
Received 20 April 2012
Received in revised form
25 March 2013
Accepted 5 April 2013
Available online 30 April 2013

*Keywords:*
Mobile robot
Localization
Harmony search
Differential evolution
Scans matching
Hybrid algorithms

## ABSTRACT

Being autonomous is one of the most important goals in mobile robots. One of the fundamental works to achieve this goal is giving the ability to a robot for finding its own correct position and orientation. Different methods have been introduced to solve this problem. In this paper, a novel method based on the harmony search (HS) algorithm for robot localization through scan matching is proposed. Simulation results show that the proposed method in comparison with a genetic algorithm-based approach has better accuracy and higher performance. Furthermore a new hybrid algorithm based on harmony search and differential evolution (DE) algorithms is proposed and evaluated on different benchmark functions. Finally the hybrid algorithm has been applied for mobile robot localization and it outperformed the HS-based approach.



## 1. Introduction

Self-localization of mobile robots is the process of obtaining the accurate position and orientation of the robot relative to an absolute world frame. In recent years laser scanners have decreased in cost, while their performance has increased. Furthermore, laser scanners are capable of providing many range measurements quickly and precisely. These features made laser scanners to become very popular in mobile robotics.

Scan matching is one of the most efficient ways for robot localization. In scan matching two different scans, one 2D reference scan gathered with a range sensing device by the robot located in a given pose and another 2D scan gathered by the robot located in another pose, are matched. Maximizing the scan overlapping between 2D rigid and the real 2D dimension is the aim of the matching algorithm. This goal is reached by comparing the coordinates of corresponding scan points to be matched. The problem of identifying the points to be matched in different scans is a fundamental aspect of many matching algorithms and has various applications, in particular, computer vision, pattern recognition, and robotics.

The Iterative Closest Point algorithm (ICP) [1] is a popular method for scan matching and has variations such as heuristic methods for aligning 2D or 3D point sets. ICP has three main steps as follows: (i) pair each point of the first set to the second one using a corresponding criterion; (ii) compute the rotation and translation transformations which minimize the mean square error between the paired points; (iii) apply the transformation to the first set. In ICP as the problem is characterized by local minima, convergence is not guaranteed, and the pre-alignment of the views to converge to the best global solution, as it is required, can be considered as the drawback of ICP.

Most of the works in shape matching have covered the 3D case situation, but the same approaches can be applied in the 2D case as well. In some works for registration of partially overlapping 2D

* Corresponding author.
*E-mail address:* rana.forsati@gmail.com (R. Forsati).

0921-8890/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.robot.2013.04.009

and 3D data by minimizing the mean square error cost function is represented and done by a genetic algorithm.

In this paper we propose two harmony search based algorithms for mobile robot localization. The proposed algorithms model the mobile localization as an optimization problem, such that the final results reveal the superiority of the work over the other evolutionary and non-evolutionary methods.

The proposed algorithms can guide the robot better without any predefined map. Furthermore, the stochastic nature of harmony search allows the procedures to handle and deal with the level of available noise in the sensors better. The other advantages of this work are its simplicity in implementation and low computational cost that allow the procedure to be applied in an online fashion. But unfortunately the harmony search itself has some problems such as premature convergence. To tackle this issue we hybridized it with the DE algorithm which results in a better convergence. In the hybridized HS, the method benefits from the convergence speed of DE and the search quality of HS. The algorithm not only was applied to the standard datasets but also other real-world datasets were considered to assess the goodness of the proposed solutions.

The rest of the paper is organized as follows: in Section 2 the recent works are surveyed. Section 3 describes the HS algorithm and its implementation for mobile robot localization. Some simulation results are reported to demonstrate the performance of localization with the HS algorithm. In Section 4 fusion of HS and DE algorithms is explained and the hybrid algorithm has been evaluated with some benchmark functions. In Section 5, we used the HIDE algorithm for mobile robot localization and compared the simulation results with the results of the HSL algorithm. Finally, Section 6 concludes the paper.

## 2. Literature review

In [2] the authors used GAs for alignment of multi view range images in order to perform evaluate registration results more precisely a novel Surface Interpenetration Measure is introduced, especially in the case of small registration errors. Although in [2] to handle partial overlap a predefined residual threshold is used, no method is given to adjust the threshold automatically. Ref. [3] represents a combined approach in which a genetic search is used to initialize an efficient iterative alignment procedure. Their system is applicable to arbitrarily oriented surfaces while preserving the precision and robustness of iterative methods.

Lu and Milios have proposed the IDC algorithm [4] based on ICP to alleviate the problem of having a good pre-alignment. In recent years many algorithms based on the ICP algorithm [5] and hybridized algorithms with ICP [6] have been proposed. In 2002 Moreno succeeded to localize a mobile robot with a soft computing method for the first time [7]. A different approach for improving a given model is to use the simulated annealing algorithm. This algorithm was used by Kirkpatrick et al. [8] to solve optimization problems associated with the design of integrated circuits like component placement and wiring. They also apply the algorithm to the classic optimization problem of the traveling salesman.

Ducket defined the SLAM problem as an optimization problem and solved it through the genetic algorithm [9] and in 2007 Moreno solved the localization problem of a mobile robot with the Differential Evolution algorithm [10]. Generally speaking, mobile robot localization finds out a robot's coordinates relative to its environment, assuming that one is provided with a map of the environment. Localization is a key component in navigation and required to execute successfully a trajectory.

We can distinguish two different cases: the re-localization case and the global localization case. The re-localization or tracking problem tries to keep track of mobile robot's pose, where the robot knows its initial position (at least approximately) and therefore has to maintain localized the robot along the given mission. The global localization problem does not assume any knowledge about the robot's initial position and therefore has to globally localize itself.

The two most important aspects that have to be dealt with when designing a localization system are how to represent the uncertain information of the environment and the robot's pose. Among the many ways to represent the knowledge about an environment, this article deals with geometrical localization methods and assumes that the environment is modeled geometrically as an occupancy grid map. In the robot's pose uncertainty representation and estimation techniques, the vast majority of existing algorithms address only the position tracking problem. In this case the small incremental errors produced along the robot motion and the initial knowledge of the robot's pose makes classical approaches such as Kalman filters or scan matching techniques applicable which stated above.

If we consider the robot's pose estimation as a Bayesian recursive problem, Kalman filters estimate the posterior distribution of robot's poses conditioned on sensor data. Based on the Gaussian noise assumption and the Gaussian-distributed initial uncertainty this method represents posterior distributions by Gaussians. The Kalman filter constitutes an efficient solution for re-localization problems. However, the assumption's nature of the uncertainty representation makes Kalman filters not robust in global localization problems. Scan Matching techniques are also an iterative local minimization technique and cannot be used for global localization. Different families of algorithms can solve the global localization problem; some frequently used are: multi-hypothesis Kalman filters, grid-based probabilistic filters and Monte Carlo localization methods. Those methods can be included in a wider scope group of Bayesian estimation methods.

Multi-hypothesis Kalman filters [11–15] represent distributions using mixtures of Gaussians, enabling them to keep track of multiple hypotheses, each of which is represented by a separate Gaussian. This solution presents some initialization problems: one of them is the determination of the initial hypotheses (the number can be very high and it is not bounded); this leads the algorithm to a high computational cost at initial stages. Besides, the Kalman filter is essentially a gradient-based method and consequently poorly robust if the initial hypothesis is bad or noise assumptions fail.

Grid-based localization algorithms [16–18] represent distributions by a discrete set of point probabilities distributed over the space of all possible poses. This group of algorithms is capable of representing multi-modal probability distributions. A third group is the Monte Carlo localization algorithms [19–21]. These algorithms represent the probability distribution by means of a set of samples drawn according to the posterior distribution over robot's poses. These algorithms can manage arbitrary noise distributions and non-linearities in the system and observation models. These methods present a high computational cost due to its probabilistic nature which requires a high number of samples to draw properly the posterior probability density function. The main advantage is its statistical robustness.

In the robotic literature many different strategies have been proposed to learn efficient geometric representations from range data.

One strategy was described by [22] who uses a Kalman filter to fit lines on range measurements obtained from sonar sensors. In the paper [23], point clusters are computed from each range scan based on the distance between consecutive points. Linear regression is then applied to fit lines to these clusters. [24] uses a hierarchical clustering approach to extract lines from the points obtained from laser data. The strategy proposed by [25] uses a Hough transform to extract linear features from a sequence of consecutive sonar measurements.

The approach presented by [26] clusters scans using the split-and-merge algorithm and combines nearby segments using a weighted variant of linear regression. A recent work [27] presents two approaches for extracting lines from laser data. They use the Hough transform and the EM algorithm to extract lines out of the set of points obtained from the range measurements. Both approaches work on the complete dataset in contrast to techniques that work on individual scans. Similar to these approaches, our techniques also work on the complete dataset. However, we do not depend on the extraction of features to construct a representation of the environment. The EM algorithm has also been used by [28] to learn planar structures from 3D data and by [29] to apply the EM algorithm to cluster different types of objects like walls and doors from sequences of range data. [30] uses the EM algorithm for learning maps using sonar data and [31] for learning the motion behaviors of persons. The $k$-means and fuzzy $k$-means algorithms we used in our work to improve the quality of a sample-based map are instances of the EM algorithm.

Another kind of problem is the Simultaneous Localization and Mapping (SLAM) problem, also known as the Concurrent Mapping and Localization (CML) problem, which is often recognized in the robotics literature as one of the key challenges in building autonomous capabilities for mobile vehicles.

The goal of an autonomous vehicle performing SLAM is to start from an unknown location in an unknown environment and build a map of its environment incrementally by using the uncertain information extracted from its sensors, whilst simultaneously using that map to localize itself with respect to a reference coordinate frame and navigate in real-time.

In SLAM the map is represented in one of the several forms. A common approach is to use features extracted from exteroceptive sensors' data [32]. Another approach is to use a grid to represent the map [33]. Initial solutions to the feature-based SLAM problem used an Extended Kalman Filter (EKF) based stochastic mapping framework, which estimated the vehicle pose and the map feature (landmark) positions in an augmented state vector using second order statistics. Although EKF-based SLAM within the stochastic mapping framework gained wide popularity among the SLAM research community, over time, it was shown to have several shortcomings. Notable shortcomings are its susceptibility to data association errors, non-Gaussian noises and inconsistent treatment of non-linearities. The grid-based approach, on the other hand, represents the environmental map as a field of random variables, arranged in an evenly spaced grid. It does not rely on predefined feature extractors, which assumes that some structures of the environments are known in advance.

The main disadvantages are the high computational costs and memory complexity when high fidelity localization and mapping are demanded by the application. Monte Carlo based probabilistic filters, in theory, provide for effective handling of non-linear and non-Gaussian models which are inherent in robot localization. The well-known Monte Carlo localization (MCL) algorithm [34], FastSLAM [35], and DP-SLAM [36] are good examples of realization of such filters. However, the standard implementation of MCL-based filters requires solving problems including the particle deprivation problem, the divergence of the proposal and target distribution and the sample impoverishment problem.

Firstly, there may be no particles in the vicinity to the correct state, especially when the area is very large compared to the small population size, or when the sensor measurement is comparatively much more accurate compared to the motion estimation. In addition, adequate spread of the proposal and target distribution must be assured. This can be explained in the way that it needs an acceptable 'initial guess'.

The efficiency of the particle filter relies crucially on the 'match' between the proposal and the target distribution. Finally, it is very common that some 'very fit' particles may dominate the whole particle set after a few re-samplings, especially when the number of particles is small. That is after several iterations particles may converge to a few 'points' and become incapable of approximating the true distribution [34,37].

To mitigate these problems, some heuristic methods, such as sample boosting [38], and mixture-MCL [34], have been proposed. The complexity and high computational costs make them difficult to implement. The evaluative optimization techniques constitute a series of probabilistic search methods that avoid derivatives or probability density estimations to estimate the best solution to a localization problem.

The harmony search (HS) algorithm is a new meta-heuristic optimization method that was recently introduced by Geem et al. [39]. The HS algorithm presents several advantages with respect to traditional optimization techniques such as imposing fewer mathematical requirements and generating new vectors after considering all of the existing vectors, whereas methods like the genetic algorithm (GA) only consider the two parent vectors.

This paper presents a novel harmony search localization (HSL) algorithm. HSL solves the global localization problem of a mobile robot in a robust and efficient way. The algorithm can deal with arbitrary noise distributions and non-linear state space systems. The main idea of HSL is to represent the uncertainty about the robot's pose by a set of possible pose estimates weighted by a fitness function. Simulation results show that the proposed localization method in comparison with the genetic approach has better accuracy and higher performance.

In the HS algorithm update of the HM is solely based on the past search experiences. DE is a gradient-based method and has good convergence speed but sometimes converges to local minima. When population converges to local minima, hence update of the solution vectors is solely based on the population members, the algorithm is not able to run away from it. With fusion of HS and DE we can conquer their individual drawbacks while benefit from each other's strengths. Enlightened by this, the DE realization concept has been used in the HS in Section 3 to explore the potential solution space and increase the speed of convergence. The proposed method, called HIDE (Harmony memory Improvement with Differential Evolution), is based on the common characteristics of both DE and HS algorithms.

## 3. Preliminaries

In this section we present the basic harmony memory, which is a basic requirement for understanding the proposed method. Harmony search (HS) is a new meta-heuristic optimization method imitating the music improvisation process where musicians improvise their instruments' pitches searching for a perfect state of harmony [39]. The harmony search algorithm has been very successful in a wide variety of optimization problems [40–45]. The HS algorithm presents several advantages with respect to traditional optimization techniques, such as follows: (a) the HS algorithm imposes fewer mathematical requirements and does not require initial value settings for decision variables, (b) derivative information is unnecessary due to usage of stochastic random searches in the HS algorithm, and (c) the HS algorithm generates a new vector, after considering all of the existing vectors, whereas a method like the genetic algorithm only considers the two parent vectors. These features increase the flexibility of the HS algorithm and produce better solutions. The main steps of the HS algorithm are as follows.

*Step* 1: Initialize the problem and algorithm parameters.

First the optimization problem and algorithm parameters should be initialized, which are as below.

- The harmony memory (HM) that is a memory location where all the solution vectors (sets of decision variables) are stored in. This HM is similar to the genetic pool in the GA.

$$\mathrm{HM}=\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix}$$

**Fig. 1.** Harmony memory sample representation.

- The harmony memory size (HMS), or the number of solution vectors in the harmony memory.
- The harmony memory consideration rate (HMCR) which is the rate of choosing one value (between intervals 0 and 1) from the historical values stored in the HM.
- The probability of pitch adjusting (PAR).
- The number of improvisations (NI) or stopping criterion.

*Step* 2: Initialize the harmony memory.

The HM matrix is filled with randomly generated solution vectors. For an $N$-dimension problem, an HM with the size of HMS can be represented as in Fig. 1.

The initial harmony memory is generated from a uniform distribution in the ranges $[LB_i, UB_i]$, where $1 \le i \le N$. This is done as follows:

$$x_i^j = LB_i + r \times (UB_i - LB_i), \quad j = 1, 2, 3, \ldots, \mathrm{HMS} \tag{1}$$

where $r \sim U(0, 1)$ and $U$ is a uniform random number generator.

*Step* 3: Improvise a new harmony.

Generating a new harmony is called improvisation. A new harmony vector, $x' = (x'_1, x'_2, \ldots, x'_N)$, is generated using the following rules:

- memory consideration: the probability of memory consideration equals HMCR;
- pitch adjustment rate: the probability of the pitch adjusting process equals HMCR×PAR;
- random selection: the probability of random selection equals (1-HMCR).

In the memory consideration, the value for a decision variable is randomly chosen from the historical values stored in the HM with the probability of HMCR. Every component obtained by the memory consideration is examined to determine whether it should be pitch-adjusted. This operation uses the PAR parameter, which is the probability of pitch adjustment. Variables which are not selected for memory consideration will be randomly chosen from the entire possible range with a probability equal to (1-HMCR). The pseudo code shown in Fig. 2 describes how these rules are utilized by the HS.

*Step* 4: Update the harmony memory.

If the new harmony vector, $x' = (x'_1, x'_2, \ldots, x'_N)$, has a better fitness function than the worst harmony in the HM, the new harmony is included in the HM and the existing worst harmony is excluded from the HM (the new harmony will replace the worst in HM).

*Step* 5: Checking the stopping criterion.

The HS is terminated when the stopping criterion (e.g., the maximum number of improvisations) has been met. Otherwise, Steps 3 and 4 are repeated. In Fig. 3 we can see the flowchart of the HS algorithm. The HMCR and PAR parameters of HS help the method in searching for globally and locally improved solutions, respectively. PAR and BW have a profound effect on the performance of HS. Thus, fine tuning of these two parameters is very important.

Between these two parameters, BW is more difficult to be tuned because it can take any value from $(0, \infty)$. To address the shortcomings of HS, a new variant of HS, called the improved harmony search (IHS), is proposed [46]. IHS dynamically updates PAR according to the following equation.

$$\mathrm{PAR}(t) = \mathrm{PAR}_{\min +} \frac{(\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min})}{\mathrm{NI}} \times t \tag{2}$$

where $t$ is the generation number, $\mathrm{PAR}(t)$ is the pitch adjusting rate for generation $t$, $\mathrm{PAR}_{\min}$ is the minimum adjusting rate and $\mathrm{PAR}_{\max}$ is the maximum adjusting rate.

In addition, BW is dynamically updated as follows:

$$bw(t) = bw_{\max} e^{\left(\frac{\ln\left(\frac{bw_{\min}}{bw_{\max}}\right)}{\mathrm{NI}} \times t\right)} \tag{3}$$

where $bw(t)$ is the bandwidth for generation $t$, $bw_{\min}$ is the minimum bandwidth and $bw_{\max}$ is the maximum bandwidth.

## 4. Proposed algorithm

In this section, we will propose a method for mobile robot localization via scan matching through the harmony search algorithm. Since PAR and BW are changeable during the running of the algorithm, the HSL algorithm may solve the reported

**Algorithm 1. Improvise a new harmony**

**1. Input : current solutions in harmony memory HM**
**2. Output: new harmony vector** $x' = (x'_1, x'_2, \cdots, x'_N)$
3. **for each** $i \in [1,N]$ **do**
4. **if** $U(0,1) \le$ **HMCR then** ***/* memory consideration */***
5. $x'_i = HM[j][i]$ where $j \sim U(1,2,\ldots,\mathrm{HMS})$.
6. **if** $U(0,1) \le$ **PAR then** ***/* pitch adjustment */***
7. $x'_i = x'_i \pm r \times bw$, where $r \sim U(0,1)$ and bw is an arbitrary distance bandwidth
8. **end if**
9. **else** ***/* random selection */***
10. $x'_i = LB_i + r \times (UB_i - LB_i)$
11. **end if**
**12. end for**

**Fig. 2.** Creating a new harmony vector.

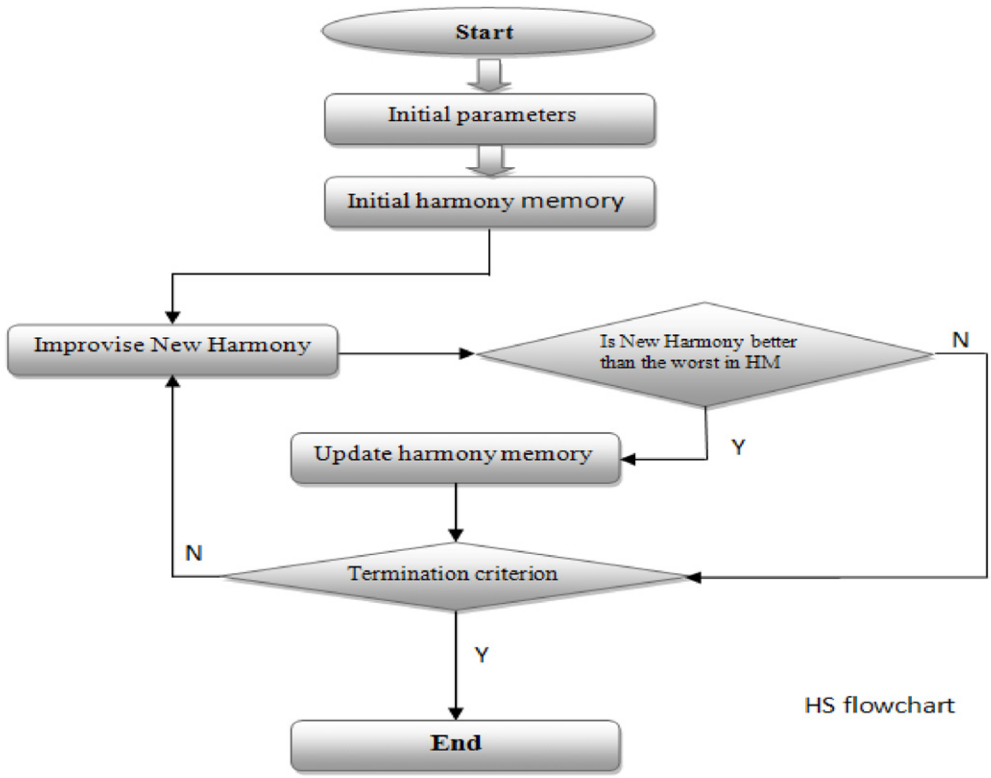



**Fig. 3.** HS algorithm flowchart.

drawbacks of the GLASM localization method [47]. Some of these drawbacks could be falling into local minima due to decreasing genetic diversity through a small variation in mutation operator and time complexity for binary to gray coding–decoding. The robot pose shows the robot location and orientation $X_t = (x, y, \theta)$. Odometry data and laser range measurements are represented by $U_t$ and $S_t$ respectively. The subscript $t$ indicates the discrete time index. The map $m_{X_t}$ in a global coordinate system will be generated upon receiving the $t$th sensor measurement $S_t$ at the pose $X_t$ as follows:

$$\begin{cases} m_{x_t} = R(\theta) \cdot S_t + (x_t + y_t) \\ R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \\ U_t = (x_t, y_t, \theta_t) \end{cases} \tag{4}$$

where $R(\theta)$ is the orientation matrix. So according to this equation, we can transform each scan from the robots' local coordinate system to the global coordinate system and compare with other scans.

### 4.1. Harmony memory initialization

According to the problem statement, harmony memory (HM) might be written as shown in Eq. (5). One of the solution vectors (for example $X_t^1(x_t^1, y_t^1, \theta_t^1)$) is obtained from robots' last pose or from odometry data (in local scan matching). $x_t^l$ is the estimated robot pose. We do as follows for generating $x_t^2$ to $x_t^{\mathrm{HMS}}$ vectors:

$$\mathrm{HM} = \begin{bmatrix} X_T^1 & Y_T^1 & \theta_T^1 \\ X_T^2 & Y_T^2 & \theta_T^2 \\ X_T^3 & Y_T^3 & \theta_T^3 \\ & \vdots & \\ X_T^{\mathrm{hms}} & Y_T^{\mathrm{hms}} & \theta_T^{\mathrm{hms}} \end{bmatrix}. \tag{5}$$

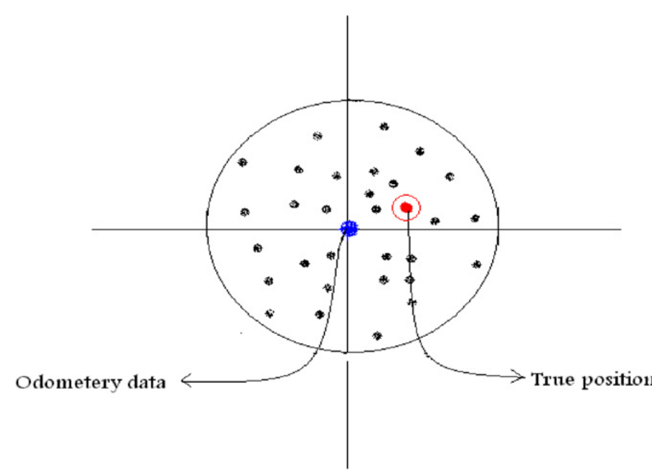



**Fig. 4.** Search space.

We arrange a circle with a radius of $R$ depending on the maximum uncertainty around $X_t^1$ and generate other solution vectors randomly inside the circle as shown in Fig. 4. So we will have other vectors inside the following areas:

$$\begin{cases} x_t^i \in (x_t^1 - R, x_t^1 + R) \\ y_t^i \in (y_t^1 - R, y_t^1 + R) \\ \theta_t^i \in (\theta_t^1 - \varphi, \theta_t^1 + \varphi). \end{cases} \tag{6}$$

Where $i = 1, 2, 3, \ldots,$ HMS, and is depending on maximum uncertainty in robot orientation.

### 4.2. Improvising a new harmony

In the improvising step, we need a technique to generate a new harmony vector (NHV), $X_t' = (x_t', y_t', \theta_t')$, from all the HMS solution vectors that are in HM. The new generated harmony vector must inherit as much information as possible from the solution vectors that are in HM. If the generated vector, which is corresponding to a new robot pose (robot location and orientation), consists mostly or entirely of assignments found in the vectors in HM, it provides good heritability.

In this algorithm, each decision variable corresponds to a robot location and orientation and its value shows the value of that location and orientation. For obtaining each component of $(x'_t, y'_t, \theta'_t)$, we should perform the following process that is shown for $x'$ and can be done in the same way as for the other components. The value of each robot pose in the new solution vector is selected with the probability of HMCR from the harmony memory and with the probability of (1-HMCR) is randomly selected from the set $\{1, 2, \ldots, k_i\}$. A random component from HM with the probability of HMCR is selected according to the following equation.

$$\begin{cases} x'_t = x^i_t \\ i = \text{rand}(1, 2, \ldots, \text{HMS}). \end{cases} \tag{7}$$

After generating the new solution, the PAR process is applied. PAR is originally the rate of allocating a different value to a robot pose. PAR in the HS algorithm is a very important parameter for fine-tuning of an optimized solution vectors and can be potentially useful in adjusting the convergence rate of the algorithm to optimal solution.

We propose a scanning method for generating a new solution. The general mechanism for scanning is to assign a marker to each row in HM as well as to the new harmony vector (NHV). The marker for the new vector traverses through all positions from left to right, one at a time. In each step, the markers for the rows of HM are updated so that in the time of choosing a value for the currently marked element in the new vector, the row markers show the possible choices. Each value is selected with the probability of HMCR from marked elements and with the probability of (1-HMCR) is randomly selected among the entire valid possible range for that location and orientation.

After generating the new solution, the PAR process is applied. In the original HS algorithm, PAR is the rate of moving from a current selected element to a neighboring element. PAR in the HS algorithm is a very important parameter in fine-tuning of optimized solution vectors and can be potentially useful in adjusting the convergence rate of the algorithm to the optimal solution. For each location of robot that is selected from harmony memory the location can be replaced by a valid value that can be gain through following equation.

$$x'_t = x'_t \pm Bw(N). \tag{8}$$

### 4.3. Update the harmony memory

If the new harmony vector $X'_t = (x'_t, y'_t, \theta'_t)$ has the better fitness function than the worst harmony in HM then the new harmony will replace the worst.

### 4.4. Check the stopping criterion

There is no stopping criterion which ensures the convergence of an HS to an optimal solution. We have used the two most usual criteria. Our HS stops:

- After a number of iterations in which the average fitness does not change.
- When the maximum number of generations is reached.

The map obtained from discrete time $t$ will be used as a reference map in discrete time $t + 1$. After meeting the stopping criterion, the best component in HM will determine the robot pose in time $t$.

### 4.5. Evaluation of solutions

The quality of the solutions, produced by HS, relies on the stochastic nature of the technique and the way in which the objective function is converted to a fitness function that can guide the algorithm to the desired region of the search space. As a result, designing a good fitness function is a key problem in solving problems with the HS method.

Each row in HM corresponds to a possible robot pose. Now we need a function to evaluate the degree of matching between two scans. The problem of aligning two partially overlapping surfaces represented by points obtained in subsequent 2D scans for mobile robot pose estimation is solved in this paper. The measured point representation contains incomplete measurements. We solve this problem by minimizing an alignment error by using the harmony search algorithm. Given a 2D reference scan gathered with a range sensing device by the robot located in a given pose and another 2D scan gathered by the robot located in another pose, the goal is to determine the 2D rigid motion that maximizes the scan overlapping.

The scan matching is performed by comparing the coordinates of corresponding scan points to be matched. We try to align the two sets of points. We perform scan matching by using corresponding point measurement. Each row in HM corresponds to a candidate scan matching. Our objective function is to discover the proper scan matching for robot localization for maximizing the scan overlapping (minimizing an alignment error), as well as minimizing the alignment error (maximizing the overlapping between two scans).

The fitness value of each row, which corresponds to a potential solution, is determined by accumulating matching error and normalizing it by the number of valid corresponding points of that row. Since the fitness computation should be carried out quickly, we have applied the fitness function presented in [47]. This fitness function does not require finding corresponding points which makes it fast and accurate. The fitness function is formulated by accumulating matching error and normalizing it by the number of valid corresponding points. The fitness function is shown in Eq. (9):

$$F_{s_{\text{new}}, s_{\text{ref}}} = \sum_{i=1}^{N} \rho(i) \tag{9}$$

where $\rho(i)$ is as follows:

$$\rho(i) = \begin{cases} 1 & \text{if a point of } s_{\text{new}} \text{ lies in square around } S_i \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Around each point a square was assumed, and if the results of the new scan falls into that region one positive point will be assigned otherwise no point will be considered. The purpose is to maximize $\rho(i)$ value, in other words to maximize the overlaps.

## 5. Hybridization

The HS algorithm is based on the past search experience and empirical studies have shown that sometimes HS suffers from a slow search speed [41]. To overcome this shortcoming, we proposed a novel algorithm with fusion of HS and DE. The proposed algorithm, called HIDE (harmony memory improvement with differential evolution), is based on the characteristics of both DE and HS algorithms. The HS algorithm provides a new way to produce new solution vectors. Different from DE and GA, the HS algorithm generates a new vector after considering all of the existing vectors. On the other hand, DE is a gradient-based method and has good convergence speed [48]. The convergence speed of DE and the search speed of HS are considered as benefits of the proposed HIDE algorithm. In this section DE is first explained and then we describe the hybrid method.

### 5.1. Differential evolution (DE) method

In 1995, Price and Storn proposed a new floating point encoded evolutionary population based algorithm for global optimization, namely as DE [49]. Easy methods of implementation and negligible parameter tuning made the algorithm popular very soon. In the next sections we have outlined the classical and one other popular version of DE.

#### 5.1.1. Simple DE

In the classical DE, chromosomes are updated by adding the weighted difference between two chromosomes to the original one. If the resulting chromosome yields a better objective function than that chromosome, this newly generated chromosome replaces the original one; otherwise, the old vector is retained. The simplest DE can be explained as follows; suppose that $x_i(k)$ is a solution vector that should be updated in iteration $k+1$. A trial update of $x_i(k)$, $\acute{x}_i(k+1)$, is as follows:

$$\acute{x}_i(k+1) = L(x_i(k)) + F(x_{r_1}(k) - x_{r_2}(k)) \tag{11}$$

where $L$ and $F$ are two pre-determined real and constant factors which control the amplification of the differential variations and $x_{r_1}(k)$, $x_{r_2}(k)$ are two solution vectors chosen randomly from the population and should be different from the running index $r_1$, $r_2$.

In order to increase the diversity of solution vectors, a crossover operator is employed to combine $x_i(k)$ and $\acute{x}_i(k+1)$ and it generates $x_i''(k+1)$. If $x_i''(k+1)$ yields a better fitness than $x_i(k)$ then replace it otherwise $x_i(k)$ is retained for the new generation.

#### 5.1.2. Scheme DE/rand to best/1

DE/rand to best/1 [50] follows the same procedure as that of the simple DE scheme. The only difference is that now the donor vector is used to perturb each population member, is created using any two randomly selected members of the population as well as the best vector of the current generation (i.e., the vector yielding the best suited objective function value at $t = k$). This can be expressed for the $i$th donor vector at time $t = k + 1$ as

$$\begin{aligned}\acute{x}_i\,(k+1) = {} & x_i(k) + \lambda\,(x_{\mathrm{best}}(k) - x_i(k)) \\ & + F\left(x_{r_1}(k) - x_{r_2}(k)\right)\end{aligned} \tag{12}$$

where $\lambda$ is another control parameter of DE in the [0, 2] interval, $x_i(k)$ is the target vector and $x_{\mathrm{best}}(k)$ is the best member of the population regarding fitness at current iteration $k$ [48]. To reduce the number of control parameters a usual choice is to put $\lambda = F$.

### 5.2. Hybrid algorithm

This part of the paper describes the implementation of the proposed improvement in HS using the DE approach. The proposed method, called HIDE, is based on the common characteristics of both DE and HS algorithms. The HS algorithm provides a new way to produce new solution vectors. Different from DE and GA, the HS algorithm generates a new vector after considering all of the existing vectors.

The HS algorithm can produce a new solution and the parameters of HMCR and PAR are introduced to allow the solution to escape from local minima and improve the global optimum prediction of the algorithm but the HS method sometimes suffers from a slow search speed because the update of the HM is solely based on the past search experiences. DE is a gradient-based method and has fine convergence speed but sometimes converges to local minima.

Since the selection of components of the NHV is based on the harmony memory solutions, any premature convergence can cause local entrapment and the algorithm cannot escape from it. With fusion of HS and DE we can address their individual shortcomings while benefit from each other's strengths. Enlightened by this, the DE realization concept has been used in the HS in this section to explore the potential solution space and increase the speed of convergence. In this hybrid algorithm all the members of the HM are regarded as the DE individuals. In summary, the realization of improved HS can be described as follows:

- Initializing the parameters of DE and HS.
- Initializing the solution vectors (HM).
- Performing HS and generating HRT[1] new solutions and replacing them if they satisfy the replacement criterion.
- Performing DE and updating HM according to Eq. (12).
- Finishing the program if the termination condition is met otherwise going to step 3.

In Fig. 5, we can see the flowchart of the proposed hybrid algorithm.

### 5.3. Mobile robot localization with the proposed hybrid algorithm

In this section, we proposed a method for mobile robot localization via scan matching through the HIDE algorithm and then compared the new localization method with the HSL method. According to the problem statement, initialize the harmony memory (HM) as in Eq. (5).

All the members of the HM are regarded as the DE individuals. After initializing the HM, we should generate new solution vectors same as in Section 4.2 through Eqs. (7) and (8). If the new harmony vectors have better objective function than the worst harmony in HM then the new vectors will replace the worst. The HS algorithm will terminate when the maximum number of iterations (HRT) has been met. Otherwise generating new solution vectors and updating the HM will be repeated. After terminating HS all the solution vectors should update through the DE algorithm. The HIDE algorithm will terminate when the number of fitness evaluation (NFE) is met, otherwise HS and DE algorithms should run again respectively. The map obtained from discrete time $t$ will be used as a reference map in discrete time $t + 1$. After meeting the stopping criterion, the best component in HM will determine the robot pose in time $t$.

## 6. Experimental results

In order to validate the presented HSL and HIDE localization algorithms a simulation program based on the real data obtained from a laser range scanner (URG-04LX) fixed on an MRL autonomous mobile robot is arranged. The robot's photo is shown in Fig. 6.

As we have tested the algorithm by real data from a navigating robot in Mechatronics Research Laboratory-MRL, the simulation results could also be interpreted as experimental results. The results of HSL and GLASM [47] and HIDE localization algorithm are compared.

Some of the terms which are used in this section are defined below.

- *Successful matching*: A matching that results in an estimated pose close to the real pose $(x_{\mathrm{true}}, y_{\mathrm{true}}, \theta_{\mathrm{true}})$.
- *Success ratio*: Ratio between the number of successful matching ($\mathrm{NM}_{\mathrm{succ}}$) and the number of total matching ($\mathrm{NM}_{\mathrm{tot}}$) performed in one set of localization trials.

[1] Harmony run time.

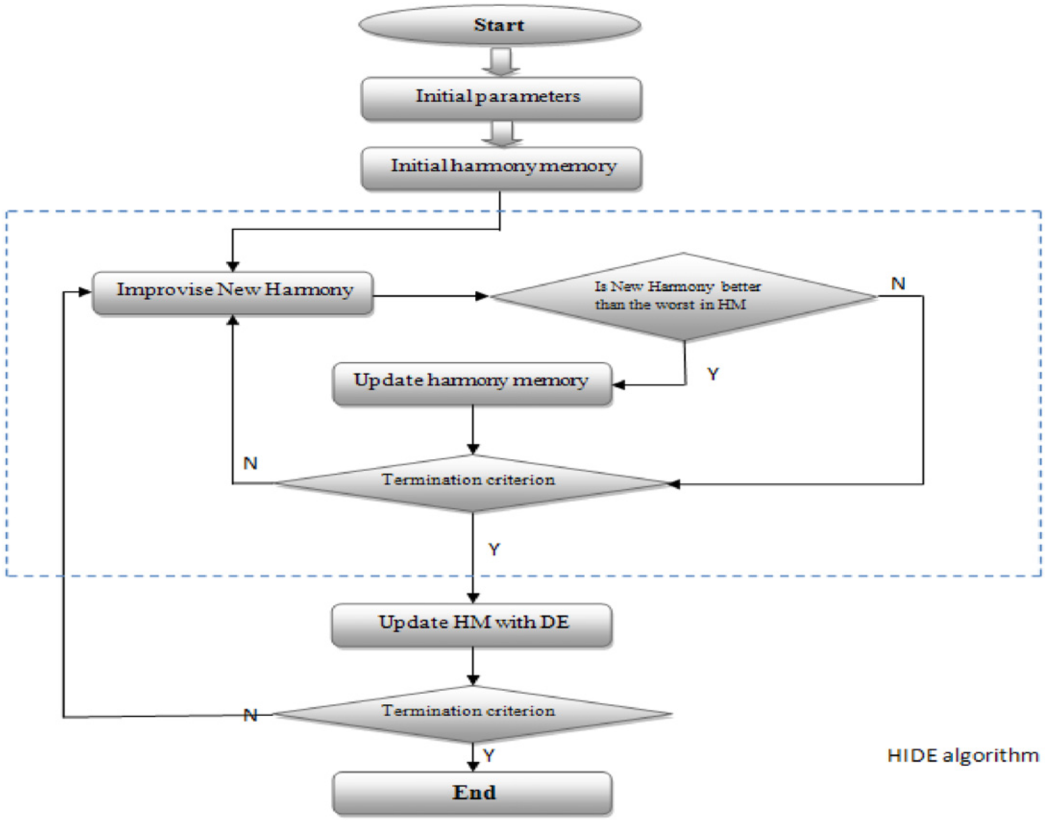



**Fig. 5.** HIDE flowchart.

**Fig. 6.** MRL autonomous robot equipped with a URG-04LX scanner.

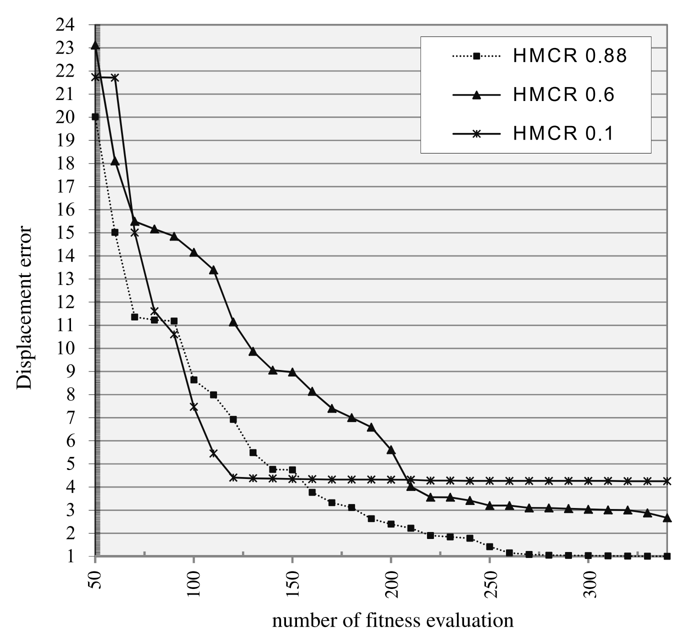



**Fig. 7.** HMCR value setting empirical studies.

- *Displacement error*: The error in the robot pose obtained from the robot position.
- *Rotation error*: The error in the robot pose obtained from the robot orientation.

$$SR = \frac{NM_{secc}}{NM_{tot}}. \tag{13}$$

The maximum uncertainty in the robot pose that defines the search space is ($\pm 15$ cm, $\pm 15$ cm, $\pm 10^\circ$) for $(x, y, \theta)$ Therefore the search space is $L = 30$ cm and $\varphi = 10^\circ$.

In this part of the paper some experiments are done regarding the justification of harmony search parameters as a dynamic parameter. These are the harmony memory size (HMS) and the harmony memory considering rate (HMCR). Keeping that in mind we will now show the effects of single parameter changes.

In Fig. 7, we see the effects of variation of the HMCR regarding the justification of HMCR as a dynamic parameter. As mentioned earlier, the HMCR determines the rate of choosing one value from the historical values stored in the HM. The larger the HMCR the less the exploration is achieved, and the algorithm further relies on the stored values in HM and this potentially leads to the algorithm getting stuck in a local optimum. On the other hand, choosing HMCR too small will decrease the algorithm efficiency and the HS algorithm behaves like a pure random search, with less assisting from historical memory.

As seen in Fig. 7 by increasing the HMCR value from one hand and then decreasing the amount of fitness evaluation, the

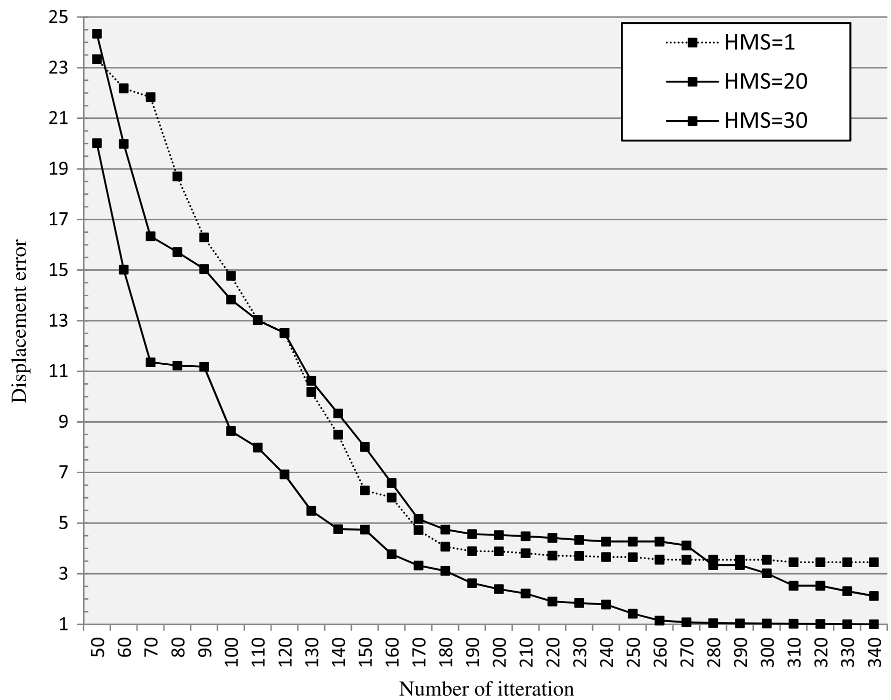



**Fig. 8.** HMS value setting empirical studies.

**Table 1**
Comparison between GLASM and HSL algorithms.

| Algorithm | Maximum uncertainty $(L, \theta)$ | Convergence time (ms) | SR | Displacement error (cm) | Rotation error (°) |
|---|---|---|---|---|---|
| GLASM | | 165 | 0.88 | 2.12 | 0.12 |
| HSL | (20, 10) | 158 | 0.97 | 1.44 | 0.11 |
| Improv% | | 4.24% | 10.22% | 32% | 8.3% |
| GLASM | | 273 | 0.81 | 2.27 | 0.13 |
| HSL | (30, 10) | 252 | 0.96 | 1.45 | 0.11 |
| Improv% | | 7.6% | 18.5% | 36% | 15.3% |

**Table 2**
Comparisons between HIDE localization and HSL algorithms.

| Algorithm | Maximum uncertainty $(L, \theta)$ | Convergence time (ms) | SR | Displacement error (cm) | Rotation error (°) |
|---|---|---|---|---|---|
| HSL | | 158 | 0.97 | 1.44 | 0.11 |
| HIDE | (20, 10) | 141 | 0.96 | 1.43 | 0.11 |
| Improv% | | 10.75% | −1.03% | 0.69% | 0% |
| HSL | | 252 | 0.96 | 1.45 | 0.11 |
| HIDE | (30, 10) | 223 | 0.95 | 1.45 | 0.11 |
| Improv% | | 11.51% | −2.06% | 0% | 0% |

displacement error rate is kept high, while through increasing the amount of fitness evaluation this rate decreases and possesses the lowest amount when HMCR is 0.88. Therefore to decline the displacement rate, a 0.88 value for HMCR has shown the best performance in our implementation.

In Fig. 8 the solution evolution for different HMS is shown. We can see that decreasing the HMS leads to premature convergence and increasing the HMS leads to significant improvements in the initial phase of a run. Note that when the time or number of iterations is finite, increasing the HMS may deteriorate the quality of the solution. In general we can say, the larger the HMS, the more the time (or iterations) is needed for the algorithm to find the optimal solution, but usually higher quality is achieved. In the proposed algorithm HMS $= 30$ is reasonable.

For the HS algorithm parameters we have found the best results in the experiments as follows:

HMS $= 30$, HMCR $= 0.88$, $\text{PAR}_{\min} = 0.55$, $\text{PAR}_{\max} = 0.98$, NI $= 250$, $\text{BW}_{\min} = 0.2$ and $\text{BW}_{\max} = 8$.

The position and orientation obtained for the robot with our presented algorithm is (0.77, 1.23, 0.11) with the HSL algorithm while robot's true pose is (0, 0, 0). In Table 1, we can see that the robot pose obtained from the HSL algorithm is better than that from the GLASM algorithm through the changeable BW, but the most difference between two algorithms is in the success ratio.

Our proposed algorithm has much higher SR through the harmony search specification to get away from local minima. If we repeat the tests with lower uncertainty, we can see that in the HS-based algorithm, just the time of convergence is decreased, but in GLASM, both time and SR are changed and it shows that the SR of GLASM depends on the uncertainty and search space but HSL is not like that. According to Table 1 results the improvement percentage of HSL compared to GLASM is in the range of 4%–36%. In Table 2 the results of HIDE and HSL are compared to show the priority of the HIDE algorithm. The most difference between the two algorithms is in convergence time. The HIDE algorithm has better convergence time due to the characteristics of the DE algorithm.

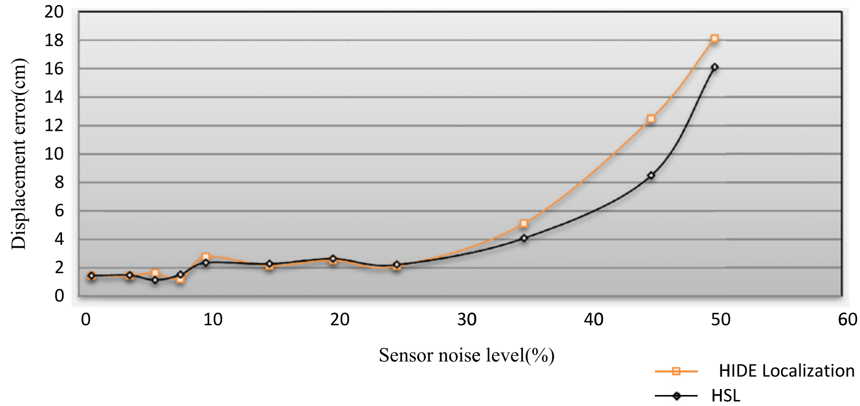



**Fig. 9.** Effect of the noise on the accuracy of the HSL and HIDE algorithms.

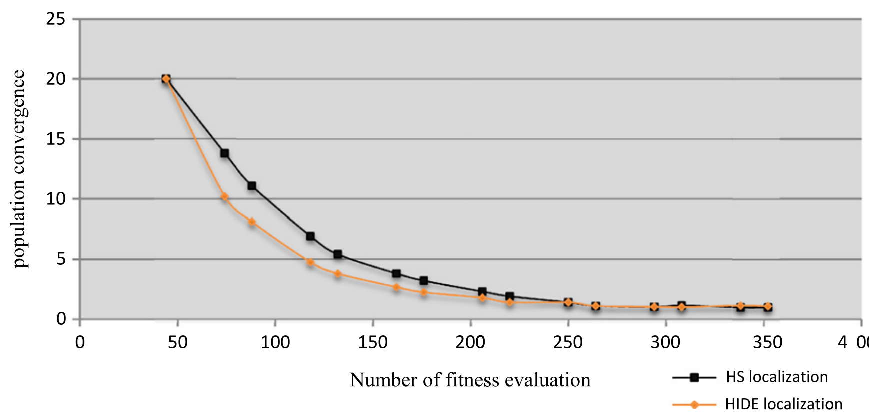



**Fig. 10.** Population convergence (displacement error) as a function of fitness evaluation.

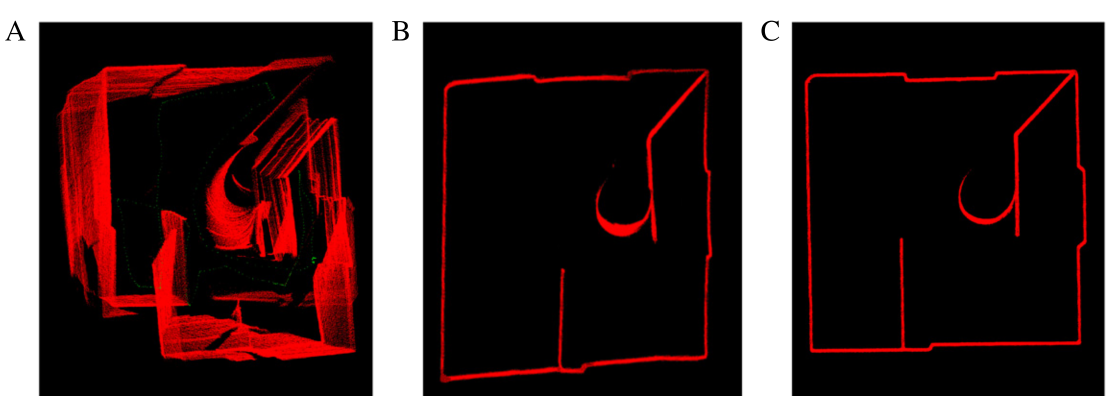



**Fig. 11.** Map obtained by odometry and mentioned algorithms.

According to Table 2 the improvement percentage of HIDE compared to HSL is in the range of −2.06%–13.09%. The HIDE algorithm parameters for which we have found the best results in our experiments are as follows.

NFE $= 220$, HMS $= 30$, HMCR $= 0.88$, PAR$_{min}$ $= 0.55$, PAR$_{max}$ $= 0.98$, HRT $= 14$, $Bw_{min} = 0.2$, $Bw_{max} = 8$, $\lambda = 0.7$ and $F = 0.44$. The position and orientation obtained for the robot with our presented algorithm is $(1.03, 0.99, 0.11)$ while the robot's true pose is $(0, 0, 0)$.

If we consider the effect of the sensor noise on the accuracy of the algorithm as shown in Fig. 9, we can notice that the presented algorithms are able to handle a high level of sensor noise with low dilapidation of the results. This behavior is due to the stochastic nature of the HS and DE algorithms. Fig. 10 shows the convergence speed of both HSL and HIDE localization algorithms. It is considerable that with improving the HM with the DE algorithm we are able to increase the convergence speed of the HSL algorithm. In Figs. 11 and 12(A)–(C), we can see the maps obtained by odometry, HSL and HIDE localization algorithms from scans taken in MRL by the mentioned robot respectively. Fig. 13(A)–(D) show area picture, map obtained by odometry, HSL and HIDE localization algorithms. In these maps the NFE in both HSL and HIDE algorithms are the same.

In Fig. 14, we can consider the map of Intel research lab, obtained with the HIDE localization algorithm.

## 7. Conclusion and future works

This paper introduced novel methods based on the harmony search algorithm for mobile robot localization. Due to the stochastic nature of the HS algorithm, the proposed algorithm,

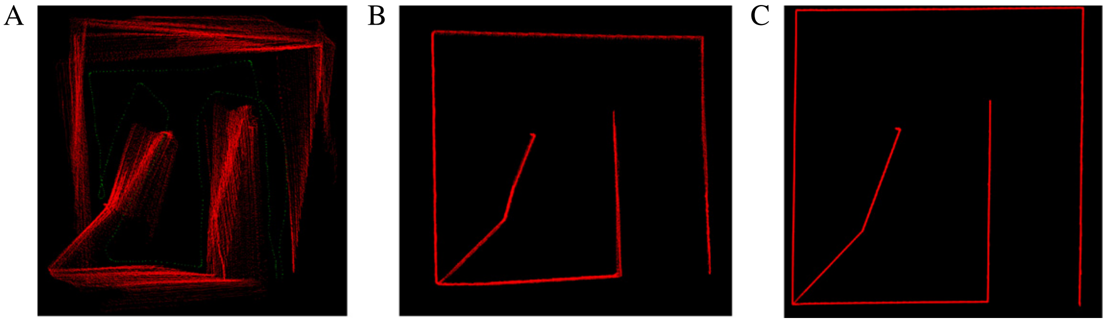



**Fig. 12.** Map obtained by odometry and mentioned algorithms.

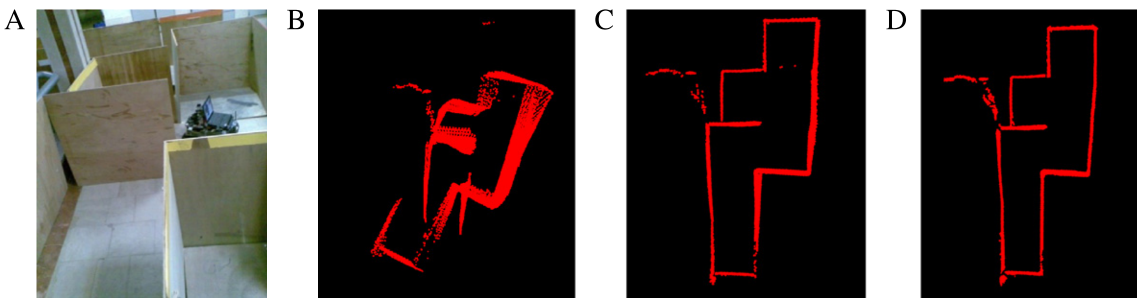
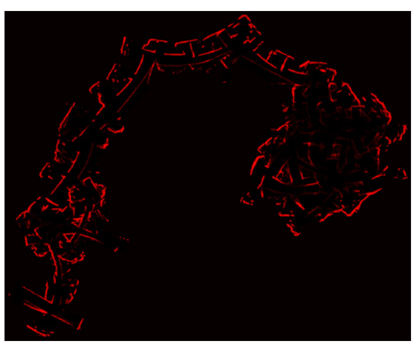
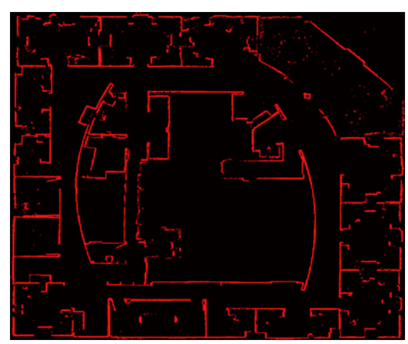



**Fig. 13.** Area picture and obtained maps.



(a) Map obtained by odometry.



(b) Map obtained by the HIDE algorithm.

**Fig. 14.** Intel research lab.

namely HSL, is able to handle a high level of sensor noise with low dilapidation of the results. The HSL algorithm is easy to implement, and the computational cost makes it able to operate online. The HSL method has shown its priority compared to the GLASM method and can guarantee safe navigation through environments without a prior map. Although HSL is faster and more accurate than the GLASM algorithm, the HS algorithm sometimes suffers from slow convergence speed. So another hybrid algorithm merging DE into HS is proposed. Experimental results have shown that the hybrid algorithm benefits from the convergence speed of the DE method and the searching quality of the HS algorithm that makes it fairly effective for mobile robot localization. In future work we will investigate the possibilities to acquire a complete and robust localization and mapping solution by embedding the HIDE localization in a SLAM framework by boosting the performance of the fitness function.

## Acknowledgment

The authors are indebted to Belmond Yoberd for his literary contribution.

## References

[1] P.J. Besl, N.D. McKay, A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 239–256.
[2] L. Silva, O. Bellon, K. Boyer, Enhanced robust genetic algorithms for multiview range image registration, in: Proceeding of International Conferenceon 3D Digital Image and Modeling, 2003.
[3] E. Lomonosov, D. Chetverikov, A. Ekrt, Pre-registration of arbitrarily oriented 3D surfaces using a genetic algorithm, in: Evolutionary Computer Vision and Image Understanding, Pattern Recognition Letters 27 (2006) 1201–1208 (special issue).
[4] F. Lu, E. Milios, Robot pose estimation in unknown environment by matching 2D range scans, Journal of Intelligent and Robotic Systems 18 (1997) 249–285.
[5] H. Yoshitaka, Mobile robot localization and mapping by scan matching using laser reflection intensity of the sokuiki sensor, in: IEEE 32nd International Conference on Industrial Electronics, 2006.
[6] J.L. Martinez, J. Gonzalez, J. Morales, A. Mandow, A.J. Garcia-Cerezo, Mobile robot motion estimation by 2D scan matching with genetic and iterative closest point algorithms, Journal of Field Robotics 23 (1) (2006) 21–34.
[7] L. Moreno, J.M. Armingol, S. Garrido, A.D.L. Escalera, M.A. Salichs, A genetic algorithm for mobile robot localization using ultrasonic sensors, Journal of Intelligent and Robotic Systems 34 (2002) 135–154.
[8] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing: quantitative studies, sciences, Journal of Statistical Physics 220 (4598) (1983) 671–680.
[9] T. Ducket, A Genetic algorithm for simultaneous localization and mapping, in: IEEE International Conference on Robotics and Automation, 1, 2003, pp. 434–439.

[10] L. Moreno, S. Garrido, D. Blanco, Mobile robot global localization using an evolutionary map filter, ACM Journal of Global Optimization 37 (3) (2007) 381–403.
[11] K. Arras, J.A. Castellanos, R. Siegwart, Feature-based multi-hypothesis localization and tracking for mobile robots using geometric constraints, in: Proceeding of the International Conference on Robotics and Automation ICRA-02, Washington DC, USA, 2002.
[12] D.J. Austin, P. Jensfelt, Using multiple Gaussian hypotheses to represent probability distributions for mobile robot localization, in: Proceeding of the International Conference on Robotics and Automation ICRA-00, San Francisco, CA, USA, 2000.
[13] P. Jensfelt, S. Kristensen, Active global localisation for a mobile robot using multiple hypothesis tracking, in: Proceeding of IJCAI Workshop on Reasoning with Uncertainty in Robot Navigation, Stockholm, Sweden, 1999.
[14] I.J. Cox, J.J. Leonard, Modelling a dynamic environment using a Bayesian multi hypothesis approach, Artificial Intelligence 66 (1994) 311–344.
[15] S.I. Roumeliotis, G.A. Bekey, Bayesian estimation and Kalman filtering: a unified framework for mobile robot localization, in: Proceeding of IEEE International Conference on Robotics and Automation, ICRA-2000, San Francisco, 2000.
[16] D. Fox, W. Burgard, S. Thrun, Markov localization for mobile robots in dynamic environments, Journal of Artificial Intelligence Research 11 (1999) 391–427.
[17] W. Burgard, D. Fox, D. Henning, T. Schmidt, Estimating the absolute position of a mobile robot using position probability grids, in: Proceeding of the National Conference on Artificial Intelligence AAAI-96, Portland, Oregon, USA, 1996.
[18] J. Reuter, Mobile robot self-localization using PDAB, in: Proceeding of IEEE International Conference on Robotics and Automation, ICRA-2000, San Francisco, CA, 2000.
[19] P. Jensfelt, O. Wijk, D.J. Austin, M. Andersson, Experiments on augmenting condensation for mobile robot localization, in: Proceeding of the International Conference on Robotics and Automation ICRA-00, San Francisco, CA, USA, 2000.
[20] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots, Artificial Intelligence 128 (2001) 99–141.
[21] F. Dellaert, D. Fox, W. Burgard, S. Thrun, Monte Carlo localization for mobile robots, in: Proceedings of International Conference on Robotics and Automation, 1999.
[22] J.L. Crowley, World modeling and position estimation for a mobile robot using ultrasonic ranging, in: Proceeding of the IEEE International Conferenceon Robotics& Automation, ICRA, 1989.
[23] J. Gonzalez, A. Ollero, A. Reina, Map building for a mobile robot equipped with a 2D laser rangefinder, in: Proceeding of IEEE International Conference on Robotics and Automation, ICRA, Spain, 1994.
[24] K.O. Arras, Y.S. Rolland, Feature extraction and scene interpretation for map-based navigation and map building, in: Proceedings of SPIE Mobile Robotics XII, 1997.
[25] J. Leonard, P. Newman, R. Rikoski, J. Neira, J. Tardós, Towards robust data association and feature modeling for concurrent mapping and localization, in: 10th Internation Symposium on Robotics Research, ISRR, 2001.
[26] D. Schroter, M. Beetz, J.S. Gutmann, RG mapping: learning compact and structured 2D line maps of indoor environments, in: Proceeding of the IEEE International Workshop on Robot and Human Interactive Communication, ROMAN'02, 2002.
[27] D. Sack, W. Burgard, A comparison of methods for line extraction from range data, in: Proceeding of the 5th IFAC Symposium on Intelligent Autonomous Vehicles IAV, 2004.
[28] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, S. Thrun, Using EM to learn 3D models with mobile robots, in: Proceeding of the International Conference on Machine Learning, ICML, 2001.
[29] D. Anguelov, D. Koller, E. Parker, S. Thrun, Detecting and modeling doors with mobile robots, in: Proceeding of the IEEE International Conferenceon Robotics & Automation ICRA, 2004.
[30] D. Burgard, W. Fox, H. Jans, C. Matenar, S. Thrun, Sonarbased mapping with mobile robots using EM, in: Proceeding of the 16th International Conference on Machine Learning, 1999.
[31] M. Bennewitz, W. Burgard, S. Thrun, Using EM to learn motion behaviors of persons with mobile robots, in: Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2002.
[32] M.W.M. Gamini Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, M. Corba, A solution to the simultaneous localization and map building(slam) problem, IEEE Transactions on Robotics and Automation 17 (3) (2001) 229–241.
[33] H.P. Moravec, Sensor fusion in certainty grids for mobile robots, AI Magazine (1988) 61–74.
[34] S. Thrun, D. Fox, W. Burgard, Probabilistic Robotics, MIT-Press, 2005, pp. 77–85.
[35] D. Hahnel, W. Bugard, D. Fox, S. Thrun, An efficient fastslam algorithm for generating maps of large-scale cyclic environment from raw laser range measurements, in: International Conference on Intelligent Robots and Systems, 2003.
[36] A. Eliazar, R. Parr, DP-SLAM: fast, robust simultaneous localization and mapping without predetermined landmarks, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2003.
[37] N.M. Kwok, G. Fang, W. Zhou, Evolutionary particle filter: re-sampling from the genetic algorithm perspective, in: Proceeding of IEEE International Conference on Intelligent Robots and Systems, 2005.
[38] N.J. Gordon, Non-linear/Non-Gaussian filtering and boostrap filter, in: Proceeding of IEEE Colloquium on Non-linear Filters, 1994.
[39] Z.W. Geem, J. Kim, G. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.
[40] R. Forsati, M. Shamsfard, P. Mojtahedpour, An dfficient meta heuristic algorithm for POS-tagging, in: 2010 Fifth International Multi-conference on Computing in the Global Information Technology, 2010.
[41] R. Forsati, M. Mahdavi, A.T. Haghighat, A. Ghariniyat, An efficient algorithm for bandwidth-delay constrained least cost multicast routing, in: Canadian Conference on Electrical and Computer Engineering, CCECE 2008, 2008, pp. 1641–1646.
[42] R. Forsati, M. Mahdavi, M. Shamsfard, M.R. Meybodi, Efficient stochastic algorithms for document clustering, Information Sciences (2013) http://dx.doi.org/10.1016/j.ins.2012.07.025.
[43] R. Forsati, M. Mahdavi, A.T. Haghigaht, Harmony search based algorithms for bandwidth-delay-constrained least-cost multicast routing, Computer Communications 31 (2008) 2505–2519.
[44] R. Forsati, M. Mahdavi, Web text mining using harmony search, in: Recent Advances in Harmony Search Algorithm, 2010, pp. 51–64.
[45] M. Mahdavi, M.H. Chehreghani, H. Abolhassani, R. Forsati, Novel meta-heuristic algorithms for clustering web documents, Applied Mathematics and Computation 201 (1–2) (2008) 441–451.
[46] M. Mahdavi, M. Fesanghasy, E. Damangir, An improved harmony search algorithm for solving optimization problems, Journal of Applied Mathematics 188 (2) (2007) 1567–1579.
[47] K. Lenac, Fast genetic scan matching using corresponding point measurements in mobile robotics, in: Springer Book of Application of Evolutionary Computing, vol. 4448, 2007, pp. 375–382.
[48] S. Das, A. Abraham, A. Konar, Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives, in: Studies in Computational Intelligence (SCI), vol. 116, 2008, pp. 1–38.
[49] R. Storn, K. Price, Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces, Journal of Global Optimization 11 (4) (1997) 341–359.
[50] U.K. Chakraborty, S. Das, A. Konar, DE with local neighborhood, in: Proceeding of Congress on Evolutionary Computation, CEC 2006, Vancouver BC, Canada, IEEE Press, New York, 2006.

**Mohsen Mirkhani** received a B.Sc. degree in Electronic Engineering in 2005 and M.Sc. degree in Mechatronic Engineering from Islamic Azad University of Qazvin, Iran, in 2009. His research interests are Mobile Robot Localization, Mobile Robot Routing, Soft Computing and Mechatronic Systems. He is also working on Real Rescue Robot at Mechatronic Research Laboratory (MRL) of the Islamic Azad University of Qazvin, Iran.

**Rana Forsati** is currently a Ph.D. student in Computer Engineering at Shahid Beheshti University, Tehran, Iran. Since 2009, she has been a faculty member at the Department of Computer Engineering, Islamic Azad University of Karaj, Iran. She received her B.Sc. and M.Sc. degrees in Computer Engineering from Qazvin University, Qazvin, Iran, in 2004 and 2008, respectively. Her current research interests include Soft Computing, Machine Learning, Web Mining, Semantic Web, Natural Language Processing, and Recommender Systems.

**Alireza Mohammad Shahri** received a B.S. degree in Electrical Engineering from K.N. Toosi University of Technology, Iran, in 1985. He then received his M.Eng. and Ph.D. degree in Electrical Engineering from University of Wollongong, Australia, in 1994 and 1998, respectively. His research interests are Instrumentation and Measurement Systems, Industrial Computer Networks, Mobile Robots, and Mechatronic Systems. Dr. Shahri is currently Faculty Member at the Department of Electrical Engineering at Iran University of Science and Technology. He is also conducting a few research programs on Mobile Robot, Flying Robot (Quadrotor) and Underwater Robot (ROV) at Mechatronic Research Laboratory (MRL) of the Qazvin Islamic Azad University.

**Alireza Moayedikia** received a B.S. degree in Computer Engineering from Islamic Azad University of Karaj in 2011 and his M.Sc. from Asia Pacific University of Malaysia and awarded Staffordshire University degree of UK in software engineering in 2013. His research interests are Data Mining, Evolutionary and Swarm based Algorithms, Machine Learning and Recommender Systems.