10.24425/acs.2018.124707

Archives of Control Sciences
Volume 28(LXIV), 2018
No. 3*, pages 349–378*

Navigation of humanoids by a hybridized regression-adaptive particle swarm optimization approach

PRIYADARSHI BIPLAB KUMAR, CHINMAYA SAHU and DAYAL R. PARHI

In the era of humanoid robotics, navigation and path planning of humanoids in complex environments have always remained as one of the most promising area of research. In this paper, a novel hybridized navigational controller is proposed using the logic of both classical technique and computational intelligence for path planning of humanoids. The proposed navigational controller is a hybridization of regression analysis with adaptive particle swarm optimization. The inputs given to the regression controller are in the forms of obstacle distances, and the output of the regression controller is interim turning angle. The output interim turning angle is again fed to the adaptive particle swarm optimization controller along with other inputs. The output of the adaptive particle swarm optimization controller termed as final turning angle acts as the directing factor for smooth navigation of humanoids in a complex environment. The proposed navigational controller is tested for single as well as multiple humanoids in both simulation and experimental environments. The results obtained from both the environments are compared against each other, and a good agreement between them is observed. Finally, the proposed hybridization technique is also tested against other existing navigational approaches for validation of better efficiency.

Key words: navigation, humanoid NAO, RA, APSO, Petri-Net, V-REP

1. Introduction

With the development of science and technology, robots are becoming an integral part of human life. Robots are used in several forms in various industries dealing with manufacturing, surgery, medical assistance, defense, etc. The ability to mimic the human behaviour makes the humanoid robots more flexible than other forms of robots. Humanoids have the compatibility of working in a platform used by the humans, and they can replace the human efforts if required. A humanoid robot needs to be equipped with computational intelligence

The authors are with Robotics Laboratory, Mechanical Engineering Department, National Institute of Technology, Rourkela-769008, Odisha, India.
E-mails: p.biplabkumar@gmail.com, mechchinu@gmail.com, dayaldoc@yahoo.com
Corresponding Author: P.B. Kumar

Received 23.12.2017. Revised 19.03.2018.

to negotiate with obstacles present in a complex terrain. Over the last few years, several researchers have attempted the analysis of path planning and navigation of several forms of robots.

Mohanty et al. (Mohanty et al. 2013; Mohanty et al. 2016; Mohanty et al. 2014) have developed several navigational techniques for path planning of mobile robots using artificial intelligence (AI). They discussed modification of controlling parameters of basic intelligent algorithms for performance improvement. Singh et al. (Singh et al. 2009; Singh et al. 2011; Parhi et al. 2009) have discussed about use of computational intelligence for smooth and collision free path generation for wheeled mobile robots. Hugel and Jouandeaue (2012) developed a 3D LIP model for the walking pattern of humanoid robots. They have assumed a model without any torque in the support phase and predefined the center of mass of the robot. Sadedel et al. (2014) used hip and foot trajectories to propose an offline path planning approach for 2D humanoid robots. They verified the stability condition of the robot along with incorporation of genetic algorithm for safe walking pattern generation. Karkowski et al. (2016) developed a real time path planning approach for humanoids using A* algorithm and adaptive 3D action set. They formulated a systematic step search by considering the height information. Ido et al. (2009) proposed a view based navigation of humanoids. They used the motion capture data as input to a view based sequence and analyzed quantitative effect on walking pattern. Dalibard et al. (2013) proposed a randomized algorithm for the dynamic walking of the humanoids by generating a collision free path. Clever and Mombaur (2016) introduced a motion transfer scheme from humans to humanoids based on inverse optimal control scheme. They extracted the motion parameters from human walking and used them in humanoid walking. Mirjalili et al. (2016) proposed online path planning approaches for SURENA-III humanoid robot based on control schemes. They calculated the required joint torques by considering an inverted pendulum model by replacing the conventional whole body dynamic model. Shakiba et al. (2013) modified the basic particle swarm optimization (PSO) technique by adding Ferguson splines to it and used the revised algorithm to generate collision free path for soccer playing humanoids. Perrin et al. (2011) discussed an equivalence between different footstep planning approaches to enable classical motion planning techniques to be applied to humanoid path planning. Ryu et al. (2013) used natural path generated by humans to use in waypoint based path generation for humanoids. Shimizu and Sugihara (2012) proposed a path planning approach for humanoids based on transitional sequence of the double support phase. Fen et al. (2012) improved basic ant colony based optimization (ACO) technique for path planning of a humanoid manipulator. Kanoun et al. (2011) developed a path planning approach for humanoids based on foot placements. They considered a virtual kinematic tree as an inverse kinematics problem to generate the data required for motion planning. Schmid and Woern (2005) used

NURBS curve to generate smooth and collision free path for humanoids. Niskiwaki et al. (2012) used a laser range finder to navigate humanoids in a complex environment. Yoo and Kim (2015) developed a gaze control based architecture for navigation of humanoid robots in complex environments. They modified the unscented Kalman filter based controller and synchronized the filter into walking pattern. The research based on navigation and path planning is primarily focused on mobile robots. A very less number of works have been reported on the navigation of humanoid robots. Although some of the researchers have attempted navigation of humanoid robots, there are some limitations associated with their approaches regarding specific environmental conditions. Along with that, the navigation of multiple humanoid robots in a single environment cluttered with obstacles is rarely reported in the available literatures according to the authors' knowledge.

The use of individual AI techniques for the path planning of humanoids may not always be self-sufficient to work in a dynamic and cluttered environment. Therefore, hybridization is attempted using multiple techniques to improve the limitations of the standalone methods. Hybridization has been attempted by several researchers in the past. Chaari et al. (2012) have formulated a hybrid technique for the path optimization of a mobile robot. They hybridized ant colony optimization with genetic algorithm (GA) to get a smart path while solving the global path optimization problem. Huang et al. (2015) have described a metaheuristic hybridization method for four wheeled mobile robots. They used a Taguchi based method to obtain an optimized path with obstacle avoidance. Contreras-Cruz et al. (2015) have given a novel thought to develop a hybrid algorithm to solve the path planning problem for mobile robots by combining the artificial bee colony algorithm with evolutionary programming algorithm. The path length obtained by the proposed method is compared with a classical road map method, and a better performance has been observed. Das et al. (2016) have proposed a hybrid method combining particle swarm optimization with gravitational search algorithm (GSA) for path planning of multiple mobile robots. The adaptive parameters exploration and exploitation have maintained the algorithm balanced by the proposed hybrid algorithm. Gigras et al. (2015) have focused on a hybrid technique to solve the path planning problems by using the metaheuristic methods like ant colony optimization and particle swarm optimization. They used the hybrid technique to optimize the path of a robot in cluttered environments avoiding the obstacles.

The literature citations suggest that the hybridization of standalone methods are primarily attempted with mobile robots. However, the use of hybrid techniques in case of humanoid robots is very rare to find. Based on the above research gap available, the objective of the current investigation is set as the design and implementation of a novel navigational controller that can be used to navigate single as well as multiple humanoid robots in a complex environment with optimization of path and time taken to reach the desired destination. In the current analysis, hybridization is attempted between regression analysis (RA) and adaptive particle swarm optimization (APSO). Humanoid NAOs are used in the current analysis for the navigational purposes.

2. Humanoid NAO

Humanoid NAO is a small sized programmable robot designed by Aldebaran Robotics group, France. NAO has been evolved in several versions, and version V4 has been used in the current work. The NAO is of 58 cm height, 5 kg weight and is equipped with several sensors (Kofinas et al. 2013) such as sonars, infrared sensors, accelerometers, gyroscope, force sensitive resistors, etc. Encoders are also fitted with the NAO so that the value of the joint torques and the force exerted on the ground can be recorded in any time. NAO can be coded and controlled with Python language with the help of choregraphe interface developed by the designers of NAO. In the current analysis, navigation is attempted using single as well as multiple NAOs. Figure 1 represents a typical humanoid NAO.

Figure 1: A typical humanoid NAO

3. General overview of regression analysis

Being a statistical approach of data forecasting, regression analysis serves as a method for establishing relationship among dependent and independent variables. In regression analysis, dependent variables are not represented directly linear to the independent ones; rather they are represented as linear to some variables that are related to the independent variables. A general equation of regression can be represented as follows.

$$y_i = a_1 + a_2 x_i + e_i\,, \quad i = 1,2,3,\ldots n. \tag{1}$$

In the above equation, y_i is a dependent variable and x_i is an independent variable with parameters as a_1 and a_2 and e_i represents an error form. Here, y_i is represented linear to the values of x_i. Figure 2 represents the scheme of a basic linear regression.

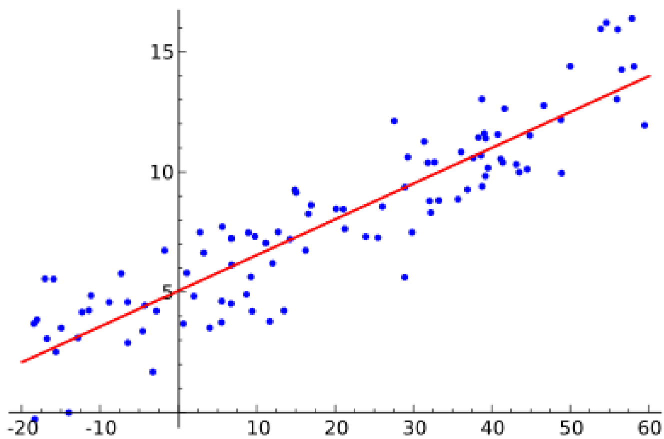

Figure 2: Representation of a linear regression

In the x-axis, ranges have been taken from -20 to 60. In the y-axis, ranges have been taken from 0 to 15. The straight line shows the basic equation of regression. The most significant feature of regression is the accumulation of the scattered data into an equation form. In context to navigational problem, the inputs of the problem can be fed to the regression controller and based on the previous training pattern; the controller generates an output solution.

4. General overview of adaptive particle swarm optimization

Robots are equipped with logics of artificial intelligent techniques to improve their decision taking ability. Particle Swarm Optimization is known to produce better results than other intelligent navigation techniques. PSO is a population based metaheuristic approach inspired by nature. It is adopted from the bird flocking or fish schooling behaviour. The group of entities that form the population in PSO are called as swarms, and the individual entities of the swarm are known as particles. As PSO is a population based method, it is assumed that all the particles in the population are moving in a definite search space and their position and velocity are recorded at each step. All the particles save their best position which is communicated through the entire population. During the optimization process, the velocity and position parameters are updated as the following governing equations.

$$\begin{aligned} v_i(j+1) &= wv_i(j) + c_1 ran1(P_{pbest} - p_i) + c_2 ran2(P_{glbest} - p_i), \\ p_i(j+1) &= p_i + v_i(j+1). \end{aligned} \tag{2}$$

To enhance the performance of basic PSO, some modifications are adopted in the control parameters which finally turns into APSO. The review of the literatures suggests that several researchers (Clerc & Kennedy 2002; Shi & Eberhart 1998) have attempted to modify different parameters of the basic PSO algorithm. However, in the APSO algorithm, three parameters such as inertial weight, social and cognitive parameters can be chosen as the governing parameters to increase the performance of the basic algorithm. The control of these three parameters are described in the subsequent sections of the paper.

4.1. Control of inertia weight

Inertia weight w is used to balance the search capabilities of local and global search of particles in the swarm. The value of w should be different in different search space. However, it is not advisable to increase the value of w with time. It is observed that the evolutionary factor f has an influence over w, and f value is large in the initial search and becomes small. Hence, it is favorable to allow w to follow a sigmoid function with the evolution factor f. Again taking in to account the value of w should be in between 0.4 and 0.9, the equation will be as follows.

$$w(f) = \frac{1}{1 + 1.4e^{-2.4f}}, \tag{3}$$

where $f = \dfrac{d_g - d_{\min}}{d_{\max} - d_{\min}} \in (0,1)$, d_g – the distance of best particle from the considered particle, $d_{\max}$ – maximum distance of best particle from the considered particle, $d_{\min}$ – minimum distance of best particle from the considered particle.

4.2. Control of acceleration coefficients

Here, an adaptive control is implemented by controlling the parameters c_1 and c_2. The parameter c_1 symbolizes self-cognition and c_2 symbolizes social-influence. Generally, c_1 tries to take the particle to its own historical best position obtained, while c_2 tries the global optimization of swarm that means it tries for the convergence of the swarm to its global best position. The total performance of the particles in the search space till getting the optimal position is divided in three scenarios named as investigation, introspection and convergence. In investigation scenario, all the particles are moving in the search space randomly to get their local optima. So we can increase the values of c_1 by 50% which helps the particles to find out their best positions. And decreasing the value of c_2 by 50% will help them not to create crowed situation as in this phase the primary aim is to find the local optima, not the global one. In introspection scenario, c_1 value is increased by 25% which will help the particles to get a local optima or P_{pbest}. At the same time, c_2 value can be increased by 10% so that it will move towards the global optima but not reach to the global optimum state. In convergence scenario, c_1 value can be decreased by 40% while increasing the c_2 value by 40% as till now all the swarms will have their local optima. So c_1 value has less effect on that, and by increasing the c_2 value, the global optima P_{pbest} can be obtained in this state.

To apply the APSO algorithm in the navigation problem, a fitness function has to be defined considering the optimization of the path. The proficiency of any path optimization algorithm is dependent on two aspects. The robot has to create a safe path by avoiding the obstacles, and the robot has to reach the goal position in shortest possible time with path optimization.

5. Control architecture for humanoid NAO

The proposed navigational controller is based on the hybridization of basic regression analysis with APSO algorithm. Hence, it considers the control architecture of both the algorithms.

5.1. Regression analyis (RA) controller

A humanoid robot navigation is based upon several governing parameters such as Left Obstacle Distance (LOD), Right Obstacle Distance (ROD), Front Obstacle Distance (FOD) and Turning Angle (TA). LOD, ROD and FOD act as the input parameters for the robot and TA acts as the output parameter. The microprocessor of the robot records the input data by the help of the sensors which sense and calculate the obstacle distances. Figure 3 represents the position of NAO in the environment with the obstacles present around.

Regression analysis is a basic technique of data forecasting in which previosu pattern is used to tackle a new situation. To implement the RA controller, the humanoid robot needs to be trained with a data pool. While designing the navigational controller for a humanoid robot, several reactive behaviours are also considered that help in optimization of the path. Those are obstacle avoidance behaviour, goal following behaviour and barrier following behaviour. In obstacle avoidance behaviour, whenever the sensors of the humanoid detect any obstacle

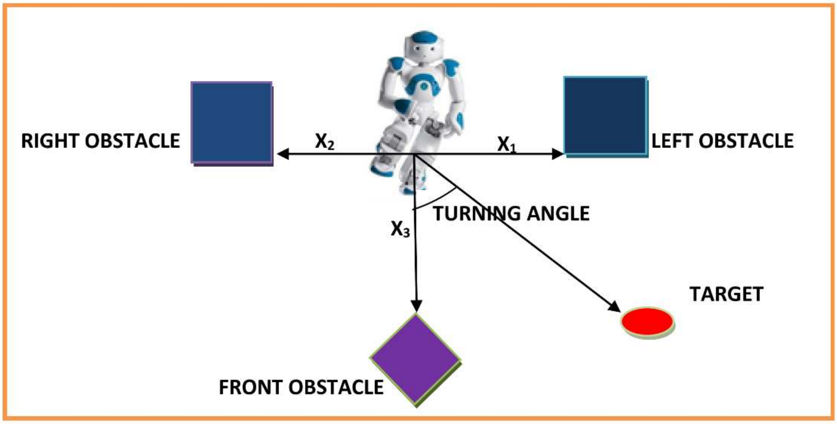

Figure 3: Initial position of the NAO in the environment

in their path, the humanoid takes the necessary turn to avoid the obstacle. In goal following behaviour, in absence of any obstacle in the path, the humanoid will always be directed towards the goal. The third one, barrier following behaviour is a complimentary behaviour. If in the path, a long barrier or wall is present and the goal of the humanoid is set at the end of the barrier, then, the humanoid simply follows the barrier without activation of the control algorithm. In this way, the humanoid reduces the consumption of energy and simultaneously optimizes the path.

A data pool of 500 samples is generated for the RA controller in which care is taken to provide almost all possible combinations of obstacle distances and necessary turning angle to the humanoid. The RA controller is fed with all the training pattern data, and the regression toolbox of the Minitab software (Khan 2013) is used to generate a standard equation of regression that can be used for the analysis. Minitab software takes into account the values of data pool as input and output parameters and based on the pattern of output and input data it generates a straight line equation which can also be termed as basic curve fitting. Figure 2 can also be referenced as the scheme of curve fitting. The equation generated for the current research is as follows.

$$K_4 = -23.0228 - 0.006183K_1 - 0.28508K_2 + 0.786367K_3\,, \tag{4}$$

where $K_1 = \text{FOD}$, $K_2 = \text{LOD}$, $K_3 = \text{ROD}$, $K_4 = \text{TA}$.

The above equation serves as the governing equation that can be used for navigation. The sensors of the humanoid measure the obstacle distances and the turning angle is generated as per the above equation.

5.2. Calculation of objective function using APSO

To apply any artificial intelligent algorithm for navigational purpose, it is required to formulate a fitness function/objective function based on the navigational parameters associated with the robot. The objective function formulated by using the APSO algorithm should satisfy two conditions; obstacle avoidance and target seeking. Based on the objective function only, the humanoids will move forward. While the humanoid starts its journey towards the target, it will map its shortest distance to the target and take steps one by one without applying any navigational algorithm. While its sensors detect any obstacle, it will stop and the implemented algorithm works which optimizes the path by avoiding the obstacles present in the path and generating a collision free path. Figure 4 illustrates the path of the humanoid in the environment by following the reactive behaviours.

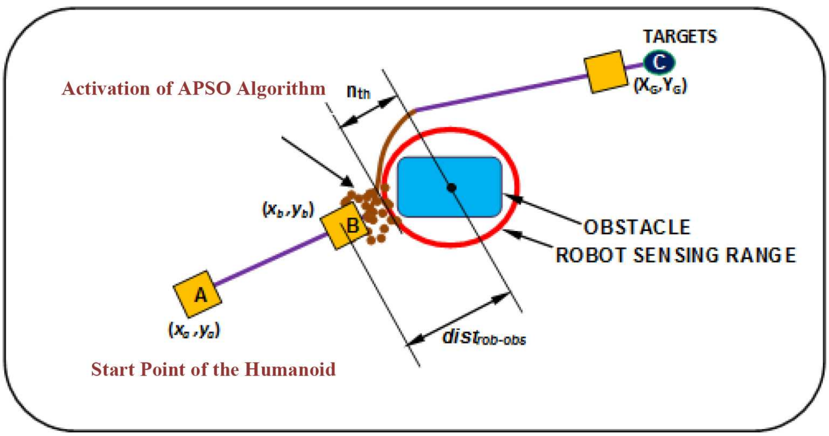

Figure 4: Path of the humanoid using APSO controller

Let the humanoid is at point 'A' to begin the journey to the target 'C'. The starting coordinate position of the humanoid is marked as A (x_a, y_a). From the same point, it starts its journey and reaches to point B (x_b, y_b) without using any navigational algorithm. After getting sensor information regarding presence of obstacle near the positon 'B', it implements the proposed algorithm to decide the next step. After point 'B', many possible points are marked in the figure which will be a next position for the humanoid, but it goes through the optimized one with the help of APSO technique. In the swarm, the position of particle having minimum value of objective function is treated as the optimal position for the humanoid robot to take a next step. While designing the path for the humanoid robot, the objective function is influenced by two conditions; avoiding collision with obstacles present in the environment and reaching to the target/goal in optimum path.

5.2.1. Objective function to avoid collision with obstacles

This objective function is designed to avoid the head on collision between obstacles (like walls, balls or any dynamic object) and humanoid robot. By considering the distance between the humanoid robot and obstacles, this is formulated and defined as follows:

$$f_{obstacle} = \begin{cases} 0, & \|Q_i - Obs_j\| > n_{th}\,, \;\; j = 1,2,\ldots,m, \\ \dfrac{1}{\|Q_i - Obs_j\|} - \dfrac{1}{n_{th}}\,, & \|Q_i - Obs_j\| \leqslant n_{th}\,, \;\; j = 1,2,\ldots m. \end{cases} \tag{5}$$

where Q_i – the position of the i-th particle in the swarm (i.e., the possible optimal step/position), obs_j – the center of the j-th obstacles, n^{th} – the threshold distance between the humanoid to obstacles, m – the numbers of obstacles.

In the current work, the threshold distance is chosen as 35 cm. The main objective here is to minimize the value of objective function f_{obstacle} to get an optimized position. The maximum allowable value of the distance between robot and obstacle in n_{th} to avoid collision with obstacles. As per the equation, we will get a non-zero value of objective function while the humanoid robot is at a very nearer position to the obstacles; otherwise, it will be zero.

Here we have considered the nearest obstacle present to the robot and the same can be evaluated by the following equation:

$$Dist_{rob-obs_j} = minimum \left\| Rob_b - Obs_j \right\|, \quad j = 1,2,\ldots m, \tag{6}$$

where Rob_b denotes the position of humanoid at point ‘B’, which will be varied according to the various environmental conditions.

5.2.2. Objective function for target seeking behaviour

The main aim in designing this objective function is to reach the target with a minimum path, and that is accomplished by considering the distance between the humanoid robot and its goal. The function is formulated as follows.

$$f_{target} = \|Q_i - G\|, \tag{7}$$

where Q_i – the position of the i^{th} particle in the swarm (i.e., the possible optimal step/position), and G – position of the target.

Here, the objective is to minimize the distance between the target and the possible optimal positions of the humanoid robot. So the total objective function to optimize the path optimization strategy of the humanoid can be formulated by combining the above two objective functions as follows:

$$f_m = \theta f_{obstacle} + \phi f_{target}\,, \tag{8}$$

where θ and ϕ represents the parameters that influence the path of humanoid robot, f_m denotes the fitness value for the M numbers of population. Here, the controlling parameters or weight parameters (θ and ϕ) have been chosen by trial and error method, and their values are kept between 0 to 1.

The value of $f_{obstacle}$ is zero until any obstacle is not in the range of the sensor of humanoid robot, and it will navigate freely to the target without any hurdles. But after detection of an obstacle, the $f_{obstacle}$ value becomes a non-zero quantity. In this condition, the APSO algorithm will be executed by the humanoid robot. Here, inside the algorithm, the particles with their fitness values will be arranged in ascending order and particle having lowest objective function value is taken as the new best position for the humanoid robot.

6. Design of Petri-Net controller to avoid inter-collison

Peterson (1981) has demonstrated the design of a Petri-Net controller that is required to avoid inter-collision among multiple robots in a dynamic environment. In the current analysis, three humanoid NAOs are considered where each NAO acts as a dynamic obstacle to the other. Figure 5 represents a standard method for design of a Petri-Net controller.

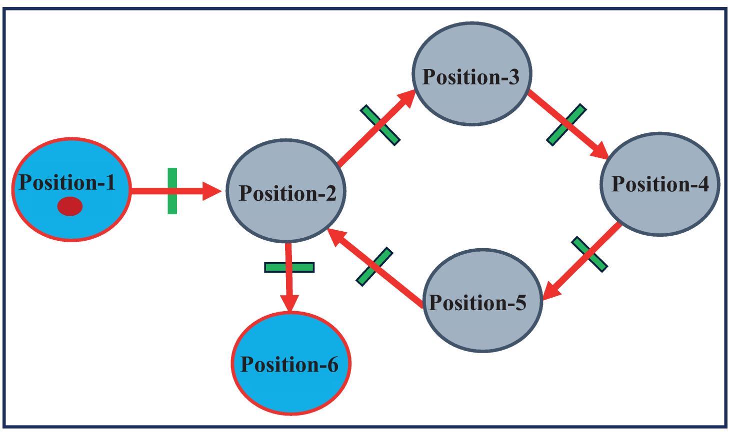

Figure 5: Designed Petri-Net controller for multiple humanoid robots

Position of a robot is represented as a circle, state of transition is shown by a bar symbol, token location represents the current position of the robot. Six positions of the robot are described in the model. In Petri-Net model, there is a token at the first position. It is assumed that all the robots are standing randomly in the environment initially without knowing each other's positions. After starting the journey to reach the target, they will try to avoid the obstacles and trace one another, which is the "Position-2" of the model. Detecting the dynamic obstacles comes in "Position-3". At this position, they have to set the priority of the robot about which one to wait and which one to move first. The priority should be given to the robot having a less distance left to reach the target. The robot with higher priority goes further, and the other one has to stop (treated as static obstacle) until first robot leaves the place. That indicates the negotiating situation of the two robots denoted as “Position-4” in the model. Next step of the robots after negotiating is to check for any other conflicting situations. If there is no such situation, the robot will move forward. That condition is named as “Position-5”. The last situation “Position-6” is a waiting condition. If any robot encounters two other robots already in a situation of conflict (both the robots in “Position-3”), then by giving them a higher priority, the robot has to behave as a static obstacle as having lower priority. After both the other robots start their journey towards their goal, this robot in “Position-6” will then start its journey taking the “Position-2”. Considering all these conditions, multiple robots can move in a clutter environment easily without making any complexity in their path which optimizes the time taken to reach the goal.

7. RA-APSO hybrid controller

To increase the efficiency of the standalone algorithms, the hybridization technique has been designed and implemented in the navigational controller. In the current analysis, RA controller is hybridized with APSO controller. For the regression analysis, there are three numbers of input parameters (FOD, LOD and ROD) and one output parameter, i.e. turning angle. To hybridize the algorithm with APSO, the output from RA controller is taken as the input for the APSO algorithm, i.e. named as interim turning angle (ITA). With ITA, the instant values of FOD, LOD, and ROD are also taken as the input values for APSO method. The final output from the APSO controller is final turning angle (FTA) that guides the humanoids to navigate smoothly in the environment. Figure 6 represents the scheme of hybridization.

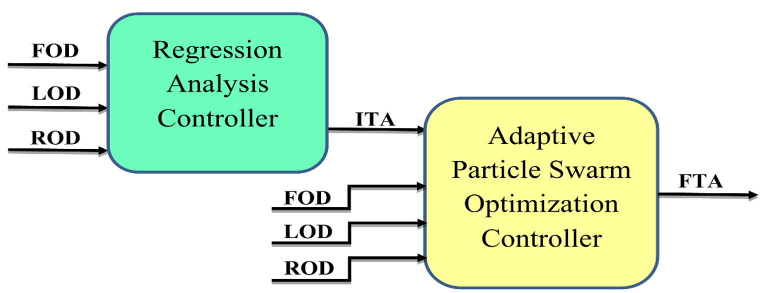

Figure 6: Proposed RA-APSO navigational controller

It can be noticed that the proposed controller works on a two-step hybridization basis. Initially, after detection of obstacle in the path, the input parameters are fed to the RA controller and the output of the RA controller is again fed to the APSO controller for generation of the final output.

The steps for the hybridization process are discussed below.

Step-1: Define the start and goal position of the humanoid robot.

Step-2: Navigate the robot towards its target until it is obstructed by any obstacle.

Step-3: Call RA algorithm when any obstacle comes in the target path.

Step-4: Sensory data regarding FOD, LOD, ROD are fed to the RA controller, and output (ITA) is evaluated.

Step-5: Initialize, the swarm of particles in the search field of the humanoid robot by taking ITA as input parameter.

Step-6: Evaluate the fitness value for every particle; sort the particles in ascending order, if the fitness value is better than the best fitness value of ITA Set current value as the new ITA.

Step-7: Choose the particle with the best fitness value of all the particles as the ITA, for each particle calculate particle velocity and position. And calculate next optimum position, Calculate FTA.

Step-8: Move the humanoid robot to the new optimal position according to the FTA.

Step-9: Repeat the step-4 to step-8 until the humanoid robot reaches to the target by avoiding all the obstacles.

Figure 7 represents the pseudo code for the RA-APSO algorithm.

```
Start
        Feed the input parameters to the RA controller
        Calculate the ITA
        Feed ITA to APSO algorithm and activate APSO Controller
        For each particle
        Initialize the particles in the swarm

        Do
        For each particle
                Calculate fitness value
                If
                  The fitness value is better than the best fitness value P_pbest
                  Set current value as the new P_pbest
                Else
                  Calculate the fitness value again
        End
        Do
        Choose the particle with the best fitness value of all the particles as the P_gbest
          For each particle
                Calculate Particle velocity by equation
                Calculate Particle position by equation
        End
While maximum iteration is reached or minimum error condition satisfied
Stop
```

Figure 7: Pseudo code of RA-APSO algorithm

The detailed process of humanoid navigation can be represented by a flow chart. Figure 8 represents the flowchart for the navigation of humanoids by the proposed RA-APSO algorithm.

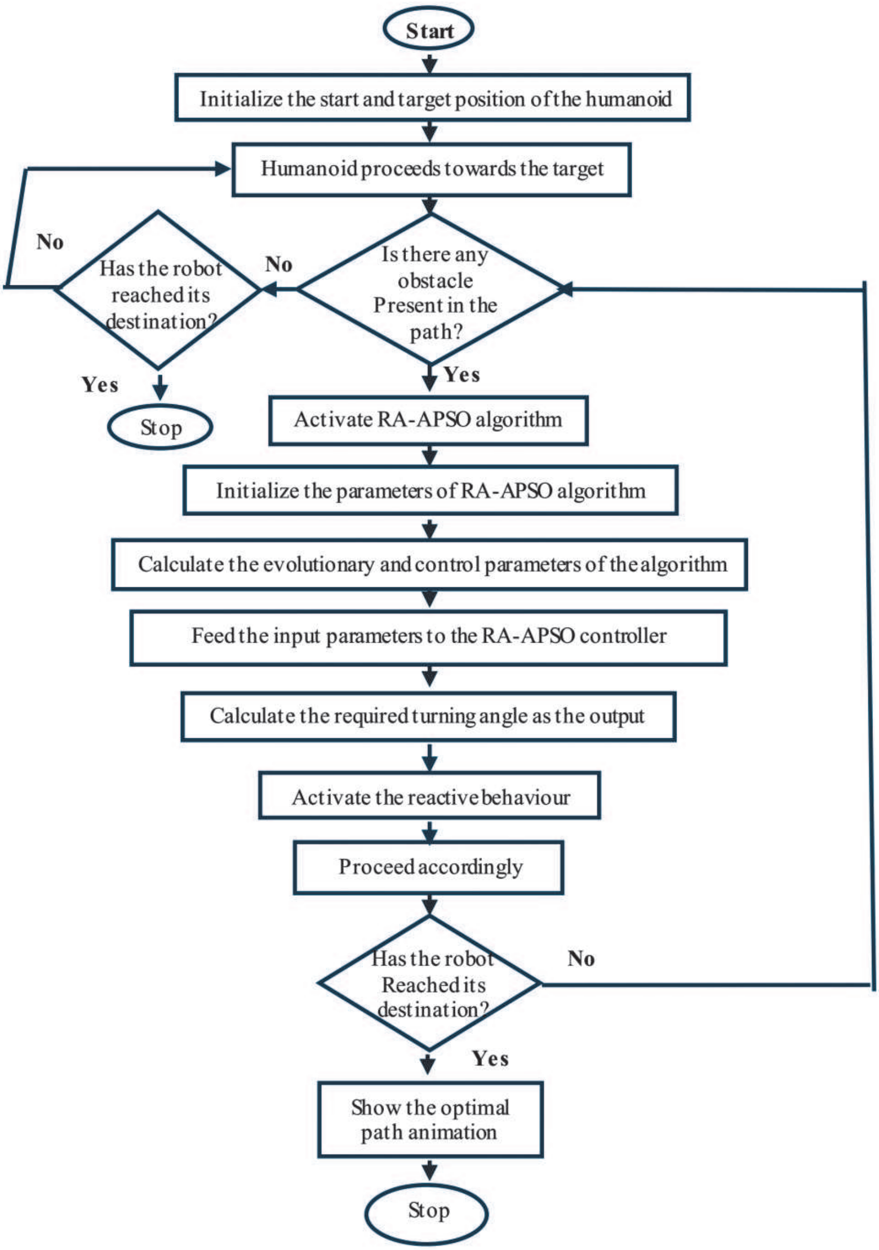

Figure 8: Flowchart of the control scheme of RA-APSO controller

8. Navigation using RA-APSO controller

After designing the control architecture for the path planning of the humanoid robots and the Petri-Net controller for avoiding inter-collision among multiple humanoids, the proposed navigational controller was tested for both simulation and experimental environments. It is to be noted that the Petri-Net controller is required when multiple humanoids navigate simultaneously in a common platform. For the navigation of a single humanoid NAO, it is not required. The purpose of this section is to check the RA-APSO navigational controller for a simulation as well as experimental platform. Finally, after the execution of the controller in both the platforms, a comparison is aimed for the validation between the simulation and experimental results.

8.1. Navigation of Single Humanoid NAO in a Complex Environment

Several number of simulation software have been developed over the past years. In the current work, V-REP is chosen as the simulation software to be used. The idea behind using V-REP as the simulation software is that for humanoid navigation, it works as an easy and suitable software. V-REP follows the programming language LUA based on the ANSI C language. Specific unique properties like collision detection, better motion planning and calculation of shortest path makes V-REP more potential candidate than other software. To analyze the effectiveness of regression navigational controller, a static environment has been created in the V-REP software. It has to be kept in mind that the working of the navigational controller must be based on the reactive behaviours such as obstacle avoidance, goal following and barrier following. For navigational analysis of the humanoid NAO, an environment has been created in the simulation window of V-REP software. The environment size was chosen as 200×250 units with five numbers of static obstacles. By considering the reactive behaviours and logic of RA-APSO controller, a program has been written and implemented in the NAO humanoid. After implementing necessary rules and regulations, obstacle avoidance, goal following behaviours were tested. The main objective of navigational analysis is to observe the shortest path calculation and time taken to reach the desired target. Figure 9 represents the simulation results obtained from the V-REP software.

It can be observed from Figure 9a that initially, the NAO was set at a source point and a specific destination point was provided. The two blue boxes represent the source point and target point. Five number of static obstacles were set at random positions. It was observed that by using the proposed RA-APSO navigational controller, NAO was able to avoid all the obstacles those were present in the path and reach the desired target safely. During the journey from Figure 9a to Figure 9g, it can be observed that the humanoid has followed the shortest possible path. The distance covered by the humanoid to reach the destination and

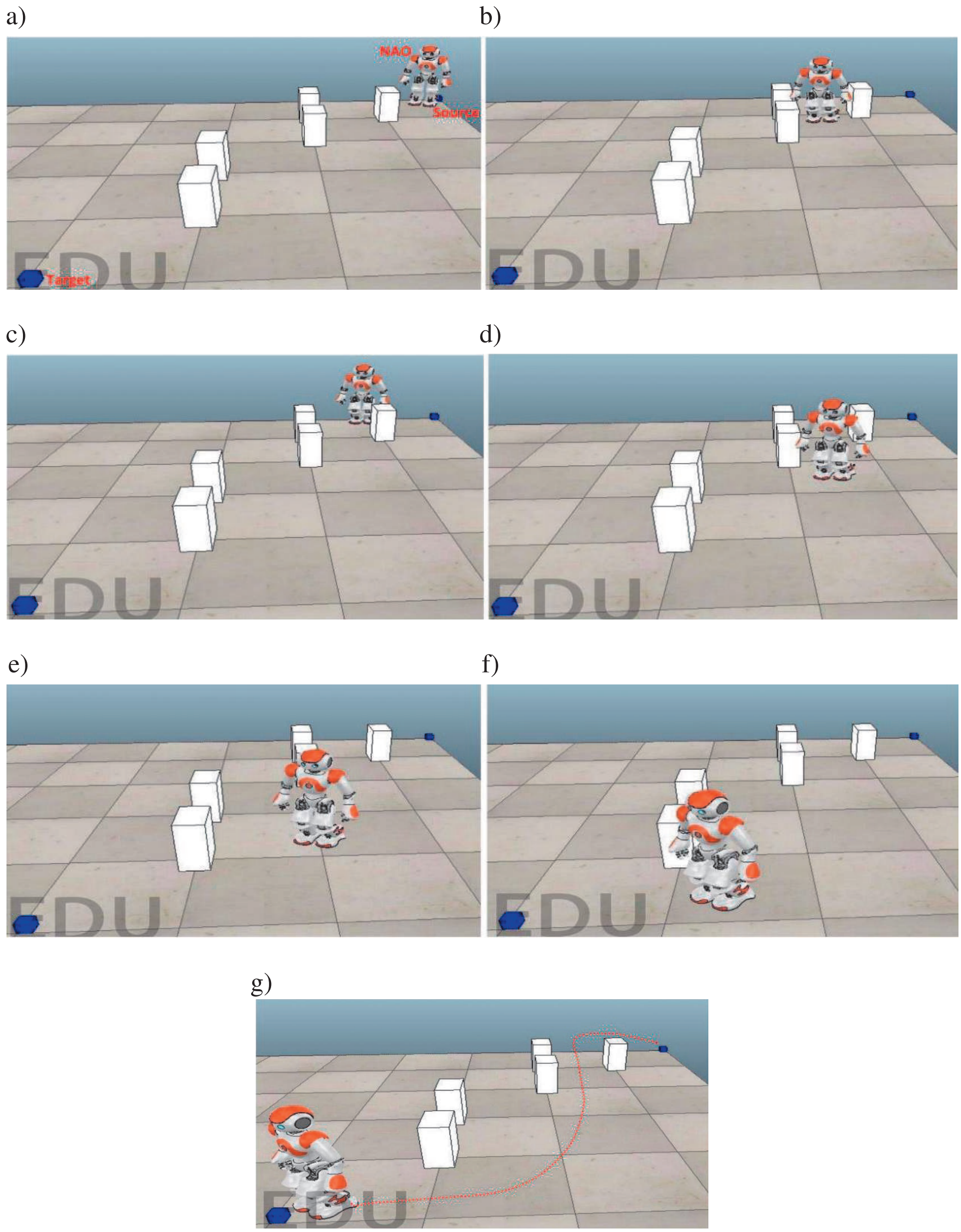

Figure 9: Illustration of navigation of single NAO in simulated environment

the time consumed to reach the target were noted from the V-REP simulation window itself and recorded for further analysis.

To validate the effectiveness of the proposed controller, it is important to repeat the simulation results in an actual environment. By creating an exactly similar environment under laboratory set up, the simulation results can be compared for practical implementation. A similar environment as was in the case of the simulation software was created under laboratory set up. To maintain the same environment size, the actual platform to conduct the experiment was chosen as 200×250 centimeters. Five numbers of static obstacles were selected and placed at same positions as that of the simulation environment. An initial and final point of the experiment was decided. By using the logic of the reactive behaviours and RA-APSO controller, a program was written and implemented in the humanoid NAO. In the actual environment, the NAO was operated by a Wi-Fi control. The robot location is defined according to the movement of the robot from the start to the target point, position of the target, location of the corners and the sidebars of the arena. After the environment was set up, the navigation of NAO was observed and analyzed. Figure 10 represents the actual experiment that was performed in our laboratory.

The two blue boxes represent the source and target positions in the Figure 10a. Five numbers of static obstacles as represented by white boxes were set at the exact similar positions as was in case of simulation. After the environment was set up, the NAO was started for its navigation. It was observed that the humanoid NAO was able to avoid all the obstacles that were present in the path and reach the desired target safely. It can be observed from Figure 10a to Figure 10f, that the humanoid has followed the shortest possible route. In the actual environment, the path length from source to target as travelled by the humanoid was measured by using a measuring tape, and a stopwatch measured the time taken to reach the target. The path length and time taken were noted and recorded for the comparison between the simulation and experimental results. As stated earlier, the effectiveness of the proposed navigational controller can only be checked by the proper comparison between the simulation and experimental results regarding the navigational parameters, which are the path length and time taken. Table 1 and Table 2 represent the comparison between the simulation and experimental results for path length and time taken respectively. It is to be noted that quite large number of experiments were performed for the navigational control of humanoid NAO using regression analysis and only a few have been analyzed here.

From Tables 1 and Table 2, it can be noticed that the navigational parameters for the experiments always show higher values than the simulation results. The simulation results are ideal ones where there are no errors like loss of data transmission, effects of friction, etc. When the humanoid navigates in a practical environment, it is influenced by several external factors like loss in Wi-Fi data transmission, presence of friction, slipping effects at the contact point between the foot of the humanoid and floor, etc. These factors increase the navigational parameters to some extent. After recording both the simulation and experimental results, the percentage of errors were calculated. It was observed that in all

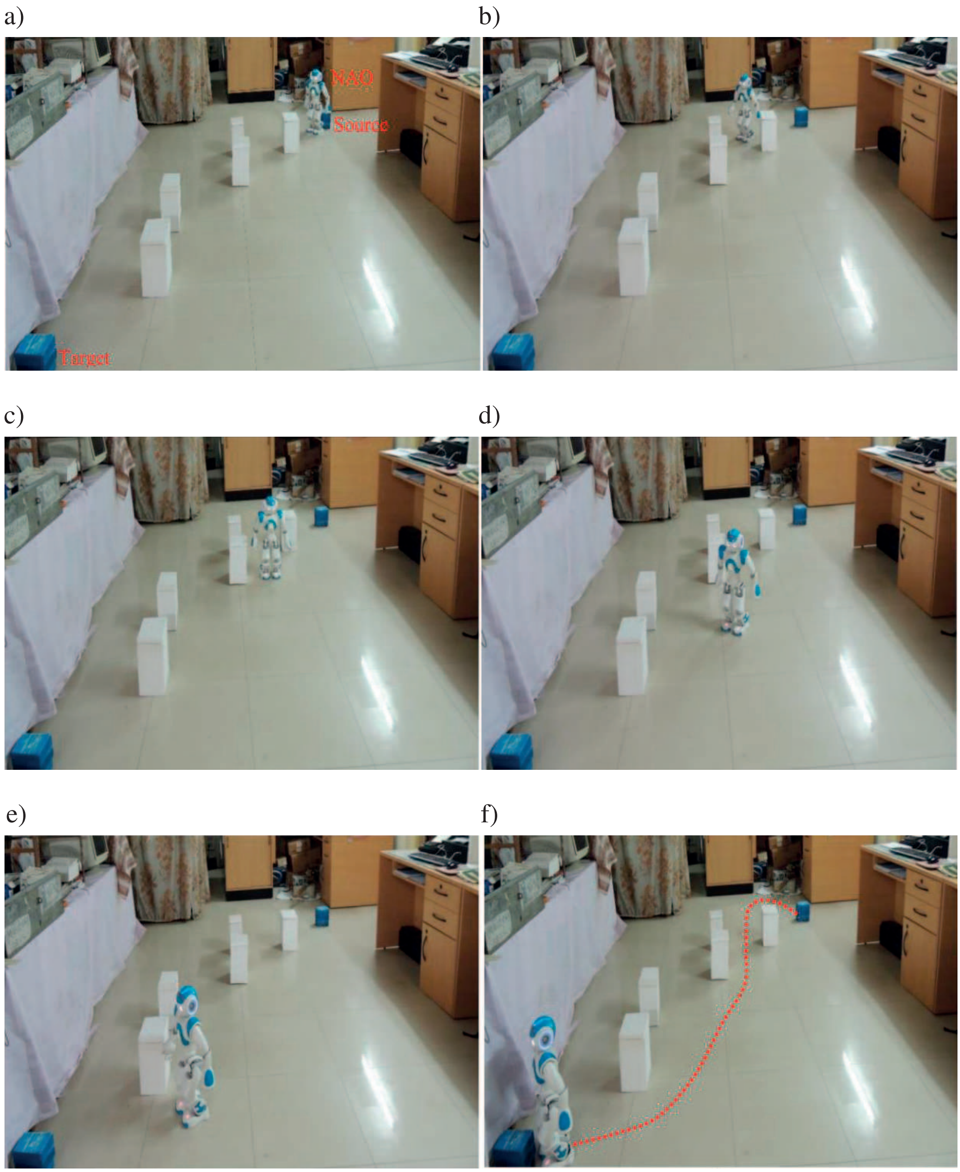

Figure 10: Illustration of navigation of single NAO in experimental environment

cases, the error percentage was within 7%, which is well below the acceptable limit.

Path planning and navigation of multiple humanoids is way too challenging than the navigation of a single humanoid. The reason behind the above case is that, in single humanoid problem, the environment is a static one and when multiple humanoids navigate, it becomes a dynamic one. In a dynamic environment,

Table 1: Comparison of path length between simulation and experiment for navigation of single NAO

No. of runs	Simulated path length (cm)	Experimental path length (cm)	Error in %
1	279.52	294.8	5.18
2	279.75	293.0	4.52
3	279.04	291.9	4.4
4	277.52	292.8	5.22
5	277.79	294.4	5.64
6	278.98	291.5	4.3
7	280.32	293.6	4.52
8	278.28	292.2	4.76
9	277.92	291.3	4.59
10	278.7	293.3	4.98
Average	278.782	292.88	4.811

Table 2: Comparison of time taken between simulation and experiment for navigation of single NAO

No. of runs	Time required in simulation (sec)	Time required in experiment (sec)	Error in %
1	35.14	37.5	6.29
2	35.54	36.85	3.55
3	34.46	35.81	3.77
4	35.12	37.15	5.46
5	33.58	35.23	4.68
6	34.37	35.44	3.02
7	35.84	37.53	4.5
8	33.98	36.18	6.08
9	33.47	35.17	4.83
10	34.24	36.14	5.26
Average	34.574	36.5	4.765

each humanoid has to avoid the static obstacles that are present in the path and the dynamic fellow humanoids, which are navigating, simultaneously in the same platform.

8.2. Navigation of multiple humanoid NAOs in a complex environment

For the simulation of multiple humanoids, V-REP was again selected as the simulation software. In the current work of navigation of multiple humanoid robots, we have considered three humanoid NAOs in a single environment. The environment size is kept exactly same as was done for single humanoid navigation. Four static obstacle are considered in the analysis at random positions. Each humanoid has its own predefined source and goal position. It has to be kept in mind that the rules of RA-APSO navigational controller can avoid the obstacles but not decide regarding the priorities if a conflicting situation arise. Therefore, along with the RA-APSO navigational controller, the logic of the Petri-Net controller is also considered in the current problem. The working pattern of the Petri-Net controller has been already described in the previous sections. The environment size for multiple humanoid navigation is kept as 200×250 units with four numbers of static obstacles. Along with the static obstacles, each humanoid acts as a dynamic obstacle for the other two. The three humanoid NAOs (denoted as N1, N2 and N3) have their predefined source or start positions (denoted as S1, S2 and S3) and goal or target positions (T1, T2 and T3). A program has been written in the LUA language using the combined logic and rules of regression navigational controller and the Petri-Net controller and implemented in all the humanoids. After the setting of the environment, the three humanoids started their journey to reach their respective goal positions. Figure 11 illustrates the simulation environmental set up for the navigation of multiple humanoids and the navigation of reach humanoid to their respective goals.

It can be observed from Figure 11a that each humanoid has been marked with their start and goal positions. Then, they started their journey to their respective goals. In the Figure 11a to Figure 11g, it can be observed that all the humanoids have avoided both the static as well as dynamic obstacles and reached their targets safely.

To validate the results of simulation analysis, a practical experimental setup was developed in the laboratory conditions as was done in case of navigation of single humanoid robot. The platform size was chosen as 200×250 centimeters. Four numbers of static obstacles were set at similar positions that of the simulation. A program was written by using the logic of regression navigational control and the Petri-Net controller and implemented in all the humanoid NAOs. After the practical platform was ready, all the humanoids started their journey towards their respective targets. Figure 12 represents the actual setup that has been used for the experiments and the navigational pattern followed by the multiple humanoid robots.

From Figure 12a, it can be noticed that all the three humanoids are marked with their corresponding start and goal positions. After the start signal, all of them moved forward towards their respective goal positions. It can be seen from

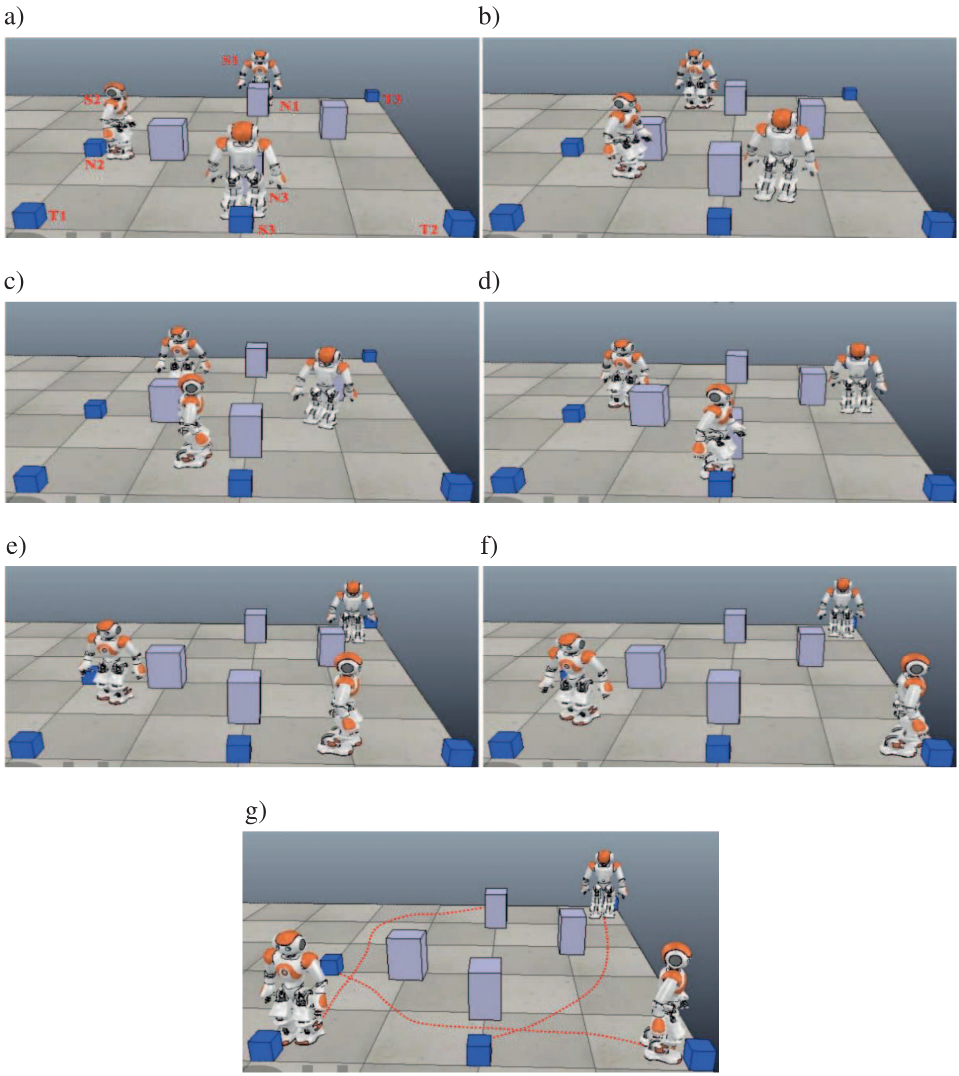

Figure 11: Illustration of navigation of multiple NAOs in simulated environment

Figure 12a to Figure 12g, all the humanoids have avoided the static obstacles and the inter-collision between them. The navigational parameters such as path length and time taken are measured in the similar way as was done in case of single humanoid robot i.e. by measuring tape and stopwatch respectively. Finally, a comparison was done among the simulation and experimental results, and the data are presented in Table 3 and Table 4. Table 3 represents a comparison of path length between the simulated and experimental environments, and Table 4 represents the comparison for time taken between simulated and experimental environments.

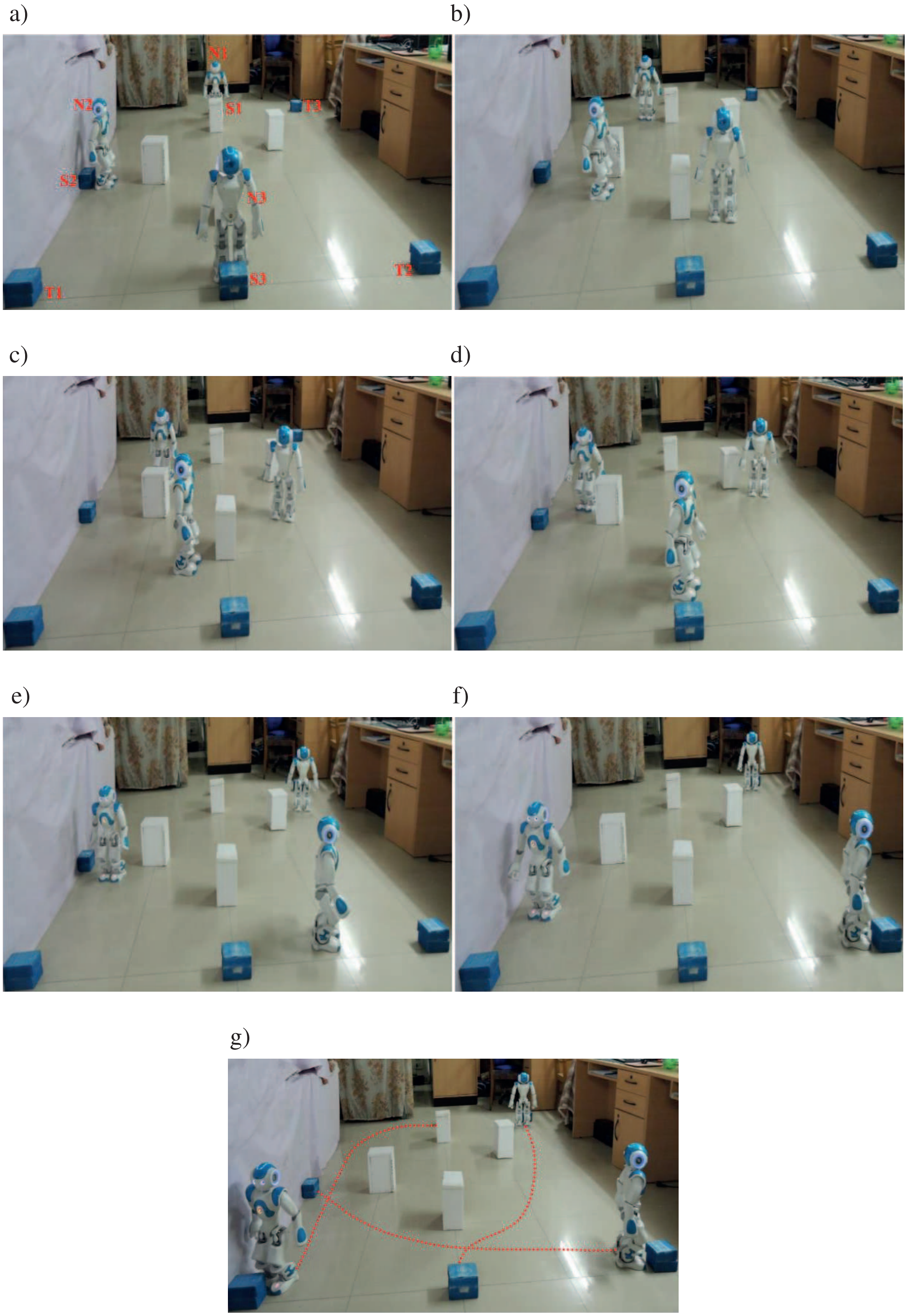

Figure 12: Illustration of navigation of multiple NAOs in experimental environment

Table 3: Comparison of path length between simulation and experiment for navigation of multiple NAOs

No. of runs	Simulation results			Experimental results			Errors in %		
	Path travelled in cm								
	N1	N2	N3	N1	N2	N3	N1	N2	N3
1	263.15	216.1	246.8	277.1	228.7	258.1	5.03	5.51	4.38
2	263.34	215.2	245.7	277.2	228.6	258.9	5	5.86	5.1
3	263.47	215.4	246.1	278.3	228.4	259	5.33	5.69	4.98
4	262.7	215.7	246.2	276.8	229.1	258.4	5.09	5.85	4.72
5	262.86	216.12	245.8	276.9	229.3	259.1	5.07	5.75	5.13
6	264.04	216.35	245.6	277.4	228.5	257.4	4.82	5.32	4.58
7	264.17	215.98	246	277.3	229	257.3	4.73	5.69	4.39
8	262.65	217.35	245.5	278.2	229.7	257.8	5.59	5.38	4.77
9	263.9	216.45	245.3	276.7	229.4	258.6	4.63	5.65	5.14
10	263.7	216.9	247.1	276.5	228.2	259.2	4.63	4.95	4.67
Average	263.398	216.155	246.0	277.24	228.9	258.4	4.99	5.56	4.79

Table 4: Comparison of Time Taken between Simulation and Experiment for Navigation of Multiple NAOs

No. of runs	Simulation Results			Experimental Results			Errors in %		
	Path Travelled in cm								
	N1	N2	N3	N1	N2	N3	N1	N2	N3
1	32.45	27.14	31.4	34.7	28.2	33.6	6.48	3.76	6.55
2	32.6	27.2	31.5	34.8	29.2	33.6	6.32	6.85	6.25
3	32.74	26.42	30.8	35.1	27.8	33.2	6.72	4.96	7.23
4	31.87	26.6	30.9	33.55	28.6	32.5	5.01	6.99	4.92
5	31.94	27.41	30.4	33.6	28.7	33.4	4.94	4.49	8.98
6	33.19	27.53	30.1	35	29.4	31.4	5.17	6.36	4.14
7	33.33	26.8	30.7	34.9	28.4	32.5	4.5	5.63	5.54
8	31.99	28.4	30	34.22	30.2	31.5	6.52	5.96	4.76
9	33	27.6	29.7	35.2	28.8	32.7	6.25	4.17	9.17
10	33.22	28	31.7	35.6	29.5	33.9	6.69	5.08	6.49
Average	32.63	27.31	30.72	34.67	28.88	32.83	5.87	5.44	6.43

It can be observed that the navigational parameters show a higher value in experimental results as compared to the simulation results. The reason for the same has already been discussed. The percentage of errors for all the comparisons are within 7%, which is well under the acceptable limit. From the implementation of the proposed RA-APSO navigational controller in humanoid navigation, it can be observed that the proposed controller is able to navigate single as well as multiple humanoids in cluttered environments. The results of the navigation obtained from the simulation software are verified with areal time experimental set up, and both the results are in good agreement with each other with very minimal percentage of errors.

9. Comparison of the RA-APSO controller with existing navigational controller

From the above sections, it was observed that the proposed RA-APSO navigational controller was successfully implemented in both single and multiple humanoid robots. The humanoids were perfectly able in avoiding both static as well as dynamic obstacles and reach their goal position safely. However, to have a detailed investigation regarding the efficiency of the proposed navigational controller, it is required to compare it with other existing techniques. To do the same, a Co-Evolutionary Improved Genetic Algorithm (CEGA) and an Improved Genetic Algorithm (IGA) are chosen. CEGA and IGA are heuristic methods as compared to the regression analysis, which is a statistical method. CEGA and IGA work on the basis of a predefined objective function while regression analysis is based on statistical formula and training data. Qu et al. (2013) developed two methods named as Improved Genetic Algorithm and CEGA. In the current analysis, navigation of single robot was compared with IGA algorithm and navigation of multiple robots was compared with CEGA algorithm. Figure 13 and

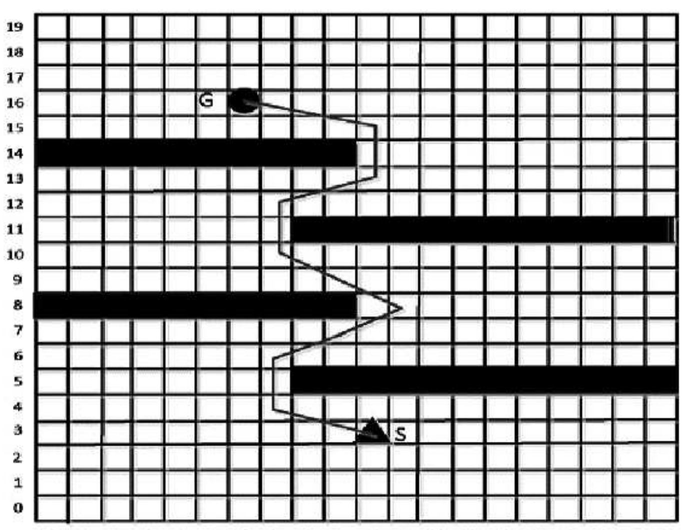

Figure 13: Simulation result for navigation of single robot using IGA algorithm

Figure 14 demonstrate a comparison between IGA and proposed technique for navigation of a single robot. Figure 15 and Figure 16 demonstrate the comparison between CEGA and proposed navigational controller for navigation of multiple robots.

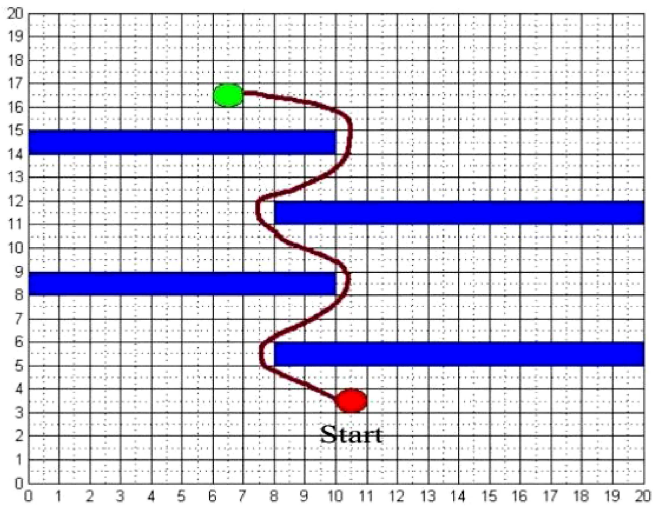

Figure 14: Simulation result for navigation of single robot using RA-APSO controller

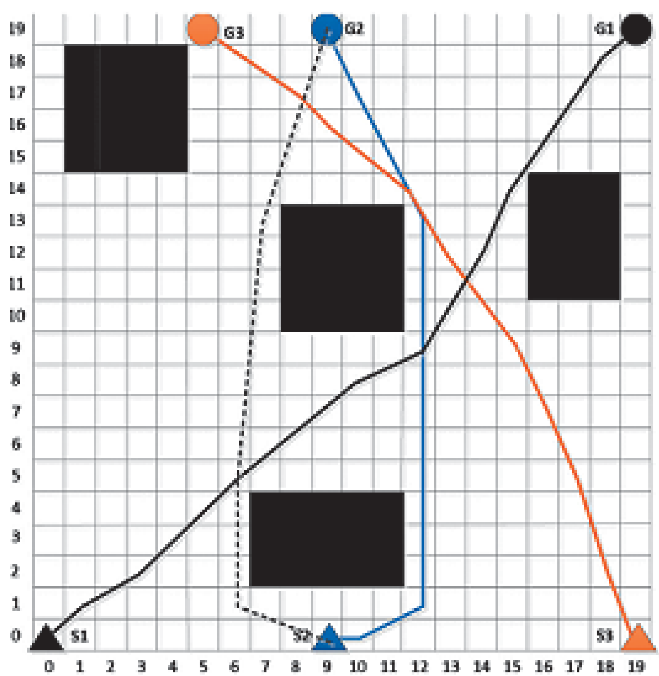

Figure 15: Simulation result for navigation of multiple robots using CEGA approach

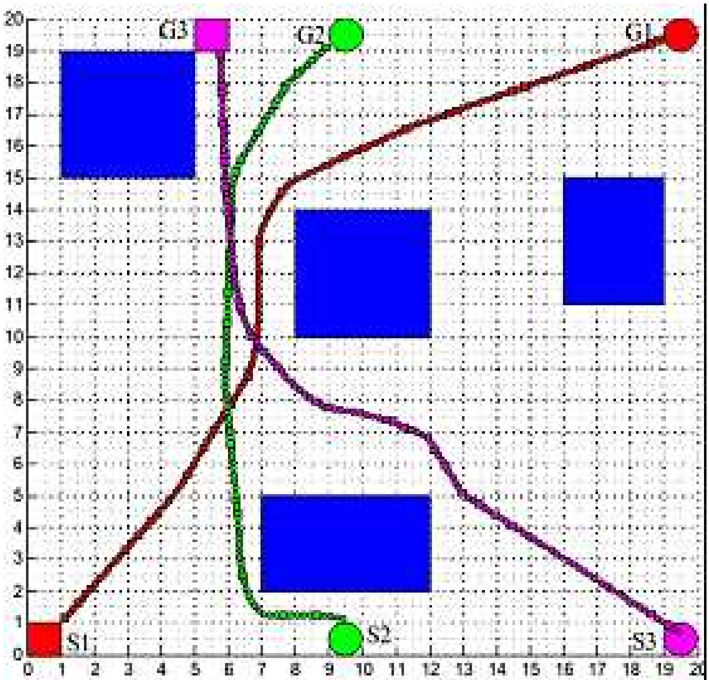

Figure 16: Simulation result for navigation of multiple robots using RA-APSO controller

The navigational parameters such as the path length and time taken are calculated for the respective existing and proposed technique, and a comparison was done between them. Table 5 and Table 6 represent the comparison for path length between existing method and proposed method for single and multiple robot navigation respectively.

Table 5: Comparison of the results obtained from Qu et al. [32] and proposed RA-APSO navigational controller for navigation of single robot

Technique used	Path length in cm	Deviation in %
By Qu et al. (2013) (Figure 13)	25.89	7.11
By RA-APSO Technique (Figure 14)	24.05	

From the obtained results, it was noticed that the RA-APSO technique has demonstrated a better path optimization than the existing ones. Hence, it can serve as a better alternative than the existing techniques. So, the efficiency of the proposed RA-APSO navigational controller is on an enhanced mode than the existing methods.

Table 6: Comparison of the results obtained from Qu et al. [32] and proposed RA-APSO navigational controller for navigation of multiple robots

Technique used	Path length in cm	Deviation in %
By Qu et al. (2013) (Figure 15)	28.19	6.88
By RA-APSO Technique (Figure 16)	26.25	

10. Conclusions

With the ever increasing demand towards industrial automation, path planning and navigation for robots has emerged as one of the most promising area of research. The humanoids robots with their flexibility of replacing the human efforts can be used in several sectors if trained intelligently. In this paper, a novel navigational approach has been designed for the path planning of humanoid robots. As the navigation of humanoid robots is a relatively new area of research in its own kind, hybridization of standalone methods is very rarely available in the published literatures. In the current investigation, a novel navigational controller has been designed by hybridizing basic regression analysis with adaptive particle swarm optimization. The proposed controller was tested in both simulated and experimental environments considering the navigational parameters of the humanoid robots. The results obtained from both the environments were compared with each other, and a good agreement between them was found. To navigate multiple humanoids in a common environment, a Petri-Net controller was designed and implemented along with the logic of RA-APSO controller. The controller was tested for single as well as multiple humanoid robots. Finally, the proposed navigational controller was also tested with the other existing navigational techniques, and an enhanced performance was observed. Therefore, the proposed controller can be used as a robust technique for navigation of humanoids as well as other forms of robots. This research would definitely add a new dimension towards robotics field dealing with navigational analysis.

References

[1] I. Châari, A. Koubaa, H. Bennaceur, S. Trigui and K. Al-Shalfan: Smartpath: A hybrid aco-ga algorithm for robot path planning, *IEEE Congress on Evolutionary Computation (CEC)*, (2012), 1–8.

[2] M. Clerc and J. Kennedy: The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE transactions on Evolutionary Computation*, **6**(1), (2002), 58–73.

[3] D. CLEVER and K. MOMBAUR: An inverse optimal control approach for the transfer of human walking motions in constrained environment to humanoid robots, In *Robotics: Science and Systems* (2016).

[4] M.A. CONTRERAS-CRUZ, V. AYALA-RAMIREZ and U.H. HERNANDEZ-BELMONTE: Mobile robot path planning using artificial bee colony and evolutionary programming, *Applied Soft Computing*, **30**, (2015), 319–328.

[5] S. DALIBARD, A. EL KHOURY, F. LAMIRAUX, A. NAKHAEI, M. TAÏX and J.P. LAUMOND: Dynamic walking and whole-body motion planning for humanoid robots: an integrated approach, *The International Journal of Robotics Research*, **32**(9–10), (2013), 1089–1103.

[6] P.K. DAS, H.S. BEHERA and B.K. PANIGRAHI: A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning, *Swarm and Evolutionary Computation*, **28**, (2016), 14–28.

[7] L. FEN, Z. JIANG-HAI, S. XIAO-BO, Z. PEI-YING, F. SHI-HUI and L. ZHONG-JIE: Path planning of 6-DOF humanoid manipulator based on improved ant colony algorithm, *24th Chinese Conference on Control and Decision (CCDC)* (2012), 4158–4161.

[8] Y. GIGRAS, K. CHOUDHARY and K. GUPTA: A hybrid ACO-PSO technique for path planning, *2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, (2015), 1616–1621.

[9] H.C. HUANG: A Taguchi-based heterogeneous parallel metaheuristic ACO-PSO and its FPGA realization to optimal polar-space locomotion control of four-wheeled redundant mobile robots, *IEEE Transactions on Industrial Informatics*, **11**(3), (2015), 915–922.

[10] V. HUGEL and N. JOUANDEAU: Walking patterns for real time path planning simulation of humanoids, In *RO-MAN*, (2012), 424–430.

[11] J. IDO, Y. SHIMIZU, Y. MATSUMOTO and T. OGASAWARA: Indoor navigation for a humanoid robot using a view sequence, *The International Journal of Robotics Research*, **28**(1), (2009), 315–325.

[12] O. KANOUN, J.P. LAUMOND and E. YOSHIDA: Planning foot placements for a humanoid robot: A problem of inverse kinematics, *The International Journal of Robotics Research*, **30**(3), (2011), 476–485.

[13] P. KARKOWSKI, S. OSSWALD and M. BENNEWITZ: Real-time footstep planning in 3D environments, *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, (2016), 69–74.

[14] R.M. KHAN: *Problem solving and data analysis using minitab: A clear and easy guide to six sigma methodology*, John Wiley & Sons (2013).

[15] N. KOFINAS, E. ORFANOUDAKIS and M.G. LAGOUDAKIS: Complete analytical inverse kinematics for NAO, *13th International Conference on Autonomous Robot Systems (Robotica)*, (2013), 1–6.

[16] R. MIRJALILI, A. YOUSEFI-KOMA, F.A. SHIRAZI and S. MANSOURI: Online path planning for SURENA III humanoid robot using model predictive control scheme, *4th International Conference on Robotics and Mechatronics (ICROM)*, (2016), 416–421.

[17] P.K. MOHANTY, D.R. PARHI, A.K. JHA and A. PANDEY: Path planning of an autonomous mobile robot using adaptive network based fuzzy controller, *IEEE 3rd International on Advance Computing Conference (IACC)*, (2013), 651–656.

[18] P.K. MOHANTY and D.R. PARHI: Optimal path planning for a mobile robot using cuckoo search algorithm, *Journal of Experimental & Theoretical Artificial Intelligence*, **28**(1–2), (2016), 35–52.

[19] P.K. MOHANTY and D.R. PARHI: Path planning strategy for mobile robot navigation using MANFIS controller, In *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*, (2014), 353–361.

[20] K. NISHIWAKI, J. CHESTNUTT and S. KAGAMI: Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor, *The International Journal of Robotics Research*, **31**(11), (2012), 1251–1262.

[21] D.R. PARHI and M.K. SINGH: Navigational strategies of mobile robots: a review, *International Journal of Automation and Control*, 3(2–3), (2009), 114–134.

[22] N. PERRIN, O. STASSE, F. LAMIRAUX and E. YOSHIDA: Weakly collision-free paths for continuous humanoid footstep planning, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (2011), 4408–4413.

[23] J.L. PETERSON: Petri net theory and the modeling of systems (1981).

[24] H. QU, K. XING and T. ALEXANDER: An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots, *Neurocomputing*, **120**, (2013), 509–517.

[25] S.H. RYU, Y. KANG, S.J. KIM, K. LEE, B.J. YOU and N.L. DOH: Humanoid path planning from hri perspective: A scalable approach via waypoints with a time index, *IEEE transactions on cybernetics*, 43(1), (2013), 217–229.

[26] M. SADEDEL, A. YOUSEFI-KOMA and M. KHADIV: Offline path planning, dynamic modeling and gait optimization of a 2D humanoid robot, *Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, (2014), 131–136.

[27] A.J. SCHMID and H. WOERN: Path planning for a humanoid using NURBS curves, *IEEE International Conference on Automation Science and Engineering*, (2005), 351–356.

[28] R. SHAKIBA, M. NAJAFIPOUR and M.E. SALEHI: An improved PSO-based path planning algorithm for humanoid soccer playing robots, *3rd Joint Conference on AI & Robotics and 5th RoboCup Iran Open International Symposium (RIOS)*, (2013), 1–6.

[29] Y. SHI and R. EBERHART: A modified particle swarm optimizer. In Evolutionary Computation Proceedings, *IEEE World Congress on Computational Intelligence*, (1998), 69–73.

[30] Y. SHIMIZU and T. SUGIHARA: Efficient path planning of humanoid robots with automatic conformation of body representation to the complexity of environments, *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, (2012), 755–760.

[31] M.K. SINGH, D.R. PARHI and J.K. POTHAL: ANFIS approach for navigation of mobile robots, *International Conference on Advances in Recent Technologies in Communication and Computing*, (2009), 727–731.

[32] M.K. SINGH and D.R. PARHI: Path optimisation of a mobile robot using an artificial neural network controller, *International Journal of Systems Science*, **42**(1), (2011), 107–120.

[33] J.K. YOO and J.H. KIM: Gaze control-based navigation architecture with a situation-specific preference approach for humanoid robots, *IEEE/ASME Transactions on Mechatronics*, **20**(4), (2015), 2425–2436.