Copyright © 2011 by ABCM

OPTIMIZATION OF KINEMATICALLY REDUNDANT MANIPULATORS FORCE CAPABILITIES USING IMPROVED HARMONY SEARCH

Lucas Weihmann, lucas.weihmann@pucpr.br
Industrial and Systems Engineering Graduate Program, PPGEPS,
Pontifical Catholic University of Parana, PUCPR
Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Parana, Brazil
Daniel Martins, daniel@emc.ufsc.br
Department of Mechanical Engineering
Federal University of Santa Catarina, UFSC
Campus Universitário, Trindade, Zip code 88040-900, Florianópolis, Santa Catarina, Brazil
Diego Luis de Andrade Bernert, dbernert@gmail.com
Leandro dos Santos Coelho, leandro.coelho@pucpr.br
Industrial and Systems Engineering Graduate Program, PPGEPS
Pontifical Catholic University of Parana, PUCPR
Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Parana, Brazil

Abstract. *The Harmony Search (HS) algorithm is a new evolutionary metaheuristic algorithm based on natural musical performance processes that occur when a musician searches for a better state of harmony, such as during jazz improvisation. In the HS algorithm, the solution vector is analogous to the harmony in music, and the local and global search schemes are analogous to musician's improvisations.The HS algorithm uses a random search, which is based on random selection, memory consideration, and pitch adjusting. In this paper, an Improved HS (IHS) approach based on truncated Cauchy distribution is proposed and evaluated. The proposed IHS presents an efficient strategy to improve the search performance in preventing premature convergence to local minima when compared with the classical HS algorithm. The efficiency and feasibility of the proposed IHS approach is demonstrated on a force optimization problem, where the force capabilities of a serial PRRR manipulator are evaluated considering actuation limits and different configurations.*

Keywords*: Harmony search, Optimization, Robotics, redundant manipulator.*

1. INTRODUCTION

Many techniques from conventional mathematical methods, such as quadratic programming and nonlinear programming, have been proposed to deal with optimization problems. Conventional optimization techniques take in advantages in computing speed and convergence with the objective function of continuous, differentiable and single peak value. In robotics, these optimization techniques can be applied to robot design, trajectory generation, actuators force distribution, and inverse kinematic computation.

However, some force optimization problems cannot be handled with conventional mathematical methods. In the presence of geometric variable parameters, evaluate the maximum wrench (force and moment) that a manipulator can apply (or sustain) for a given pose without exceeding the actuators limits generates non-linear and non-convex functions over the search space that most methods often give only a local optimum solution. In such cases, alternative optimization methods like Evolutionary Algorithms (EAs) (Michalewicz and Fogel, 2010; Eiben and Smith, 2010) should be used to find a global optimum solution, avoid being trapped in a local minimum.

Nowadays, the use of EAs to solve optimization problems is a common practice due to their competitive performance on complex search spaces. EAs are well known for their ability to deal with nonlinear and complex optimization problems. The primary advantage of EAs over other numerical methods is that they just require the objective function values, while properties such as differentiability and continuity are not necessary.

In terms of EAs, the Harmony Search (HS) algorithm developed by Geem *et al*. (2001) was originally conceptualized using the musical improvisation process of searching for a perfect state of harmony. The HS algorithm uses a random search, which is based on random selection, memory consideration, and pitch adjusting. In comparison to other meta-heuristics in the literature, the HS algorithm imposes fewer mathematical requirements and can be easily adapted for solving various kinds of engineering optimization problems (Mahdavi and Fesanghary, 2007; Omran and Mahdavi, 2008). The HS algorithm has been successfully applied to various benchmark problems and has been extensively applied to various real-world industrial problems (Geem, 2008; Ingram and Zhang, 2009).

In this paper, an Improved HS (IHS) approach based on truncated Cauchy distribution is proposed and evaluated. The proposed IHS presents an efficient strategy to improve the search performance in preventing premature convergence to local minima when compared with the classical HS algorithm.

The efficiency and feasibility of the proposed IHS approach is demonstrated on a force optimization problem, where the force capabilities of a serial PRRR manipulator are evaluated considering actuation limits and different configurations.

The remainder of this paper is organized as follows. In section 2, the fundamentals of HS and IHS approaches are detailed. In section 3, the description of case study related to force optimization is presented. Simulation results are presented in section 4. Finally, section 5 outlines the conclusion and future research.

2. FUNDAMENTALS OF CLASSICAL HS AND IHS APPROACHES

The HS algorithm conceptualizes a behavioral phenomenon of musicians in the improvisation process whereby each player searches to improve the tune in order to produce a better state (i.e. global optimum) of harmony (Geem *et al*., 2001). In music improvisation, each player sounds any pitch within the possible range, together making one harmony vector. If all the pitches make a good solution, that experience is stored in each variable's memory, and the possibility to make a good solution is also increased next time. Similarly, the optimization process seeks a superior vector in terms of objective function. This is the core analogy between improvisation and optimization in the HS algorithm (Geem, 2008).

The next subsections describe the HS algorithm. First, a brief overview of the classical HS is provided for formulating solution vectors in which the optimization process is generated and the object function is evaluated; and finally, the proposed IHS is explained.

2.1. Classical HS

The optimization procedure of the HS algorithm can be synthesized in the following steps (Coelho and Bernert, 2009):

Step 1. Initialize the optimization problem and HS algorithm parameters. First, the optimization problem is specified as follows:

$$\text{Minimize } f(x) \text{ subject to } x_i \in X_i, \quad i = 1,..., N$$

where $f(x)$ is the objective function, x is the set of each decision variable (x_i); X_i is the set of the possible range of values for each design variable (continuous design variables), that is, $x_{i,lower} \le X_i \le x_{i,upper}$, where $x_{i,lower}$ and $x_{i,upper}$ are the lower and upper bounds for each decision variable; and N is the number of design variables. In this context, the HS algorithm parameters that are required to solve the optimization problem are also specified in this step. The number of solution vectors in harmony memory (HMS) that is the size of the harmony memory matrix, harmony memory considering rate (HMCR), pitch adjusting rate (PAR), and the maximum number of searches (stopping criterion) are selected in this step. Here, HMCR and PAR are parameters that are used to improve the solution vector. Both are defined in *Step 3*.

Step 2. Initialize the harmony memory. The harmony memory (HM) is a memory location (matrix) where all the solution vectors (sets of decision variables) are stored. The HM matrix, shown in Eq. (1), is filled with randomly generated solution vectors using uniform distribution, where

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

In Eq. (1), each row represents each solution vector, and the number of total vectors is HMS (harmony memory size).

Step 3. Improvise a new harmony from the HM. A new harmony vector, $x' = (x_1', x_2',..., x_N')$, is generated based on three rules: i) memory consideration, ii) pitch adjustment, and iii) random selection. The generation of a new harmony is called 'improvisation'.

In the memory consideration, the value of the first decision variable (x_1') for the new vector is chosen from any value in the specified HM range $\left(x_1' - x_1^{HMS}\right)$. Values of the other decision variables $(x_2',..., x_N')$ are chosen in the same manner. The HMCR, which varies between 0 and 1, is the rate of choosing one value from the historical values stored in the HM, while (1 - HMCR) is the rate of randomly selecting one value from the possible range of values.

$$x_i^{'} \leftarrow \begin{cases} x_i^{'} \in \left\{x_i^1, x_i^2, ..., x_i^{\mathrm{HMS}}\right\} & \text{with probability HMCR} \\ x_i^{'} \in X_i & \text{with probability (1 - HMCR).} \end{cases} \tag{2}$$

After, 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 rate of pitch adjustment as follows:

$$\text{Pitch adjusting decision for } x_i^{'} \leftarrow \begin{cases} \text{Yes} & \text{with probability PAR} \\ \text{No} & \text{with probability (1 - PAR).} \end{cases} \tag{3}$$

The value of (1 - PAR) sets the rate of doing nothing. If the pitch adjustment decision for $x_i^{'}$ is Yes, $x_i^{'}$ is replaced as follows:

$$x_i^{'} \leftarrow x_i^{'} \pm r \cdot bw, \tag{4}$$

where *bw* is an arbitrary distance bandwidth, *r* is a random number generated using uniform distribution between 0 and 1. In *Step 3*, HM consideration, pitch adjustment or random selection is applied to each variable of the new harmony vector in turn.

Step 4. Update the HM. If the new harmony vector, $x' = (x_1^{'}, x_2^{'}, ..., x_N^{'})$ is better than the worst harmony in the HM, judged in terms of the objective function value, *F*, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

Step 5. Repeat *Steps 3* and *4* until the stopping criterion has been satisfied, usually a sufficiently good objective function or a maximum number of iterations (generations), t_{max}. Maximum number of iterations criterion is adopted in this work.

2.2. Proposed IHS algorithm

A significant amount of research has already been undertaken on the application of HS for solving difficult practical optimization problems as well as to improve the performance of HS by tuning its parameters and/or blending it with other powerful optimization techniques. The efficiency of most EAs, as the HS approaches depends on how they balance between the explorative and exploitative tendencies in the course of search. Exploitation means the ability of an algorithm to use the information already collected and thus to orient the search towards the goal while exploration is the process that allows introduction of new information into the population (Mukhopadhyay *et al*., 2008).

Several papers have presented promising results using EAs combined with Cauchy distribution (see examples in Rudolph, 1997; Yao *et al*., 1999). In this paper, the proposed IHS uses truncated Cauchy distribution in the range [0,1] to generate the values for HMCR, described in Eq. (2), is proposed. The utilization of Cauchy distribution can be useful in IHS. Due to Cauchy be more expanded than Gaussian distribution, it allows, probabilistically speaking, large steps and in this way, generating more different values for the HMCR. Furthermore, the proposed IHS employs *bw* with decreasing linear during the generations with initial value equal to 0.5 and final value equal to 0.01.

3. DESCRIPTION OF CASE STUDY

In robotics, force capability can be defined as the maximum wrench (force and moment) that a manipulator can apply or sustain in a given direction (Nokleby *et al*., 2005). In this work, force capability of the PRRR manipulator will be considered a pure force analysis. The optimization problem consists in maximize F_x force in *x* direction, while F_y force and M_z moment are equal to zero.

The manipulator being considered in the optimization problem is a PRRR serial manipulator shown in Fig. 1. It has one prismatic and three revolute joints and, due to its geometry, motion possibilities are restricted to *x-y* plane. When executing a task that requires three degree of freedom (DOF), the PRRR manipulator has one additional DOF and is said to be kinematically redundant. This additional mobility allows the manipulator to achieve the same posture (end effector position and orientation) with infinite configurations. Kinematic redundancy can be used to improve the force capabilities of manipulators.

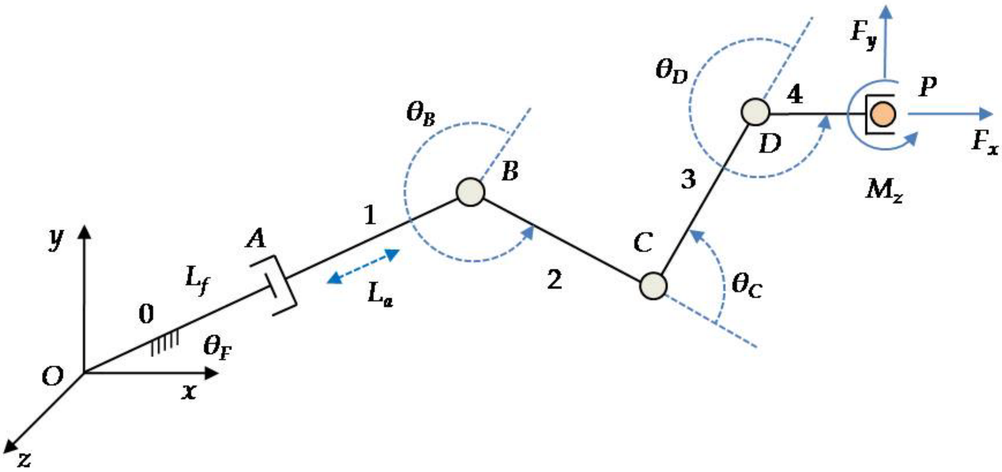

Figure 1. PRRR serial manipulator.

For the PRRR manipulator, the force equations that define the relation between contact wrenches and actuation torques can be written as:

$$\begin{Bmatrix} F_A \\ F_x \\ F_y \\ M_z \end{Bmatrix} = [K]_{4,3} \begin{Bmatrix} \tau_B \\ \tau_C \\ \tau_D \end{Bmatrix} \tag{5}$$

where F_A, τ_B, τ_C and τ_D denote *A*, *B*, *C* and *D* joints wrenches, respectively. The *K* matrix is function of link lengths, of *La* prismatic joint displacement and of X_P and Y_P coordinates. For each individual of the population the wrenches *F* can be calculated by Eq. (5). The maximum force capability for prismatic actuator *A* is ±50 N. The maximum torque capabilities for revolute actuators *B*, *C* and *D* are ±10, ±8 and ±6 Nm, respectively.

The links of the PRRR manipulator are labeled from 0 to 4, where 0 is the base fixed link and 4 is the end effector (EF) link. Joints are identified from the base to the extremity by the letters *A*, *B*, *C* and *D*, respectively. Links 2 and 3 lengths are equal to 0.3 m and link 4 length is 0.05 m. Angle θ_f will be held constant during this work at 30 degrees and L_f length is also constant an equal to 0.3 m. Manipulators configuration is defined by L_a prismatic joint displacement and by θ_B, θ_C and θ_D joint angles. Actually, given L_a, there are two possible configurations know in the literature as 'elbow up' and 'elbow down'. In this study only the configuration 'elbow up', where $-\pi \leq \theta_C \leq 0$, will be considered. Point *P* is the contact point between manipulator and environment with coordinates X_P and Y_P. In the optimization process, X_P coordinate will be constant and equal to 0.7 m. Y_P coordinate is a decision variable and can assume values in the range $0.1 \leq Y_P \leq 0.5$. Orientation of the EF is always pointing in the *x* direction, as shown in Fig. 1.

Taken these assumptions into account, coordinates X_A and Y_A of point *A* and coordinate X_D of point *D* are known. Given Y_P coordinate, the whole configuration of the manipulator can be evaluated with respect to L_a length only. Since L_a is a decision variable, during the optimization process the manipulator will achieve a feasible configuration when the link lengths evaluated by the inverse position kinematic are equal to their real values. Y_P, τ_B, τ_C and τ_D are also decision variables and the force optimization problem can be stated as:

$$\text{Maximize:} \quad F_x(Y_P, L_a, \tau_B, \tau_C, \tau_D) \tag{6}$$

$$\text{Restricted to:} \begin{cases} F_y(Y_P, L_a, \tau_B, \tau_C, \tau_D) = 0 \\ M_z(Y_P, L_a, \tau_B, \tau_C, \tau_D) = 0 \\ 0.1 \text{ m} < Y_P < 0.5 \text{ m} \\ 0.05 \text{ m} < L_a < 0.3 \text{ m} \\ -10 \text{ Nm} < \tau_B < 10 \text{ Nm} \\ -8 \text{ Nm} < \tau_C < 8 \text{ Nm} \\ -6 \text{ Nm} < \tau_D < 6 \text{ Nm} \\ -50 \text{ N} < F_A < 50 \text{ N} \\ \sqrt{(X_D - X_C)^2 + (Y_D - Y_C)^2} = 0.3 \text{ m} \\ \sqrt{(X_C - X_B)^2 + (Y_C - Y_B)^2} = 0.3 \text{ m} \end{cases} \tag{7}$$

Restrictions in the decision variables were handled directly by the HS and IHS algorithms. Geometric and force/moment restriction must be introduced as penalties in the objective function. The geometric penalty part is of the objective function can be written in the form:

$$f_{geo} = \left(\sqrt{(X_D - X_C)^2 + (Y_D - Y_C)^2} - 0.3\right)^2 + \left(\sqrt{(X_C - X_B)^2 + (Y_C - Y_B)^2} - 0.3\right)^2 \quad (8)$$

Since the proposed algorithm works with minimization problems, F_x maximization can be treated as a minimization of the inverse of F_x and the whole objective function is:

$$f_{obj} = R_1\left(f_{geo}\right) + R_2\left(1/F_x\right) + R_3\left(|F_y| + |M_z|\right) \quad (9)$$

where R_1, R_2 and R_3 are scalars that properly weights the contributions. In the optimization process, with R_1=50, R_2=100, and R_3=1 convergence were achieved for HS and IHS algorithms.

4. OPTIMIZATION RESULTS

In order to eliminate stochastic discrepancy, in each case study, 30 independent runs in MATLAB (MathWorks) were made for each of the optimization methods involving 30 different initial trial solutions for each optimization method.

The total number of solution vectors in classical HS and IHS, i.e., the HMS was 30 and t_{max} = 1,000 generations. All tested HS approaches adopted 30,000 objective function evaluations in each run. Furthermore, the *bw*, HMCR and PAR were 0.01, 0.9 and 0.3, respectively, in setup of the classical HS. On the other hand, IHS employed the procedure mentioned in section 2.2 for tuning the *bw* and HMCR. However, the IHS adopted PAR equal to 0.3.

The results obtained for case study of force optimization described in the section 3 are given in Tab. 1, which shows that the IHS presented best mean and minimum values of objective function than the results of HS in 30 runs. The best results obtained for solution vector with HS and IHS with minimum objective function values are given in Tab. 2.

Table 1. Results of HS and IHS in terms of objective function for the case study in 30 runs.
A result with **Boldface** means the best value found.

optimization method	maximum ($/h)	mean ($/h)	minimum ($/h)	standard deviation ($/h)
Classical HS	4.600664	2.863870	1.780236	0.779215
IHS	1.705538	**1.629157**	**1.459840**	0.056001

Table 2. Best solution obtained using HS and IHS.

Decision variables	Value using HS	Value using IHS
Y_P – coordinate of point *P*	0.4359	0.4454
L_a – displacement of joint *A*	0.2530	0.2992
τ_B – torque of joint *B*	-9.0170	-10.000
τ_C – torque of joint *C*	6.9518	8.000
τ_D – torque of joint *D*	0.0139	-0.0016

The distribution of the best solutions along variable space in 30 runs is another important point of analysis. Figure 2 and Fig. 3 show the histogram of the variables #1 to #5 for HS and IHS, respectively. An important observation could be extracted from Fig. 2 and Fig. 3: the most sensitive approach is the HS.

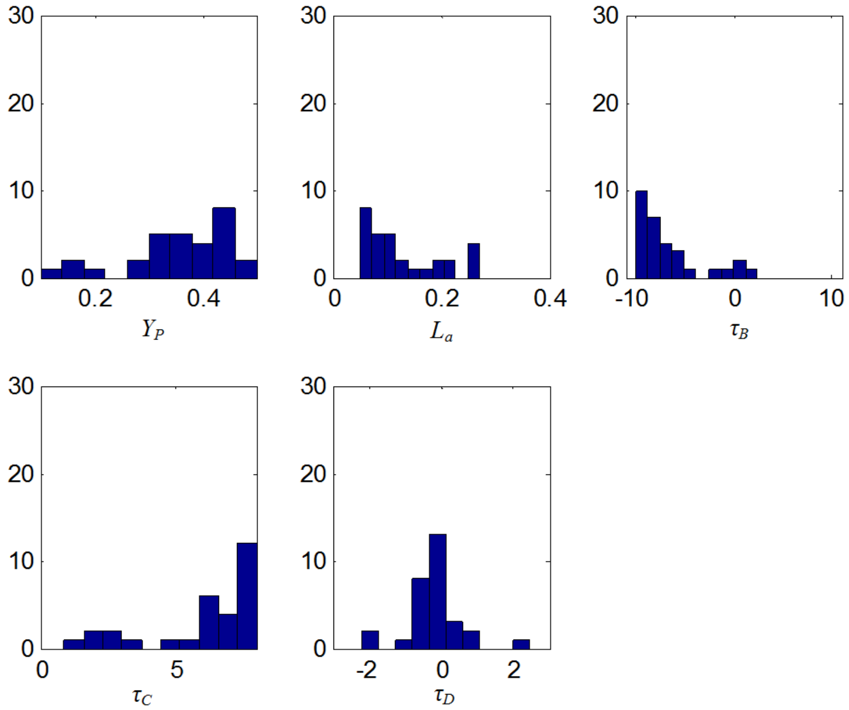

Figure 2. Histogram of the optimization variables using HS.

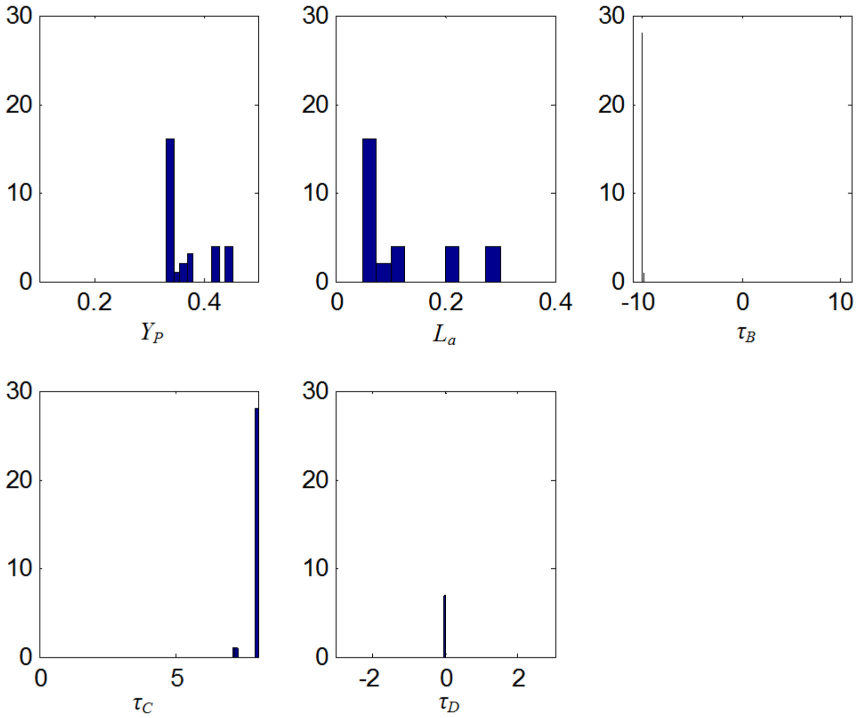

Figure 3. Histogram of the optimization variables using IHS.

As seen in Tab. 2, in the best solution found with HS, joint torques did not reach their limits. This means that better force capabilities values could be found. Using IHS, joints *B* and *C* are saturated and full joint actuation capacity are in use when the best solution is found. Figure 4 shows manipulators configuration in the best solution found using IHS.

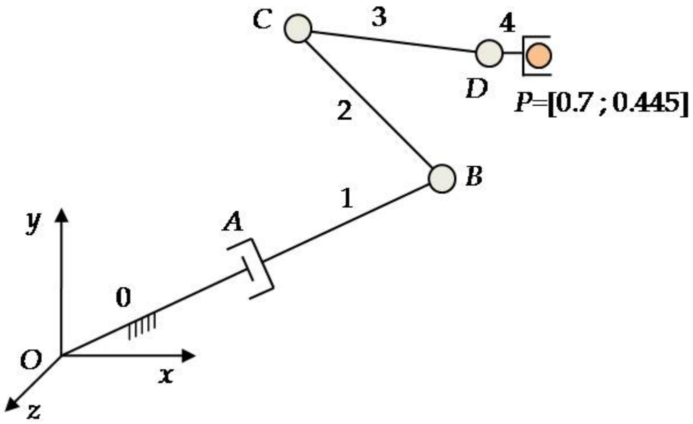

Figure 4. PRRR manipulator configuration in the best solution found with IHS

5. CONCLUSION

The HS is a music-inspired algorithm, mimicking the improvisation process of music players. In the HS algorithm, musical performances seek a perfect state of harmony determined by aesthetic estimation, as the optimization algorithms seek a best state (i.e. global optimum) determined by objective function value, achieved by assign suitable values to each design variable. Furthermore, the HS is simple in concept, few in parameters, and easy in implementation, with theoretical background of stochastic derivative.

This paper has introduced a harmony search algorithm to solve a force optimization problem of a planar PRRR serial manipulator. Simulation results for a force optimization problem presented in Tab. 1 and Tab. 2 revealed that the proposed IHS presented promising results in terms of quality of solution when compared the classical HS. In the future, formulation of IHS including effect of diversity control mechanisms should be investigated in order to deal with multiobjective optimization problem in robotics field.

6. ACKNOWLEDGEMENTS

This work was partially supported by the National Council of Scientific and Technologic Development of Brazil — CNPq — under Grants 303963/2009-3/PQ and 'Fundação Araucária' under Grant 14/2008-416/09-15149.

7. REFERENCES

Coelho, L. S., Bernert, D. L. A. Bernert, 2009, "An improved harmony search algorithm for synchronization of discrete-time chaotic systems", Chaos, Solitons & Fractals, Vol. 41, No. 5, pp. 2526-2532.

Eiben, A. E., Smith, J. E., 2010, "Introduction to evolutionary computing", Natural Computing Series, Springer, Heidelberg, Germany.

Geem, Z. W., 2008, "Soft computing applications in industry", STUDFUZZ 226 (B. Prasad, editor), pp. 117-134.

Geem, Z. W., Kim, J. H., Loganathan, G. V., 2001, "A new heuristic optimization algorithm: harmony search", Simulation, Vol. 76, No. 1, pp. 60-68.

Ingram, G., Zhang, T., 2009, "Overview of applications and developments in the harmony search algorithm", Music-inspired harmony search algorithm (Z. W. Geem, editor), Springer, Heidelberg, Germany, pp. 15-37.

Mahdavi, M., Fesanghary, M., Damangir, E., 2007, "An improved harmony search algorithm for solving optimization problems, Applied Mathematics and Computation, Vol. 188, No. 2, pp. 1567-1579.

Michalewicz, Z., Fogel, D. B., 2010, "How to solve it: modern heuristics", Springer, Heidelberg, Germany, 2nd edition, Revised and Extended Edition.

Mukhopadhyay, A., Roy, A., Das, S., Abraham, A., 2008, "Population-variance and explorative power of harmony search: an analysis", Proceedings of the 3rd International Conference on Digital Information Management, London, UK, pp. 775-781.

Nokleby S.B., Fisher R., Podhorodeski R.P., Firmani F., 2005, "Force capabilities of redundantly-actuated parallel manipulators", Mechanism and Machine Theory, Vol. 40, No. 5, pp. 578-599.

Omran, M. G. H., Mahdavi, M., 2008, "Global-best harmony search", Applied Mathematics and Computation, Vol. 198, No. 2, pp. 643-656.

Rudolph, G., 1997, "Local convergence rates of simple evolutionary algorithms with Cauchy distributions", IEEE Transactions on Evolutionary Computation, Vol. 1, No. 4, pp. 249-258.

Yao, X., Liu, Y., Lin, G., 1999, "Evolutionary programming made faster", IEEE Transactions on Evolutionary Computation, Vol. 3, No. 2, pp. 82-102.