Machine Vision and Applications (2014) 25:405–419
DOI 10.1007/s00138-013-0515-z

FULL LENGTH PAPER

# Harmony search-based hybrid stable adaptive fuzzy tracking controllers for vision-based mobile robot navigation

**Kaushik Das Sharma · Amitava Chatterjee · Anjan Rakshit**

Received: 26 September 2012 / Accepted: 17 April 2013 / Published online: 3 May 2013


**Abstract** In this paper the harmony search (HS) algorithm and Lyapunov theory are hybridized together to design a stable adaptive fuzzy tracking control strategy for vision-based navigation of autonomous mobile robots. The proposed variant of HS algorithm, with complete dynamic harmony memory (named here as DyHS algorithm), is utilized to design two self-adaptive fuzzy controllers, for $x$-direction and $y$-direction movements of a mobile robot. These fuzzy controllers are optimized, both in their structures and free parameters, such that they can guarantee desired stability and simultaneously they can provide satisfactory tracking performance for the vision-based navigation of mobile robots. In addition, the concurrent and preferential combinations of global-search capability, utilizing DyHS algorithm, and Lyapunov theory-based local search method, are employed simultaneously to provide a high degree of automation in the controller design process. The proposed schemes have been implemented in both simulation and real-life experiments. The results demonstrate the usefulness of the proposed design strategy and shows overall comparable performances, when compared with two other competing stochastic optimization algorithms, namely, genetic algorithm and particle swarm optimization.

**Keywords** Harmony search algorithm · Autonomous mobile robot · Tracking control · Vision-based navigation · Adaptive fuzzy controller · Hybrid optimization approaches

K. Das Sharma (✉)
Department of Applied Physics, University of Calcutta, Kolkata 700009, India
e-mail: kdassharma@gmail.com

A. Chatterjee · A. Rakshit
Department of Electrical Engineering, Jadavpur University, Kolkata 700032, India

## 1 Introduction

Autonomous mobile robot navigation has been regarded as a popular research area for the last few decades. Some earlier attempts to solve the problem of path planning were presented in [1,2]. Various classical theoretical approaches such as Dijkstra's algorithm [3] and A* algorithm [4], designed originally for some other purposes, have also been suitably extended to solve navigation problems. The potential field method [5,6] and probabilistic roadmap methods [7], have been extensively utilized by many researchers for navigation in static environments only. On the contrary, for the moving obstacle environments, other potential solutions based on dynamic potential field method [8,9], and sensor-based path planning methods [10–12] have been extensively used. A robot typically calculates and estimates the motion of the moving obstacles based on sensory data form infra red (IR), sonar sensors, laser range finder, optical sensor and RFID [12,13]. The chief problem associated with most of the sensor-based techniques is that the data fusion algorithm, used to synchronize the sensory data, cannot process the data at the same time stamp due to the out-of-sequence measurements (OOSM) problem. In order to make precise navigation, this OOSM problem should be solved and thus, vision has become a popular alternative as a sensing mechanism [14]. In some vision-based approaches, the navigation system is used to construct a high-level topological representation of the world and the robot learns to recognize rooms or spaces and to navigate between them by building models of those spaces and their connections [14,15]. Many robot navigation systems also focus on producing a detailed metric map of the world using expensive hardware, such as a laser scanner, telefocus camera, etc. After this map is built, the robot then has to solve a complicated path planning algorithm [16,46]. These methods pose significant, additional computational burden to the processing unit and create difficulties during real-life applications.

The present paper uses a novel idea of formulating a mobile robot navigation problem as a tracking control problem, where vision is used as the sensing element for navigation [17]. The problem has been solved using hybrid stable adaptive fuzzy controllers. Fuzzy systems and fuzzy control philosophies have been successfully employed in many problems in the area of robotics [18–22] and other interesting engineering applications [23–26]. In this present scheme, single camera-based vision is used to generate the reference path and adaptive state feedback fuzzy controllers are utilized to track that path. A simple path planning algorithm is devised to create the reference path for the mobile robot navigation purpose. The planned path is then utilized to create the reference signals for $x$-direction and $y$-direction movements and two previously tuned adaptive fuzzy controllers are implemented, for $x$-direction and $y$-direction movement of the mobile robot. For the actual navigation of the mobile robot following the planned path, the controllers have to generate suitable control signals to produce the requisite drive signals for left and right wheel actuators of the robot. The major contribution of this work is that each tracking controller is separately tuned with some arbitrary reference signal, generated from a similar environment to that in which the robot will actually navigate, and, once the tuning part is successfully concluded, these controllers are readily applied to the real-life applications. The system is so designed that once the reference path is generated, the robot is commanded to track a specified fraction of this reference path. Then a new image of the environment is acquired and the whole process of new path planning, control actuation generation and robot navigation for a specified fraction of the reference path is repeated again and again. This is done with the objective of equipping the system with the flexibility of handling dynamically varying environments, so that the robot navigation can still be carried out with dynamic variations in positions of obstacles during its navigation.

In the present paper, a hybridization of locally operative Lyapunov theory and a globally adaptive harmony search algorithm-based stochastic approach [27,28] has been utilized to design the hybrid stable adaptive state feedback fuzzy controllers for autonomous mobile robot navigation, utilizing the concept of tracking control. The relationship between music and mathematics dates back to ancient civilization. In 2001, Z. W. Geem developed an optimization method, namely harmony search, for function optimization and engineering applications, inspired by the musical phenomena of harmony improvisations. It is a meta-heuristic algorithm based on music [29]. Many more meta-heuristics are found in literature, most of them mimicking some natural or artificial phenomena. The most commonly used class of algorithms is that of genetic algorithms (GAs), which are based on natural selection and mechanism of population genetics [30]. Other such computational algorithms, similar to GA, include evolutionary strategies, genetic programming, particle swarm optimization, ant colony optimization, simulated annealing and tabu search [31,32]. In contrast to the other metaheuristic search techniques, HS algorithm is based on the musical process of searching for perfect state of harmony, utilizing the esthetic and acoustic criteria, which impose fewer mathematical requirements [29,33] and can be easily adopted for various types of engineering optimization problems. In recent years, HS method has been successfully applied in several fields including function optimization [34,35] mechanical structure design [34,36], and pipe network optimization [37]. Some exciting applications of HS algorithm, specifically in the domain of robotics, have also been recently reported in [38], where Tangpattanakul et al. proposed a method for optimal trajectory planning of robot manipulators with six degrees of freedom. In their work, Tangpattanakul et al. proposed a hybrid methodology, combining the HS and the sequential quadratic programming (SQP), to develop a globally optimal trajectory for a manipulator movement. The works of Fourie et al. are also included in [38], where a visual tracking system has been developed employing a proposed formulation of the HS algorithm-based harmony filter. The method has been successfully applied to accurately track a poorly modeled target under challenging environments.

The Lyapunov theory and harmony search algorithm have been hybridized to create a superior method by combining the strong points of both the methods. In our proposed hybrid methodologies, the Lyapunov theory-based adaptation and HS algorithm-based stochastic global search are operated concurrently or preferentially, to optimize both the structure and the free parameters of the fuzzy logic controllers (FLCs) over the solution space, with the objective of controlling the plant. Vision-based path planning and the design of tracking controllers for controlling the $x$-direction and $y$-direction movements are performed with the objective of autonomous mobile robot navigation without using any IR or other proximity sensors. In this design methodology, the number of membership functions (MFs) for each input variable, the supports of input MFs, the number of rules, the values of the scaling gains and the positions of output singletons are optimized to evaluate a candidate controller setting for optimal performance, in terms of the integral absolute error (IAE) value between the path planned using vision sensor and the actual path traversed by the mobile robot. The obtained results illustrate, on the whole, the superiority of the preferential hybrid approach over the other approaches. The results obtained in both simulation and experimental environments are compared with similar strategies developed using GA- and PSO-based fuzzy logic tracking controllers for vision-based robot navigation [17].

The rest of the paper is organized as follows: Sect. 2 details the design of stable adaptive fuzzy controllers. Section 3 describes the HS algorithm-based tracking controller design technique as well as the hybrid controller design strategies. Section 4 discusses in detail the implementation of the proposed stable adaptive fuzzy tracking controller-based mobile robot navigation scheme in an indigenously developed mobile robot platform and also the comparative study among different design strategies. Section 5 concludes the paper.

## 2 Design of stable adaptive fuzzy tracking controller

Let us consider the motion of the mobile robot in a plane which can be described as [39]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \underline{\dot{p}} = \begin{bmatrix} \cos\varphi & 0 \\ \sin\varphi & 0 \\ 0 & 1 \end{bmatrix} \underline{m} \tag{1}$$

$$M(\underline{p})\underline{\dot{m}} + V(\underline{p}, \underline{\dot{p}})\underline{m} + G(\underline{p}) = \tau \tag{2}$$

where $\underline{p} = \begin{bmatrix} x & y & \varphi \end{bmatrix}^{\mathrm{T}}$ is the present pose of the robot in the world Cartesian coordinate system, the coordinate $(x, y)$ denotes the centre of mass of the robot, the heading direction $\varphi$ is taken counterclockwise from the $x$-axis, $\underline{m} = \begin{bmatrix} v & \omega \end{bmatrix}^{\mathrm{T}}$ is the vector containing linear velocity $v$ and angular velocity $\omega$, $\tau \in \Re^{n\times 1}$, is the input vector, $M(\underline{p}) \in \Re^{n\times n}$ is a symmetric and positive definite inertia matrix, $V(\underline{p}, \underline{\dot{p}}) \in \Re^{n\times n}$ is the centripetal and Coriolis matrix, and $G(\underline{p}) \in \Re^{n\times 1}$ is the gravitational vector.

In this present work, our control objective is that for a given reference trajectory $\underline{p}_r(t)$ and orientation of mobile robot, we must design two controllers that can generate suitable torque $\tau$ such that the current robot position $\underline{p}_c(t)$ achieves the desired reference position $\underline{p}_r(t)$:

$$\underset{t\to\infty}{Lt}\,(\underline{p}_r(t) - \underline{p}_c(t)) = 0 \tag{3}$$

To achieve the control objective, the system should be so designed that a $\tau(t)$ is derived based on the differentially steered drive velocities to the left and right wheels of the robot according to the steering system in (1). This objective is satisfied by designing two stable adaptive fuzzy logic controllers, for $x$-direction and $y$-direction movements, that can ensure asymptotic stability, achieve satisfactory transient performance and can suitably configure this mobile robot navigation problem as a tracking control problem, where the controllers guide the robot to track or navigate a desired tracking or navigation path.

Now to design such $x$-direction and $y$-direction tracking controllers, let us consider that our objective is to design an adaptive strategy for an $n$th order SISO plant in general, and is given as [27,40–42]:

$$\begin{cases} q^{(n)} = f(\underline{q}) + bu \\ r = q \end{cases} \tag{4}$$

Here, the state vector is given as $\underline{q} = (q_1, q_2, \ldots, q_n)^{\mathrm{T}} = (q, \dot{q}, \ldots, q^{(n-1)})^{\mathrm{T}} \in R^n$ where $q^{(n)}$ denotes the $n$th derivative of state $q$, $q_1$ is the first state where $q_1 = q$, both plant input and plant output are scalar quantities, given as $u \in R$ and $r \in R$, respectively, and $f(\cdot)$ is an unknown continuous function. The control objective is to force the plant output $r(t)$ to follow a given bounded reference signal $r_m(t)$ under the constraints that all closed-loop variables involved must be bounded to guarantee the closed loop stability of the system. Thus, the tracking error is $e = r_m - r$. Let the parameter vector of the output singletons of the fuzzy controller be denoted as $\underline{\theta}$. Hence our control objective is to find the structure of the fuzzy controller as well as a feedback control strategy $u = u(\underline{q}|\underline{\theta})$, using fuzzy logic system and an adaptive law for adjusting $\underline{\theta}$ such that the following conditions are satisfied:

(i) The closed loop system must be globally stable in the sense that all variables, $\underline{q}(t)$, $\underline{\theta}(t)$ and $u(\underline{q}|\underline{\theta})$ must be uniformly bounded, and
(ii) The tracking error $e(t)$ should be as small as possible under the constraints in i) and the $\underline{e} - \underline{\theta}$ space should be stable in the large for the system [40,41].

To accomplish these control objectives with a direct adaptive fuzzy controller, let the error vector be $\underline{e} = (e, \dot{e}, \ldots, e^{(n-1)})^{\mathrm{T}}$ and $\underline{k} = (k_1, k_2, \ldots, k_n)^{\mathrm{T}} \in R^n$ be such that all the roots of the Hurwitz polynomial $s^n + k_n s^{n-1} + \cdots + k_2 s + k_1$ are in the open left half of s-plane [40–42]. Now the ideal control law for the system in (4) is given as [40,43]:

$$u^* = \frac{1}{b}\left[-f(\underline{q}) + r_m^{(n)} + \underline{k}^{\mathrm{T}}\underline{e}\right] \tag{5}$$

where $r_m^{(n)}$ is the $n$th derivative of the output of the reference model.

This definition implies that $u^*$ guarantees perfect tracking, i.e. $r(t) \equiv r_m(t)$ if $\mathrm{Lim}_{t\to\infty}\, e(t) = 0$ [41,42]. Here the vector $\underline{k}$ describes the desired closed-loop dynamics for the error. In practical situations, since $f$ and $b$ are not known precisely, the ideal $u^*$ of (5) cannot be implemented in practice. Thus a suitable solution can be to design a fuzzy logic system to approximate this optimal control.

Now, to ensure stability, we assume that the control $u(t)$ is given by the summation of a fuzzy control, $u_c(\underline{q}|\underline{\theta})$, and an additional supervisory control strategy, $u_s(\underline{q})$, given as:

$$u(t) = u_c(\underline{q}|\underline{\theta}) + u_s(\underline{q}) \tag{6}$$

Let us assume that the adaptive fuzzy logic controller (AFLC) is constructed using a zero order Takagi–Sugeno (TS) fuzzy system. Then $u_c(\underline{q}|\underline{\theta})$ for the AFLC is given in the form [27,40,42]

$$u_c(\underline{q}|\underline{\theta}) = \frac{\sum_{l=1}^{N} \theta_l * \alpha_l(\underline{q})}{\sum_{l=1}^{N} \alpha_l(\underline{q})} = \underline{\theta}^{\mathrm{T}} * \xi(\underline{q}) \tag{7}$$

where $\underline{\theta} = [\theta_1\, \theta_2 \ldots \theta_N]^{\mathrm{T}}$ is the vector of the output singletons, $\alpha_l(\underline{q})$ is the firing degree of rule '$l$' $= \prod_{i=1}^{r} \mu_i^l(q_i)$, $N$ is the total number of rules, $\mu_i^l(q_i)$ is the membership value of the $i$th input membership function (MF) in the activated $l$th rule, $\underline{\xi}(\underline{q})$ is the vector containing normalized firing strength of all fuzzy IF-THEN rules $= (\xi_1(\underline{q}), \xi_2(\underline{q}), \ldots, \xi_N(\underline{q}))^{\mathrm{T}}$ and $\xi_l(\underline{q}) = \frac{\alpha_l(\underline{q})}{\sum_{l=1}^{N} \alpha_l(\underline{q})}$.

Let us define a quadratic form of tracking error as $V_e = \frac{1}{2}\underline{e}^{\mathrm{T}} P \underline{e}$ where $P$ is a symmetric positive definite matrix satisfying the Lyapunov equation. The supervisory control action, $u_s(\underline{q})$, should be so designed that $\dot{V}_e$ should be negative semi-definite, i.e. $\dot{V}_e \leq 0$. Here $u_s(\underline{q})$ is constructed as given in [40,43] and can be presented as

$$u_s(\underline{q}) = I_1^* \mathrm{sgn}\left(\underline{e}^{\mathrm{T}} P \underline{b}_c\right)\left[|u_c| + \frac{1}{b_{\mathrm{L}}}\left(f^U + \left|r_m^{(n)}\right| + |\underline{k}^{\mathrm{T}}\underline{e}|\right)\right] \tag{8}$$

where $\begin{cases} I_1^* = 1 \text{ if } V_e > \overline{V} \\ I_1^* = 0 \text{ if } V_e \leq \overline{V} \end{cases}$, $\overline{V}$ is a constant specified by the designer, $f^U \geq |f(\underline{q})|$ and $0 < b_{\mathrm{L}} < b$. With this $u_s(\underline{q})$ it can be shown that $\dot{V}_e \leq -\frac{1}{2}\underline{e}^{\mathrm{T}} Q \underline{e} \leq 0$, where $Q$ is a positive definite matrix. Thus, as $P > 0$, boundedness of $V_e$ implies the boundedness in $\underline{q}$. Hence, the closed loop stability is guaranteed.

The zero order TS-type fuzzy control $u_c(\underline{q}|\underline{\theta})$ can be so constructed that it will produce a linear weighted combination of adapted parameter vector $\underline{\theta}$. Thus a simple singleton-based adaptation law as proposed in [40–43], can be given as

$$\underline{\dot{\theta}} = \nu \underline{e}^{\mathrm{T}} \underline{p}_n \underline{\xi}(\underline{q}) \tag{9}$$

where $\nu > 0$ is the adaptation gain or learning rate and $\underline{p}_n$ is the last column of $P$.

## 3 Design of harmony search algorithm-based AFLC

The HS algorithm is inspired by musicians' behavior where a combination of pitches determine the quality of harmony generated, here a set of values of the decision variables is judged by the corresponding value of the objective function. As in musical harmony, if a combination of decision variables can produce good result, then this solution vector is stored in memory and this helps to produce an improved solution in near future.

### 3.1 Harmony search-based approach (HSBA) [29,33]

In Lyapunov theory-based design of stable AFLC's [40], only the output singletons of the fuzzy controllers were adapted and the other free parameters e.g. supports of the input MFs, input and output scaling gains and also the structural parameters of the controllers, i.e. the number of MFs in which each input is fuzzified, the number of rules, etc. were chosen a priori. Furthermore, the AFLC was implemented for a fixed controller structure. However, while designing the HS algorithm-based optimal controller, all these parameters are obtained automatically by encoding them as part of the solution vector, i.e. each of these parameters forms a decision variable. This helps to determine the optimal structure of the FLC as well as its complete settings.

In HS algorithm-based design, a harmony in solution space is formed as [27,28]:

$$\begin{aligned} \mathrm{Z} = [&\text{structural flags for MFs} \mid \text{center locations of the MFs} \mid \\ &\ldots \mid \text{scaling gains} \mid \text{positions of the output singletons}] \end{aligned} \tag{10}$$

The structural flags are implemented to determine whether a particular MF will be present for an input to the FLC or not. It can only take binary values where 0 indicates the non-existence and 1 indicates the existence of the corresponding MF. However, HS is an algorithm where each entry in the harmony vector can take continuous values. Hence, for each structural flag in the vector, the universe of discourse is set as [0, 1] and the flag is set to 0 if the continuous value of the variable is $<0.5$ and the flag is set to 1 if the variable is $\geq 0.5$. The total number of 1s in the structural flags keeps changing in each iteration and hence the total number of MFs in which an input variable is fuzzified also changes in each iteration. This changes the structure of the AFLC in each iteration as it changes the total number of MFs for each input variable, which changes the total number of rules of the fuzzy rule base and hence changes the total number of active output singletons. Each input is fuzzified in the range $[-1, 1]$, with two fixed triangular MFs having their peaks fixed at $-1$ and 1 respectively and the centre locations of all other triangular MFs are stored in the particle vector. All intermediate MFs for that input variable are flexible in nature. They can be either active or inactive during an iteration and their peaks are also adapted by HS algorithm in each iteration. For each MF the peaks of the immediate adjacent active MFs on either side of its own peak forms the left and right base support of it. Whether the immediate adjacent MF is an active MF or not is determined by the content of its corresponding structural flag. Hence, some of the centre locations of MFs are ignored while evaluating the AFLC output, because their corresponding entries in the structural flags are zero. A similar logic holds true for the output singletons in a harmony vector. If a singleton has one or more antecedent MF that is inactive, then it becomes inactive itself.

***HSBA Algorithm***

The outline of the HS [29] based controller design algorithm employed in this paper is given as follows [27]

(i) The optimization problem can be stated as:

$$\text{Minimize}\, g(\underline{Z}) \text{such that} z_i \in \hat{Z}_i, \quad i = 1, 2, \ldots, d \tag{11}$$

$\underline{Z}$ is a candidate solution vector, $\underline{Z} = [z_1, z_2, \ldots, z_d]^{\mathrm{T}} \in R^d$ and $\hat{Z}_i$ is the universe of discourse of $z_i$ and $g(\underline{Z}) =$ integral absolute error (IAE) $= \sum_{n=0}^{\mathrm{PST}} e(n)\Delta t_c$, where PST is the plant simulation time and $\Delta t_c =$ step size or sampling time.

(ii) Generate HMS number of randomly generated candidate solution vectors and store them in the HM matrix (stored by the values of the objective function $g(\underline{Z})$):

$$HM = \begin{bmatrix} z_1^1 & z_2^1 & \cdots & z_d^1 \\ z_1^2 & z_2^2 & \cdots & z_d^2 \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{\mathrm{HMS}} & z_2^{\mathrm{HMS}} & \cdots & z_d^{\mathrm{HMS}} \end{bmatrix} \tag{12}$$

(iii) Determine the structure of the candidate controller based on structural flags as incorporated in the harmony $\underline{Z}$.

(iv) Calculate the fitness value (e.g. integral absolute error (IAE)) of the controller for each harmony that acts as a candidate controller in this work, by using candidate controller simulation (CCS) algorithm as shown in Algorithm 1.

(v) Improvise a new harmony vector $\underline{Z}' = \left(z_1', z_2', \ldots, z_d'\right)$ either by choosing a value for a decision variable from the past history, stored in HM matrix, or by choosing any value in its permissible universe of discourse, with probability HMCR $\in$ (0, 1) [33,34,44]. Hence a new decision variable $z_j'$ in $\underline{Z}'$ can be determined as:

$$\left.\begin{array}{ll} z_j' \in \left\{z_j^1, z_j^2, \ldots, z_j^{\mathrm{HMS}}\right\} & \text{with probability } HMCR \\ \text{or} & \\ z_j' \in \hat{Z}_j & \text{with probability } (1 - HMCR) \end{array}\right\} \tag{13}$$

Each decision variable of the new vector $\underline{Z}'$, if generated from HM matrix, is further examined for a potential pitch-adjustment possibility with a probability PAR. In pitch-adjustment, the decision variable adjusts its value according to the formula:

$$z_j' = z_j' + \mathrm{bw} \times ud(-1, +1) \tag{14}$$

where bw is an arbitrary distance bandwidth and *ud* is a uniform distribution.

(vi) Evaluate the objective function $g(\underline{Z}')$ for this new harmony. If $g(\underline{Z}') = \min\left(g(\underline{Z}^1), g(\underline{Z}^2), \ldots, g(\underline{Z}^{\mathrm{HMS}})\right)$, then replace the solution vector in existing HM matrix that produced the worst harmony, with this new harmony. Then rearrange the HM matrix by sorting all decision vectors according to their objective function values.

(vii) The termination criterion can be set in two ways, either by choosing the maximum number of HS improvisation/iteration ($\mathrm{iter}_{\max}$) or by using a minimum error criteria, i.e. continue the HS improvisation until a maximum pre-specified allowable error is attained by the controller.

Further, in 2007 Mahdavi et al. [34] suggested an improvement to the original HS algorithm as presented in [29], where the PAR and bw in step-v are varied throughout the generations as linearly increasing and logarithmically decreasing manner respectively, but the HMCR parameter is kept constant. In [34] the values of PAR and bw change dynamically with the generation of harmony improvisation as:

$$PAR(g) = \mathrm{PAR}_{\min} + \frac{(\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min})}{\mathrm{iter}_{\max}} \times g \tag{15}$$

where $\mathrm{PAR}_{\min}$ is minimum pitch adjusting rate, $\mathrm{PAR}_{\max}$ is the maximum pitch adjusting rate, $g$ is the present generation number, and

$$\begin{aligned} \mathrm{bw}(g) &= \mathrm{bw}_{\max} \exp(c \cdot g), \\ c &= \frac{\mathrm{Ln}\left(\frac{\mathrm{bw}_{\min}}{\mathrm{bw}_{\max}}\right)}{\mathrm{iter}_{\max}} \end{aligned} \tag{16}$$

where $\mathrm{bw}_{\min}$ is the minimum bandwidth, $\mathrm{bw}_{\max}$ is the maximum bandwidth.

**BEGIN**
1. Initialize plant simulation time (*PST*), step size ($\Delta t$) of the computation, reference trajectory.
2. **WHILE** (*iter_count* < *PST* / $\Delta t$)
    2.1. *iter_count* = *iter_count* + 1.
    2.2. Fuzzify each input variable using structural flags present in a harmony.
    2.3. Calculate the membership values of each input MF's ( $\mu(\underline{q})$ ).
    2.4. Calculate the fuzzy basis function ( $\xi(\underline{q})$ ).
    2.5. Calculate the control signal $u_c(\underline{q}|\underline{\theta})$ from FLC
    2.6. Calculate the supervisory control $u_s(\underline{q})$ and calculate the control signal to the plant as $u(t) = u_c(\underline{q}|\underline{\theta}) + u_s(\underline{q})$.
    2.7. Calculate the plant's dynamical behavior using 4th order Runge – Kutta Method.
**ENDWHILE**
3. Calculate integral absolute error (IAE).
**END**

Algorithm 1. Candidate controller simulation (CCS) algorithm.

This modification showed successful results for several constrained and unconstrained optimization problems.

### 3.2 Newly proposed completely dynamic harmony memory improvisation-based HSBA (DYHSBA) [27]

Another modification of the original HS algorithm has been proposed in [27], where HMCR and PAR parameters are varied simultaneously but the bw parameter is kept constant throughout the generation of harmony improvisations to obtain the optimal solution vector. In this proposed modification of the harmony improvisation of the basic HS algorithm, the parameters HMCR and PAR both are increased linearly from $\mathrm{HMCR}_{\min}$ to $\mathrm{HMCR}_{\max}$ and $\mathrm{PAR}_{\min}$ to $\mathrm{PAR}_{\max}$ respectively and bw is kept fixed as:

$$\begin{cases} \mathrm{HMCR}(g) = \mathrm{HMCR}_{\min} + \frac{(\mathrm{HMCR}_{\max} - \mathrm{HMCR}_{\min})}{\mathrm{iter}_{\max}} \times g \\ \mathrm{PAR}(g) = \mathrm{PAR}_{\min} + \frac{(\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min})}{\mathrm{iter}_{\max}} \times g \\ \text{bw: fixed} \end{cases} \tag{17}$$

The physical implication of this modification is that as the generation of harmony improvisation increases the optimization approach relies more on the harmony memory because the harmony memory would be better one due to the improvisations throughout the generations. The PAR parameter is also increased because the fine tuning is required as the optimization algorithm relies more on the past values of the harmonies stored in the memory. That signifies the modifications of the HMCR and PAR parameter of the basic HS algorithm.

Along with this modification, another modification in the improvisation stage is also proposed in [27]. In the original HS algorithm during the harmony improvisation, only one harmony is improvised in each iteration and consequently updates the harmony memory if the newly improvised harmony performs better in terms of the objective function. Thus only one harmony is replaced and remaining (HMS - 1) number of harmonies remain unaffected in each iteration. Instead of using such a concept of harmony memory update, if the entire harmony memory is improvised in each iteration then it can be hoped that the convergence of the algorithm will be better as the algorithm will then potentially explore more solution space in each iteration. Therefore, in this modification, the harmony memory is improvised HMS number of times rather than single harmony improvisation in each iteration. It is also reported in [27] that the proposed modification with entire HM improvisation concept shows a superior result than the other HS variants. Therefore, from now onwards this modification is implemented for the design of different hybrid adaptation-based control strategies.

### 3.3 Design of hybrid stable AFLC by HSBA and Lyapunov theory

In this design methodology, the Lyapunov theory-based local adaptation of output singletons and the modified HS algorithm-based global optimization technique (DyHSBA) are combined to design stable adaptive fuzzy controllers. In this paper this approach is referred as the hybrid adaptation strategy-based approach (HASBA). Two different modes of hybrid models such as concurrent and preferential, are proposed and developed to design the stable adaptive fuzzy controllers for vision-based mobile robot navigation purpose. These two hybrid techniques are described in the following sections.

#### *3.3.1 Hybrid concurrent model (HASBA-Con)*

In this design process, the Lyapunov theory-based adaptation and the DyHSBA run concurrently to optimize the structure of the controller, i.e. the number of MFs required to fuzzify the input variables, center locations of the input MFs, positions of the output singletons etc. and the scaling gains of the input and output variables.

In this method also, a solution vector $\underline{Z}$ is divided into two sub-groups given as [27,42,43]:

$$\underline{Z} = [\underline{\psi} \mid \underline{\theta}] \tag{18}$$

where

$$\left.\begin{aligned} &\underline{\psi} = [\text{structural flags for MFs} \mid \text{center locations} \\ &\text{of the MFs} \mid \text{scaling gains}] \\ &\underline{\theta} = [\text{position of the output singletons}] \end{aligned}\right\} \tag{19}$$

The decision variables comprising $\underline{\psi}$ have a non-linear influence and the decision variables comprising $\underline{\theta}$ have a linear influence on the control signal $u_c$ [27,42]. For each harmony chosen as a candidate controller, it is first subjected to the Lyapunov theory-based adaptation of the $\underline{\theta}$ portion of the chosen harmony and the finally adapted values are then used to update the candidate controller. Then this updated harmony is subjected to a usual pass of the DyHSBA algorithm. In this process $\underline{\theta}$ is subjected to both local and global search experiences in every improvisation of a harmony. This process of concurrently employing the Lyapunov theory-based adaptation and DyHSBA algorithm for each candidate controller is continued in quest for the best solution.

#### *3.3.2 Hybrid preferential model (HASBA-Pref)*

In this proposed hybridization scheme, the DyHSBA method and the HASBA-Con method are both used to achieve the minimum possible tracking error. Here, each harmony in the harmony memory, i.e. each candidate controller, evaluates the CCS algorithm and also the Lyapunov theory-based adaptation separately. Then, the algorithm for which the better fitness value is obtained is used as the guiding factor to explore the solution space. If the Lyapunov theory-based adaptation is better for that candidate solution vector in that iteration, then the process will follow the concurrent hybrid model, otherwise, the process will follow the DyHSBA method in that iteration. Hence each harmony in the HM matrix is subjected to different guiding rules of improvisation and the preference, or the choice of the guiding rule, is based on the evaluation process just described before. This process of evaluation is performed in each iteration afresh for each harmony and the whole harmony memory is improvised to get an improved convergence in the design process.

## 4 Vision-based mobile robot navigation using stable AFLC's

### 4.1 Mobile robot hardware description

A differentially steered drive robot, indigenously designed and built in our laboratory, is used as our experimental platform as shown in Fig. 1. The proposed algorithms are implemented in Visual Basic on an Asus EEE PC Laptop, with 1GB solid-state HD, 1GB RAM, Windows XP operating system, mounted on-board [45]. The Laptop communicates with the robot base which comprises a PIC18F4550 processor, through USB link. The robot has one wheel on each front side, connected with RC servo motors, and two rear wheels, which are not motorized. The robot is equipped with encoder feedback from the left and right wheel encoders which generates four pulses/rotation and uses hall-effect switches in its operation. The Laptop camera with auto-focus facility serves as the mono-vision sensor. The robot base is energized (5V, 1A) from the Laptop through two USB cables. No separate power source is required for mobile robot operation and all RC servos are power controlled for energy saving. Six IR proximity sensors and an IR range sensor processor PIC12F683 are also attached to the robot base although they are not utilized in this work [45].

### 4.2 Image processing strategy and generation of the reference path

During the process of navigation, the robot intermittently stops, the camera acquires image of the environment in front of the robot, image processing is performed on that image to determine the reference path, the controllers determine the suitable drive commands and the robot navigates for a desired duration [17]. Then the robot stops again, acquires another image and repeats the process. This process of navigating the robot for a desired duration and then stopping it and activating the vision sensor is carried out in an iterative fashion, until the robot completes the navigation job. Algorithm 2 describes the image processing steps employed in this work which produces a binary image suitable for subsequent reference path planning [17]. Several path planning techniques have been reported over the last two decades [1–3,5–10,13–15,18,19,21–24,46,47]. In this work a simple but accurate path planning algorithm as proposed in [17] is utilized, where the path is planned in image plane and that path is subsequently project to real-world Cartesian coordinate system. A more detailed description of the algorithm for generation of the reference path is available in [17]. The algorithm is so developed that it is capable of coping with a dynamic environment. The acquisition of image and development of the reference path for an environment is not performed at a single time. This is because the environment may change after the robot starts navigation based on the generation of a reference path. To take this into account, 25 % of the reference path is used by the robot for immediate tracking purposes. Then the robot acquires a new image, generates a new reference path and again utilizes 25 % of it for further navigation. This process is continued in an iterative fashion, so that the robot can better cope with changing environments.

**BEGIN**
1. Acquire RGB image using Laptop camera.
2. Convert this RGB color image into a grayscale image.
3. Perform Gaussian blur filtering to de-noise this image.
4. Perform contrast adjustment to sharpen the distinguishable boundaries of the objects in the image.
5. Apply geometric mean filtering to efficient remove Gaussian type noise and preserve edge features.
6. Perform thresholding based segmentation on the processed image to generate binary image.

**END**

Algorithm 2. The image processing strategy employed .as pre-processing steps for generation of the reference path.

### 4.3 Image plane to real space conversion algorithm

After calculating the reference path in the image plane, i.e. in terms of image pixel positions, it is required to project these locations to the real world coordinates, where the robot will actually navigate. The coordinate transformation is calculated by measuring four distances as shown in Fig. 2. These include the height of the camera from the floor, $h$. It is to be noted that there is always a blind area immediately in front of the robot. Hence, distance of the nearest point that can be viewed by the camera is denoted as $y_b$. The furthest point visible to the camera at a distance is $(y_b + y_l)$ and the fourth distance is the furthest point to the left or right of the camera lateral view, denoted as $x_l$. With these measurements, the three transformation angles are calculated as:

$$\alpha = \tan^{-1}\left[\frac{h}{y_b}\right],$$
$$\beta = \tan^{-1}\left[\frac{h}{y_b+y_l}\right], \gamma = \tan^{-1}\left[\frac{x_l}{y_b}\right] \tag{20}$$

The point $P(x, y)$ in the world coordinate, which is transformed from the point $P'(u, v)$ in the image plane, is obtained as:

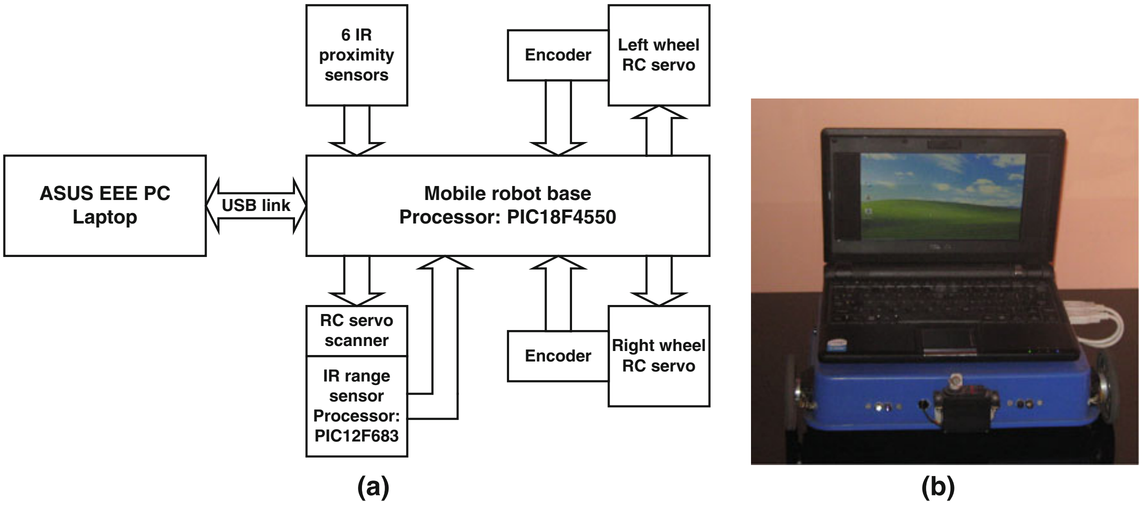


**Fig. 1** **a** Block diagram of internal circuitry of the indigenously developed mobile robot and **b** the actual robot built [45]

$$y = h\tan\left[\left(\frac{\pi}{2}-\alpha\right)+\left[\frac{u}{S_y}\right](\alpha-\beta)\right] \tag{21}$$

$$x = y\tan\left[\frac{2v}{S_x}\times\gamma\right] \tag{22}$$

where $S_x$ and $S_y$ are the number of image pixels in the vertical axis and horizontal axis respectively [48]. The transformations used in (21) and (22) are based on the assumption that each pixel subtends an equal angle of view.

### 4.4 Simulation case study

To evaluate the effectiveness of the proposed control strategies, first the autonomous mobile robot navigation schemes are developed in simulation. During simulation, fuzzy controllers are trained according to different control schemes as stated before. These trained controllers are implemented on-line to obtain their implementation performances in test conditions. In simulation study, the process model is simulated with a sampling time of 1 s., keeping the hardware constraints for the robot in real implementations, in mind.

The steering equations of the mobile robot as in (1) can be rewritten as:

$$\begin{aligned}\dot{x} &= v\cos\varphi = u_x\\ \dot{y} &= v\sin\varphi = u_y\\ \dot{\varphi} &= \omega\end{aligned} \tag{23}$$

Now, (23) is a special case of (4), where $f(\cdot) = 0$ and $b = 1$. $u = u_x$ is the control signal from $x$-direction tracking controller and $u = u_y$ is the control signal from $y$-direction tracking controller and $\varphi$ can be obtained from $\varphi(t) = \tan^{-1}\left[\frac{\dot{y}(t)}{\dot{x}(t)}\right]$. It is assumed that the reference point lies at the midpoint of the line joining the two drive wheels. Let $V_{\mathrm{L}}$ and $V_{\mathrm{R}}$ denote the velocities of the left and the right wheels respectively. The linear velocity $V$ and the orientation $\varphi$ of the mobile robot can be described as:

$$V = \sqrt{\left(u_x^2+u_y^2\right)}: \qquad \varphi = \tan^{-1}\left(\frac{u_y}{u_x}\right) \tag{24}$$

However, the theoretical expressions are suitably modified, to satisfy the hardware constraints of the robot system and to maintain robustness of the mechanical components of the robot developed for long-term operation. The $V_{\mathrm{L}}$ and $V_{\mathrm{R}}$, left and the right wheel velocities, respectively, are individually determined from the linear velocity $V$ and the orientation $\varphi$ according to the Algorithm 3 [17].

The magnitudes of $u_x$ and $u_y$ are so generated from the respective fuzzy controllers, that the linear velocity $V$ and the corresponding $V_{\mathrm{L}}$ and $V_{\mathrm{R}}$ will drive the robot to track the generated path by the vision sensor.

**BEGIN**
  **IF** $|\Delta\varphi| \le 10°$,
    **THEN** $V_L = V$; $V_R = V$;   % the robot will move straight
  **ELSE IF** $((\Delta\varphi > 10°)$ and $(\Delta\varphi < 90°))$,
    **THEN** $V_L = 0$; $V_R = V$;   % the robot will rotate left
  **ELSE IF** $((\Delta\varphi > -90°)$ and $(\Delta\varphi < -10°))$,
    **THEN** $V_L = V$; $V_R = 0$;   % the robot will rotate right
  **ENDIF**
**END**

Algorithm 3. Calculation of left and right wheel velocities [27].

The current position of the real robot can be determined form the position encoders as follows:

$$\begin{aligned}\delta_{\text{new}} &= \delta_{\text{old}} + (\Delta p_{\mathrm{L}} - \Delta p_{\mathrm{R}}) * 10°\\ x_{\text{new}} &= x_{\text{old}} + (\Delta p_{\mathrm{L}} + \Delta p_{\mathrm{R}})\cos(\delta_{\text{new}})\\ y_{\text{new}} &= y_{\text{old}} + (\Delta p_{\mathrm{L}} + \Delta p_{\mathrm{R}})\sin(\delta_{\text{new}})\end{aligned} \tag{25}$$

where $\Delta p_{\mathrm{L}}$ and $\Delta p_{\mathrm{R}}$ are the increments of the left and right encoders respectively and $\delta$ is the heading angle of the robot. $x_{\text{new}}$ and $y_{\text{new}}$ are used as inputs to the state feedback

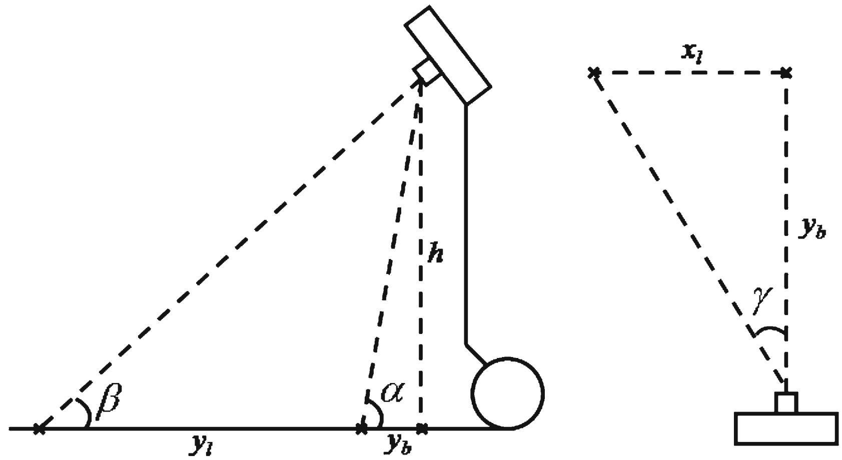


**Fig. 2** Relationship between the image coordinates and the mobile robot coordinates

**Table 1** Comparison of results of mobile robot navigation using DyHS-based different schemes

| Control Strategy | No. of MFs in $x$-dir controller | No. of MFs in $y$-dir controller | Best IAE | Ave. IAE | Std. Dev. |
|---|---|---|---|---|---|
| Simulation case study | | | | | |
| DyHSBA | 5 | 5 | 1.0176 | 1.5029 | 0.3371 |
| DyHS-HASBA-Con | 5 | 5 | 0.9410 | 1.2843 | 0.3034 |
| DyHS-HASBA-Pref | 5 | 5 | 0.9031 | 1.1250 | 0.1500 |
| DyHSBA-V | 3 | 5 | 1.0179 | 1.4897 | 0.3170 |
| DyHS-HASBA-Con-V | 4 | 4 | 1.0016 | 1.3210 | 0.2328 |
| DyHS-HASBA-Pref-V | 4 | 6 | 1.0432 | 1.3423 | 0.2098 |
| Real-life case study: Environment-I | | | | | |
| DyHSBA | 5 | 5 | 0.6356 | 1.0791 | 0.4267 |
| DyHS-HASBA-Con | 5 | 5 | 0.5761 | 1.0522 | 0.3984 |
| DyHS-HASBA-Pref | 5 | 5 | 0.5897 | 1.0549 | 0.3730 |
| DyHSBA-V | 3 | 5 | 0.6840 | 1.1313 | 0.4201 |
| DyHS-HASBA-Con-V | 4 | 4 | 0.6737 | 1.0552 | 0.3743 |
| DyHS-HASBA-Pref-V | 4 | 6 | 0.6174 | 1.0065 | 0.3464 |
| Real-life case study: Environment-II | | | | | |
| DyHSBA | 5 | 5 | 0.5551 | 1.0794 | 0.4178 |
| DyHS-HASBA-Con | 5 | 5 | 0.5818 | 1.0889 | 0.4079 |
| DyHS-HASBA-Pref | 5 | 5 | 0.4646 | 0.9132 | 0.3615 |
| DyHSBA-V | 3 | 5 | 0.8219 | 1.1341 | 0.4020 |
| DyHS-HASBA-Con-V | 4 | 4 | 0.7415 | 1.0720 | 0.3786 |
| DyHS-HASBA-Pref-V | 4 | 6 | 0.7096 | 1.0474 | 0.4031 |

$x$-direction and $y$-direction controllers respectively, for both simulation and real implementations.

The process model is simulated using a fixed step 4th order Runge–Kutta method in this case study. All the parameters of $\underline{Z}$ are optimized in HSBA algorithm by the DyHSBA algorithm. In this simulation study, a harmony memory of ten harmonies is chosen and, for each simulation, 200 improvisations of the harmony memory are performed. Each control scheme, in both fixed and variable structure AFLC, is simulated for 10 sample runs and the best IAE values, average IAE values and standard deviations among the results are computed to compare the performances of the navigation schemes. In this case study too, the concurrent hybrid model and the preferential hybrid model are implemented utilizing the locally adaptive Lyapunov theory and DyHSBA algorithm-based global optimization technique. The adapta-

**Table 2** Comparison of best results of mobile robot navigation during simulation case study and real-life case studies for two sample environments

| Control Strategy | No. of MFs in $x$-dir controller | No. of MFs in $y$-dir controller | Best IAE | | |
|---|---|---|---|---|---|
| | | | Simulation | Env.-I | Env.-II |
| DyHSBA | 5 | 5 | 1.0176 | 0.6356 | 0.5551 |
| DyHS-HASBA-Con | 5 | 5 | 0.9410 | 0.5761 | 0.5818 |
| DyHS-HASBA-Pref | 5 | 5 | 0.9031 | 0.5897 | 0.4646 |
| PSOBA | 5 | 5 | 1.0153 | 0.8488 | 0.8325 |
| PSO-HASBA-Con | 5 | 5 | 1.6289 | 0.8133 | 0.6983 |
| PSO-HASBA-Pref | 5 | 5 | 0.9880 | 0.6715 | 0.6168 |
| GABA | 5 | 5 | 2.5849 | 1.6276 | 1.3656 |
| GA-HASBA-Con | 5 | 5 | 1.9328 | 1.5091 | 1.2647 |
| GA-HASBA-Pref | 5 | 5 | 1.4791 | 1.4145 | 1.1066 |
| DyHSBA-V | 3 | 5 | 1.0179 | 0.6840 | 0.8219 |
| DyHS-HASBA-Con-V | 4 | 4 | 1.0016 | 0.6737 | 0.7415 |
| DyHS-HASBA-Pref-V | 4 | 6 | 1.0432 | 0.6174 | 0.7096 |
| PSOBA-V | 4 | 4 | 0.9860 | 0.8260 | 0.9854 |
| PSO-HASBA-Con-V | 3 | 5 | 1.6280 | 0.9003 | 0.8887 |
| PSO-HASBA-Pref-V | 3 | 6 | 1.6004 | 0.8434 | 0.8444 |
| GABA-V | 5 | 4 | 1.8746 | 0.8716 | 0.8998 |
| GA-HASBA-Con-V | 5 | 5 | 1.2005 | 0.9883 | 0.9723 |
| GA-HASBA-Pref-V | 4 | 4 | 1.7710 | 1.0791 | 0.9707 |

tion gains $\nu_x = 0.01$ and $\nu_y = 0.1$ are taken for $x$-direction and $y$-direction controllers respectively. The results of DyHS algorithm-based simulation studies are tabulated in Table 1 which show the performances of robot navigation, in simulation, evaluated in implementation phase for different competing controllers, adapted during separate training procedures.

To show the wider usefulness of the proposed controller design, we next compare their performances vis-à-vis similar other controllers designed by competing popular stochastic optimization techniques, namely, genetic algorithm (GA) and particle swarm optimization (PSO). Table 2 first provides a comparative study among the DyHS-, PSO- and GA-based controllers designed for mobile robot navigation in terms of their best IAE values, in the context of the simulation studies. Among the competing controllers in fixed structure configuration, the HASBA-Pref model shows the best performance. In variable structure case, PSOBA model provides the best solution and the HASBA-Con model provides the second best solution, in terms of the IAE value computed between the planned path and the actual path traversed by the mobile robot, as shown in Table 2. In variable structure configuration, the number of input MFs and hence, the number of rules get reduced to some extent which will be very useful from the point of view of real implementation.

The overall best IAE performance is obtained with a fixed structure configuration. Hence it has been shown that the DyHS-HASBA-Pref strategy-based controller designed with fixed structure evolved as the best solution among the competing strategies developed, including those using PSO and GA, as reported in [17].

## 4.5 Experimental case study with real robot

The proposed HS algorithm-based design strategies are next implemented for the navigation of the autonomous mobile robot in two test environments with real robot and here also, for each case study, each control scheme is tested for 10 sample runs. The mobile robot navigation in real environments is achieved by using the adaptive fuzzy controllers trained during the HS algorithm-based simulation experiments. Figure 3 shows the robot navigation performance for a sample run avoiding obstacles, for test environment-I, utilizing fixed and variable structure versions of the DyHSBA, HASBA-Con model and HASBA-Pref model controllers. Figure 4 shows the robot navigation performances for another sample run for real test environment-II with fixed and variable structure versions of the DyHSBA, HASBA-Con model and HASBA-Pref model controllers. Table 1 also presents quantitative comparisons of performances of competing controllers, in terms of best IAE, average IAE and standard deviation in implementation phase, for both these real test environments chosen. Among all the mobile robot navigation schemes in two envi-

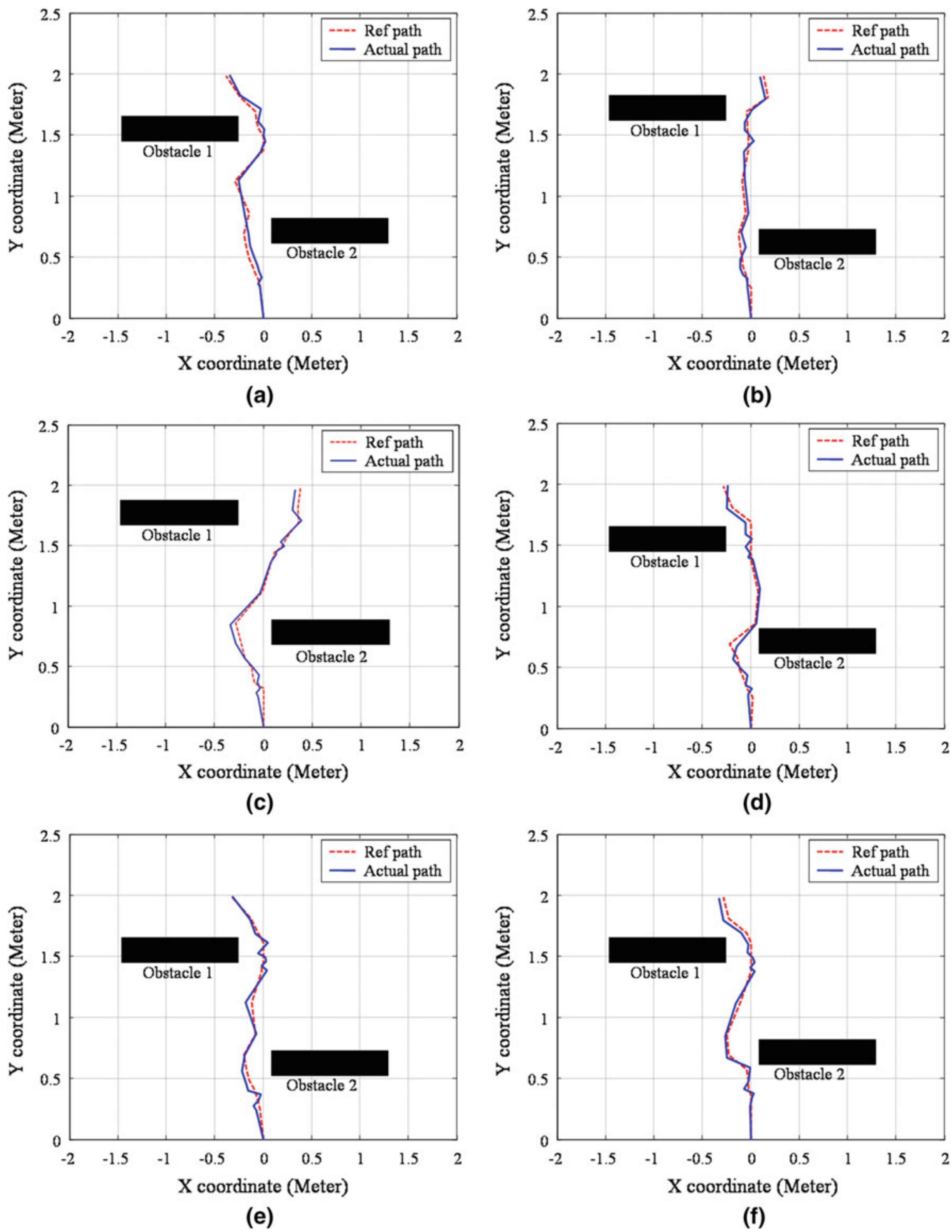


**Fig. 3** Trajectory of robot movement for real implementation in sample test environment-I, in a sample run, for: **a** fixed structure DyHSBA, **b** fixed structure DyHS-HASBA-Con model, **c** fixed structure DyHS-HASBA-Pref model, **d** variable structure DyHSBA, **e** variable structure DyHS-HASBA-Con model, and **f** variable structure DyHS-HASBA-Pref model, based adaptive fuzzy controller

ronments, considering both fixed structure and variable structure configurations, as shown in Table 1, the DyHS-HASBA-Pref model evolved as the best solution in three cases. In one case, DyHS-HASBA-Con model evolved as the best solution and DyHS-HASBA-Pref model evolved as the second best solution. Hence it can be inferred that the proposed

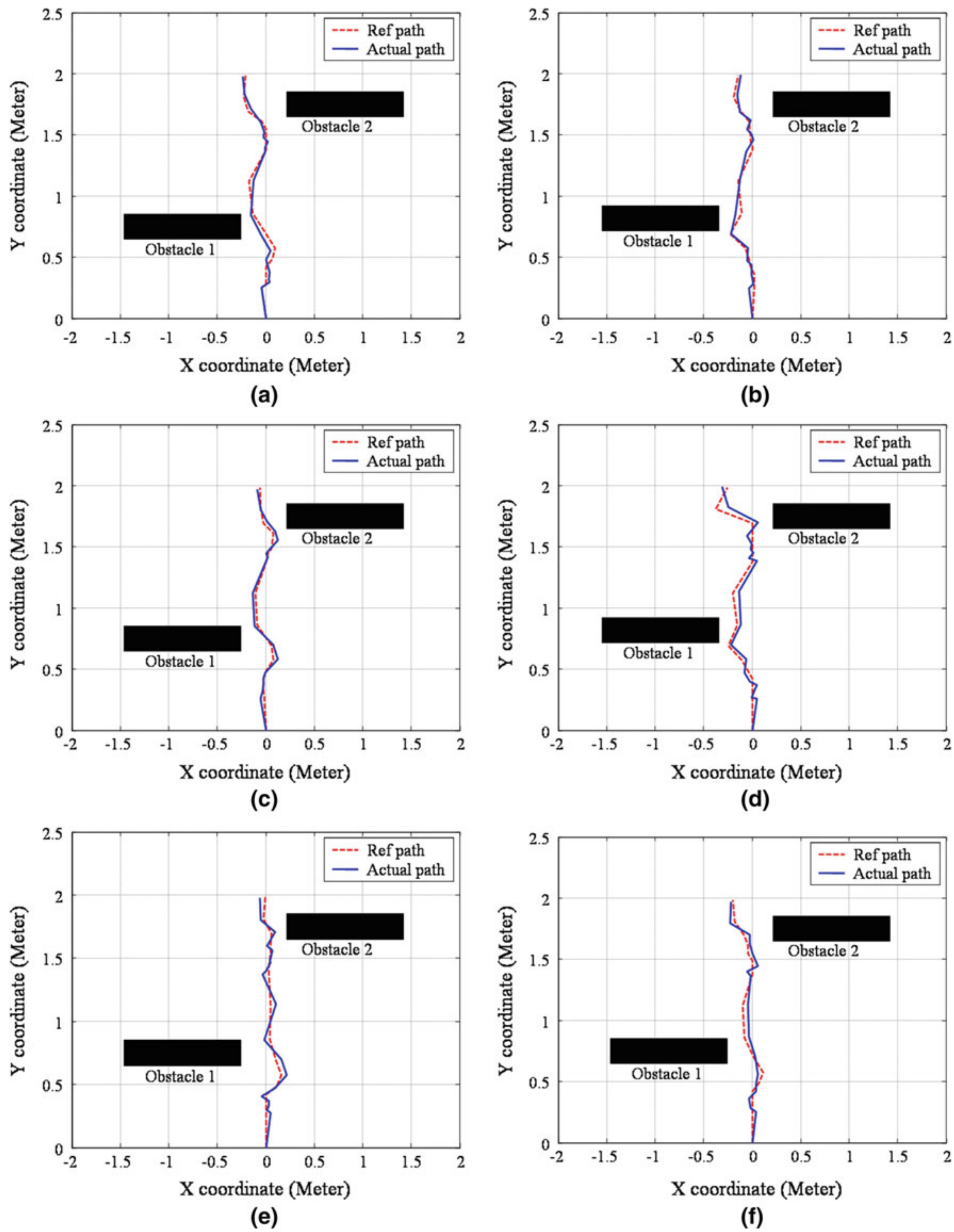


**Fig. 4** Trajectory of robot movement for real implementation in sample test environment-II, for a sample run, for: **a** fixed structure DyHSBA, **b** fixed structure DyHS-HASBA-Con model, **c** fixed structure DyHS-HASBA-Pref model, **d** variable structure DyHSBA, **e** variable structure DyHS-HASBA-Con model, and **f** variable structure DyHS-HASBA-Pref model, based adaptive fuzzy controller

DyHS-HASBA-Pref model of controller design evolved as the overall superior strategy, in terms of its best IAE values, average IAE values as reported in Table 1, considering both real implementations. From Table 1 it is also evident that the results obtained in proposed DyHS-HASBA-Pref model of controller design are most consistent as the standard deviations in IAE values obtained are minimum for this controller option in all cases except one case. As in the case of the simulation case studies, here also we demonstrate wider usefulness of the proposed DyHS-based control strategies by comparing their performances for the real environments vis-a-vis similar controllers designed using PSO- and GA-based strategies. These results are similarly demonstrated in Table 2 in terms of best IAE values.

Therefore, in both the simulation case studies and in real implementations in two test environments, the DyHS algorithm-based control strategies have performed their desired objectives of providing safe navigation path through the obstacles. It can also be concluded that, among the competing controller design strategies, DyHS-based hybrid preferential model evolved as the superior design methodology, on the whole, for the application of vision-based mobile robot navigation technique using adaptive state feedback fuzzy tracking controllers.

However, all these simulation and experimental results are obtained on the basis of performances obtained for sample environments and there is not sufficient statistical evidence available to claim that the proposed HS-based variants can significantly outperform the competing PSO and GA-based variants, under all situations. Hence it will be logical to claim that, overall, the proposed methods perform comparably with the competing counterparts and, in some situations, these proposed methods demonstrate superior performances.

## 5 Conclusion

The present paper proposed a novel idea for vision-based mobile robot navigation utilizing the stable adaptive fuzzy tracking controllers, designed by hybrid methodologies utilizing Lyapunov theory and a dynamic version of HS algorithm-based metaheuristic optimization. The stability of the designed controllers is guaranteed by the Lyapunov theory and the desired automation is achieved by the application of a proposed dynamic HS algorithm-based optimization strategy. A single camera-based vision sensing mechanism is used for capturing image of the environment, subsequently a path planning algorithm is implemented and the adaptive fuzzy tracking controllers are subsequently employed to enhance the applicability of the proposed scheme for both navigation and obstacle avoidance. Periodic image acquisition and path planning technique facilitates the scheme to tackle the dynamic environment of navigation. The proposed strategies are successfully implemented both in simulation and in two real environments. It has been successfully demonstrated that the proposed dynamic harmony search algorithm-based preferential hybrid design methodology evolved as a superior approach among the competing controllers for the sample experiments carried out. Overall it can be inferred that these proposed methods perform comparably with the competing counterparts and, in some situations, these methods demonstrate superior performances.

One of the important points to be noted is that in our formulations in this work we have considered all candidate solutions to be evolved using HS algorithms are fixed-length vectors. For variable structure FLCs we have utilized structural flags whose each individual value determines whether the corresponding MF will be considered or not in that iteration. Hence, it is natural that, in many iterations and for several candidate solutions, there will be parameters for MFs which will have no effect on the evaluation of the fitness/function. Hence, essentially some parameters will undergo random variations without affecting the overall solution. It is probable that this increased diversity will equip the HS algorithm with additional capability of exploring the search space more efficiently and it will be more likely to escape local optima. However, on the other hand, it may decrease the rate of convergence or may even prevent convergence in some cases. The authors wish to pursue this as their future scope of research where they will attempt to solve similar problems using variable-length vectors in the harmony memory [44] and will attempt to perform systematic analysis of solving identical problems using both fixed-length vectors and variable-length vectors.

## References


1. Rueb, K.D., Wong, A.K.C.: Structuring free space as a hypergraph for roving robot path planning and navigation. IEEE Trans. Pattern Anal. Mach. Intell. **9**(2), 263–273 (1987)
2. Habib, M.K., Yuta, S.: Efficient online path planning algorithm and navigation for a mobile robot. Int. J. Electron. **69**(2), 187–210 (1990)
3. Dijkstra, E.W.: A note on two problems in connection with graphs. Numer. Math. **1**, 269–271 (1959)
4. Nilsson, N.J.: Principles of artificial intelligence. Tioga Publishing Company, USA (1980)
5. Ge, S.S., Cui, Y.J.: New potential functions for mobile robot path planning. IEEE Trans. Robot. Auto. **16**(5), 615–620 (2000)
6. Valavanis, K.P., Hebert, T., Kolluru, R., et al.: Mobile robot navigation in 2-D dynamic environments using an electrostatic potential field. IEEE Trans. Syst. Man Cybern. A Syst. Humans **30**(2), 187–196 (2000)
7. Thrun, S., Burgard, W., Fox, D.: A probabilistic approach to concurrent mapping and localization for mobile robots. Mach. Learn. **31**(1–3), 29–53 (1998)
8. Ge, S.S., Cui, Y.J.: Dynamic motion planning for mobile robots using potential field method. Auton. Robots **13**(3), 207–222 (2002)
9. Ren, J., McIsaac, K.A., Patel, R.V.: Modified Newton's method applied to potential field based navigation for nonholonomic robots in dynamic environments. Robotica **26**, 285–294 (2008)
10. Ashokaraj, I.A.R., Silson, P.M.G., Tsourdos, A., White, B.A.: Robust sensor-based navigation for mobile robots. IEEE Trans. Instrum. Meas. **58**(3), 551–556 (2009)
11. Nirmal Singh, N., Chatterjee, A., Chatterjee, A., Rakshit, A.: A two-layered subgoal based mobile robot navigation algorithm with vision system and IR sensors. Measurement **44**, 620–641 (2011)

12. Hillel, A.B., Lerner, R., Levi, D., Raz, G.: Recent progress in road and lane detection: a survey. Mach. Vis. Appl. (2012). (published online on 07 February 2012)
13. Gueaieb, W., Miah, Md.S.: An intelligent mobile robot navigation technique using RFID technology. IEEE Trans. Instrum. Meas. **57**(9), 1908–1917 (2008)
14. DeSouza, G.N., Kak, A.C.: Vision for mobile robot navigation: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **24**(2), 237–267 (2002)
15. Shi, Z., Liu, Z., Wu, X., Xu, W.: Feature selection for reliable data association in visual SLAM. Mach. Vis. Appl. (2012). (published online on 07 July)
16. Shiao, Y., Yang, J., Su, D.: Path tracking laws and implementation of a vision-based wheeled mobile robot. WSEAS Trans. Circuits Syst. **12**(7), 965–976 (2008)
17. Das Sharma, K., Chatterjee, A., Rakshit, A.: A PSO-lyapunov hybrid stable adaptive fuzzy tracking control approach for vision based robot navigation. IEEE Trans. Instrum. Meas. **61**(7), 1908–1914 (2012)
18. Chatterjee, A., Watanabe, K.: An adaptive fuzzy strategy for motion control of robot manipulators. Soft Comput. **9**(3), 185–193 (2005)
19. Chatterjee, A., Chatterjee, R., Matsuno, F., Endo, T.: Augmented stable fuzzy control for flexible robotic arm using LMI approach and neuro-fuzzy state space modeling. IEEE Trans. Ind. Electron. **55**(3), 1256–1270 (2008)
20. Chatterjee, A., Chatterjee, R., Matsuno, F., Endo, T.: Neuro-fuzzy state modeling of flexible robotic arm employing dynamically varying cognitive and social component based PSO. Measurement **40**(6), 628–643 (2007)
21. Chatterjee, A., Watanabe, K.: An optimized Takagi-Sugeno type neuro-fuzzy system for modeling robot manipulators. Neural Comput. Appl. **15**(1), 55–61 (2005)
22. Chatterjee, A., Pulasinghe, K., Watanabe, K., Izumi, K.: A particle swarm optimized fuzzy-neural network for voice-controlled robot systems. IEEE Trans. Ind. Electron. **52**(6), 1478–1489 (2005)
23. Banerjee, S., Chakrabarty, A., Maity, S., Chatterjee, A.: Feedback linearizing indirect adaptive fuzzy control with foraging based online plant model estimation. Appl. Soft Comput. **11**(4), 3441–3450 (2011)
24. Chakrabarty, A., Banerjee, S., Maity, S., Chatterjee, A.: Fuzzy model predictive control of non-linear processes using convolution models and foraging algorithms. Measurement (2013, accepted). http://dx.doi.org/10.1016/j.measurement.2012.11.046
25. Bhattacharya, S., Chatterjee, A., Munshi, S.: An improved PID-type fuzzy controller employing individual fuzzy P, fuzzy I and fuzzy D controllers. Trans. Inst. Measure. Control **25**(4), 352–372 (2003)
26. Bhattacharya, S., Chatterjee, A., Munshi, S.: A new self-tuned PID-type fuzzy controller as a combination of two-term controllers. ISA Trans. **43**(3), 413–426 (2004)
27. Das Sharma, K., Chatterjee, A., Rakshit, A.: Design of a hybrid stable adaptive fuzzy controller employing Lyapunov theory and harmony search algorithm. IEEE Trans. Control Syst. Tech. **18**(6), 1440–1447 (2010)
28. Das Sharma, K., Chatterjee, A., Rakshit, A.: Adaptive fuzzy controller design employing harmony search algorithm for a class of non-linear system, pp. 42–45. In: Proc. of National Conf. Instrumentation and Control, Kolkata, India (2011)
29. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation **76**(2), 60–68 (2001)
30. Goldberg, D.E.: Genetic Algorithms in Search. In: Optimization and Machine Learning. Kluwer, Boston (1989)
31. Konar, A.: Computational Intelligence: Principles. In: Techniques and Applications. Springer, Berlin (2005)
32. Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. Wiley, New York (2006)
33. Lee, K.S., Geem, Z.W.: A new structural optimization method based on the harmony search algorithm. Comput. Struct. **82**, 781–798 (2004)
34. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. Appl. Math. Comput. **188**(2), 1567–1679 (2007)
35. Tian, Y.-H., Bo, Y.-M., Gao, M.-F.: Harmony annealing algorithm for multi-dimensional function optimization. Comput. Simul. **21**(10), 79–82 (2004)
36. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comp. Methods Appl. Mech. Engg. **194**(36–38), 3902–3933 (2005)
37. Geem, Z.W.: Optimal cost design of water distribution networks using harmony search. Eng. Optim. **38**(3), 259–280 (2006)
38. Geem, Z.W.: Recent Advances in Harmony Search Algorithm. Springer, Berlin (2010)
39. Astudillo, L., Castillo, O., Melin, P., Alanis, A., Soria, J., Aguilar, L.T.: Intelligent control of an autonomous mobile robot using type-2 fuzzy logic. Eng. Lett. **13**(2), 93–97 (2006)
40. Wang, L.X.: Stable adaptive fuzzy control of nonlinear system. IEEE Trans. Fuzzy Syst. **1**(2), 146–155 (1993)
41. Fischle, K., Schroder, D.: An improved stable adaptive fuzzy control method. IEEE Trans. Fuzzy Syst. **7**(1), 27–40 (1999)
42. Das Sharma, K., Chatterjee, A., Rakshit, A.: A hybrid approach for design of stable adaptive fuzzy controllers employing Lyapunov theory and particle swarm optimization. IEEE Trans. Fuzzy Syst. **17**(2), 329–342 (2009)
43. Das Sharma, K., Chatterjee, A., Matsuno, F.: A Lyapunov theory and stochastic optimization based stable adaptive fuzzy control methodology, pp. 1839–1844. In: Proc. of SICE Intl. Conf. Instrumentation, Control and Information Technology, Japan (2008)
44. Alia, O.M., Mandava, R., Ramachandram, D., Aziz, M.E.: Dynamic fuzzy clustering using Harmony Search with application to image segmentation. In: Proc. IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 538–543 (2009) (14–17 Dec. 2009)
45. Singh, N.N.: Vision based autonomous navigation of mobile robots. Jadavpur University, Ph.D. dissertation (2010)
46. Sasaki, T., Brscic, D., Hashimoto, H.: Human-observation-based extraction of path patterns for mobile robot navigation. IEEE Trans. Ind. Electron. **57**(4), 1401–1410 (2010)
47. Subramanian, R., Spalding, E.P., Ferrier, N.J.: A high throughput robot system for machine vision based plant phenotype studies. Mach. Vis. Appl. (2012). (published online on 16 June)
48. Kim, P.G., Park, C.G., Jong, Y.H., Yun, J.H., Mo, E.J., Kim, C.S., Jie, M.S., Hwang, S.C., Lee, K.W.: Obstacle avoidance of a mobile robot using vision system and ultrasonic sensor. In: Proc. 3rd Intl. Conf. on Advanced Intelligent Computing Theories and Applications, ICIC'07, vol. 4681, pp. 545–553 (2007)

## Author Biographies

**Kaushik Das Sharma** received his B. Sc. (Physics Hons.), B. Tech. (Electrical Engg.) and M. Tech. (Electrical Engg.) degrees from University of Calcutta, Kolkata, India, in 1998, 2001 and 2004, respectively and Ph.D. (Engg.) degree from Jadavpur University, Kolkata, India in 2012. Presently he is an Assistant Professor of Department of Applied Physics, University of Calcutta, India. His research interests include fuzzy control system design, stochastic optimization applications, image processing etc. He has authored or coauthored about 20 technical papers, including 9 international journal papers.

**Amitava Chatterjee** received the B.E., M.E., and Ph.D. degrees in electrical engineering from Jadavpur University, Kolkata, India, in 1991, 1994, and 2002, respectively. In 1997, he joined the Department of Electrical Engineering, Jadavpur University, where he is currently an Associate Professor. In 2003, he received the Japanese Government (Monbukagakusho) Scholarship and visited Saga University, Saga, Japan. From November 2004 to November 2005, he was with the University of Electro-Communications, Tokyo, Japan, on a Japan Society for the Promotion of Science (JSPS) Post-Doctoral Fellowship for Foreign Researchers. In March–May, 2004, and in May–June, 2009, he visited University of Paris XII, Val de Marne, France, as an invited Teacher. His research interests include fuzzy based nonlinear control, intelligent instrumentation, robotics, image processing and pattern recognition, signal processing, and stochastic optimization techniques. He presently serves as an Editor of *IEEE Transactions on Vehicular Technology* and as an Associate Editor of *IEEE Transactions on Instrumentation and Measurement*, *IEEE Sensors Journal*, and *Control Engineering Practice*. He has authored or coauthored about 95 technical papers, including 65 international journal papers.

**Anjan Rakshit** received the ME and PhD degrees in electrical engineering from Jadavpur University, Kolkata, India, in 1978 and 1987, respectively. He is a Retired Professor from the Department of Electrical Engineering, Jadavpur University. His research interests include digital signal processing, Internet-based instrumentation, smart instrumentation, intelligent control, and design of real-time systems. He has published about 60 technical papers in his areas of research interest.