

An Evolutionary Computational Approach to Humanoid Motion Planning

Regular Paper

Dhammika Suresh Hettiarachchi^{1,*} and Hitoshi Iba²¹ Dept. of Electrical Engineering & Information Systems, Graduate School of Engineering, The University of Tokyo² Dept. of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo

* Corresponding author E-mail: suresh@iba.t.u-tokyo.ac.jp

Received 15 May 2012; Accepted 26 Jul 2012

DOI: 10.5772/51905

© 2012 Hettiarachchi et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract The theme of our work is centred on humanoid motion planning and balancing using evolutionary computational techniques. Evolutionary techniques, inspired by the Darwinian evolution of biological systems, make use of the concept of the iterative progress of a population of solutions with the aim of finding an optimally fit solution to a given problem. The problem we address here is that of asymmetric motion generation for humanoids, with the aim of automatically developing a series of motions to resemble certain predefined postures. An acceptable trajectory and stability is of the utmost concern in our work. In developing these motions, we are utilizing genetic algorithms coupled with heuristic knowledge of the problem domain. Unlike other types of robots, humanoids are complex in both construction and operation due to their myriad degrees of freedom and the difficulty of balancing on one or more limbs. The work presented in this paper includes the adopted methodology, experimental setup, results and an analysis of the outcome of a series of evolutionary experiments conducted for generating the said asymmetric motions.

Keywords Genetic Algorithm, Asymmetric Motion, Humanoid

1. Introduction

Irrespective of the form and configuration of a robot, the determination of its trajectory is performed through the application of forward dynamics and inverse-kinematics. Forward dynamics is used for the simulation and real-time feedback control of a robot; hence, is used by many robotics simulators. In forward dynamics calculations, current joint coordinates and their first derivatives with respect to time are known at a given instant. A calculation finds the joint coordinate values and their time-derivatives at a later sampling instant, using the available information mentioned above. On the other hand, the application of inverse kinematics is conceptually simpler to grasp. However, inverse kinematics involves non-linear simultaneous equations and there can be multiple, and sometimes an infinite number of solutions or even no solutions at all. Nevertheless, inverse kinematics is essential for the computed-torque control of robots and robotic manipulators. In inverse kinematics, a time-history of the joint coordinates is given, and from that knowledge plus the architecture and inertial parameters of the robotics system, the torque or force requirements at the different actuated joints are determined. However, the application of inverse kinematics to complex robots,

especially ones with higher degrees-of-freedom, such as humanoids, is a challenging process.

The balancing of a humanoid is generally governed by complex equations that are specific to the generated motion pattern. A well-known approach for balance control is the Zero Moment Point (ZMP) method [1]. However, the ZMP calculation requires precise knowledge of the robot's dynamics, the location of the centre of mass and also the inertia for all those links in motion. The Inverted Pendulum Model (IPM) is another approach that requires relatively limited knowledge about dynamics and is inapplicable to on-the-fly motion planning. It has been shown that IPM falls short for dynamically-varied motion paths, unlike with predefined motion paths. As such, hybrid approaches that combine ZMP and IPM have also been proposed [2]. The most significant problem with the said balancing methods is the need for precise knowledge of the humanoid's dynamics, which is sometimes unavailable. Furthermore, the ZMP method is not directly applicable for applications where the robot's feet have no contact with the ground (e.g., hand standing) [3]. Moreover, efforts have been made to generate motion using Interactive Evolutionary Computation (IEC), which does not require specialized knowledge of humanoid robotics [4]. Nonetheless, the interactive methods have the characteristic drawback of subjective evaluation. On the other hand, evolutionary methods have several advantages over other methods. Specifically, Genetic Algorithms (GA) alleviate the need for ZMP- and IPM-based methods, and they do not require the developer to perform inverse and forward kinematics calculations [5]. Furthermore, the developer should have only limited knowledge about robot dynamics when evolutionary methods are employed.

Our work makes multiple contributions to the domain of robot motion planning. We present a methodology for the natural and intuitive design of humanoid controllers inspired by biological evolution, while empowering the designer by relieving her of advanced knowledge of dynamics. Moreover, this novel method is a relatively simpler approach than ZMP- or IPM-based balancing. We demonstrate that the proposed method has the ability to generate asymmetric motion trajectories which are more complex than their symmetric counterparts and which have the ability to control a humanoid with a higher DOF than in previous attempts. The heuristically-driven GA-based method solves the search space complexity of the problem at hand and brings out the potential for developing any type of humanoid motion controller using our method. The goal of our work is twofold; first, we propose a methodology for developing asymmetric motion controllers for humanoids using GA. Second, we test our method by trying to evolve three asymmetric yoga motions. A binary hill-climbing approach is also attempted for solving the same problem as the baseline approach.

In section II, we present the architecture of the proposed evolutionary system. Subsequently, the adopted methodology and the results of a series of experiments will be presented. Finally, the conclusion will be followed by a discussion of future work.

2. The System Architecture

The asymmetric motion trajectories selected in this work are the three yoga poses shown in Figure 1. The first posture is called the frontal leg-raise (*Aabhimukhya Vikasitha Paadasana* in Sanskrit), the second posture is called the warrior pose (*Weerabhadraasana*) and the third posture is called the hand-to-toe pose (*Uththita Hastha Paadangusthaasana*).



Figure 1. Target motion trajectories

The world of yoga provides an extensive set of standard asymmetric motions that essentially require the practitioner to maintain their balance while stretching various parts of the body to acquire the desired poses. Yoga motions vary in complexity - the more complex motions being difficult for imitation by humans let alone robots [6]. The limitations of humanoids, including their limited flexibility and inadequate degrees of freedom, make it difficult to program them to perform these complex motions [7]. Nevertheless, our preliminary research revealed that a considerable number of yoga poses could be performed by humanoids, at least in theory. We selected Fujitsu's HOAP-2 robot as the target humanoid platform to apply the evolved programs. A concise specification of HOAP-2 is given in Table 1.

We selected the robotics simulation environment 'webots' partially due to the availability of a virtual HOAP-2 model [8] in the software package that alleviated the burden of developing a physics-based HOAP-2 simulation from scratch. The simulator allowed for the creation and direct compilation of controller programs written in C/C++ while providing a fast simulation mode to expedite graphics mode processing. Figure 2 shows the abstract schematic of the system.

Height	50 cm
Weight	7 kg
Joint Mobility	6DOF/foot * 2, 4DOF/arm * 2, 1DOF/waist, 1DOF/hand * 2, 2DOF/neck
Sensors	Joint Angle Sensor, Foot Load Sensor

Table 1. HOAP-2 design specification

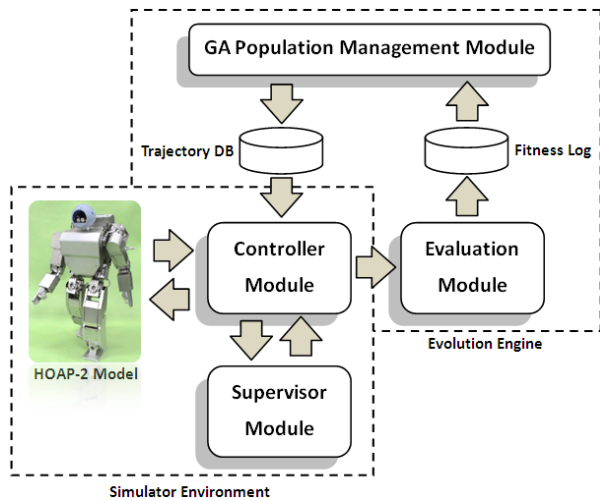


Figure 2. The abstract schematic of the system

The evolutionary approach is explained in the next section.

3. The Evolutionary Approach

One important design decision in formulating yoga motions is the encoding method for a joint controller. Naturally, the servos corresponding to each joint of HOAP-2 operate within a continuous range, comprising real-valued angles given in radians. As such, the behaviour of a joint can be represented as a series of real values in the time domain for constructing the chromosome required for evolution. Although a real valued chromosome might be a straightforward mapping from the real-world representation of a HOAP-2 motion file, there is a disadvantage to this approach, namely, a bloated search space that may be costly in terms of computational time and efficiency. On the other hand, a joint can be represented as a sequence of transition points and types of motions. These motions are merely primitive actions, such as “decrease angle”, “increase angle” or “maintain angle”, represented by the triplet $\{-1, 0, +1\}$. The chromosome’s representation for a joint controller is shown in Figure 3. Note that there is one chromosome for each joint and the cell indices represent the absolute time intervals.

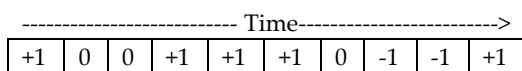


Figure 3. The ternary representation of a joint trajectory

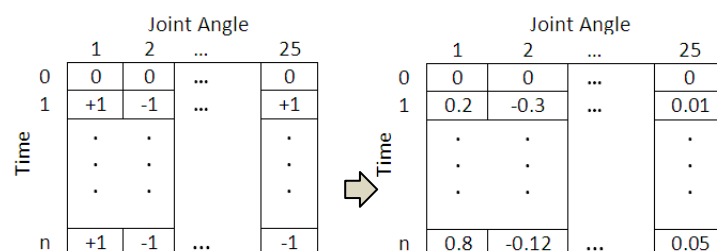


Figure 4. Ternary file (left) and the real-valued motion file (right)

We make use of this representation scheme for all the joints of the HOAP-2 model. The collection of twenty-five such ternary vectors is used to construct a single, two-dimensional motion file for the humanoid. This ternary file should be processed to convert it into a real-valued motion trajectory prior to its application to the HOAP-2 model. This conversion is illustrated in Figure 4.

A motion file is considered as an individual in our genetic algorithm and the initial population is constructed in a purely random manner. We use the conventional genetic algorithm, as illustrated in Figure 5.

Some important GA parameters used in our work are illustrated in Table 2. The crossover and mutation probabilities were empirically determined.

The number of active joints subjected to optimization varies depending upon the type of motion. The number of active joints for the first motion is 11, while the second and third postures make use of 14 and 15 joints, respectively.

The fitness of a posture is determined based on the type of motion. The general criteria for measuring the fitness of a controller has two aspects, as follows.

1. Stability of the humanoid
2. The proximity of the evolved posture to the desired yoga pose

In a previously conducted GA-based symmetric motion generation experiment [3], the stability criterion was not taken into consideration in determining fitness. Rather, only the resultant final posture of the forward dynamics calculation was evaluated. Although the stability of the humanoid can be found in several contrasting ways, we use the simple intuitive method of reading the value of the ground-touching foot load sensor. Ideally, the sensor reading should return approximately 70 N, since a stable posture indirectly means resting the whole body mass (7 kg) on the ground-touching foot. If not for this intuitive method, complex calculations are required to determine stability by projecting the centre of gravity onto the horizontal plane and checking the boundaries of the ground contacts to ascertain the inclusion of the projection coordinates within those boundaries. In fact, this calculation adds further overhead to the fitness calculation when the large populations and number of maximum generations are considered. Unstable postures are allocated a penalty of -1 in order to increase the selection pressure on the population.

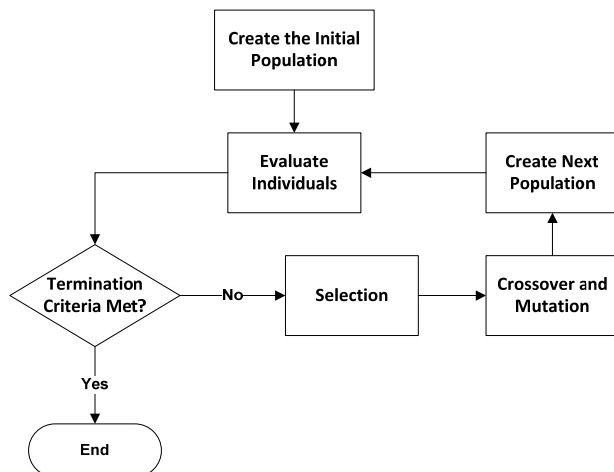


Figure 5. The genetic algorithm

Parameter	Value
Population Size	200
Maximum Generations	100
Chromosome Size	25 * 20 (Two-Dimensional)
Selection	Tournament
Crossover	Multipoint, Uniform
Crossover Probability	0.9
Mutation Probability	0.3
Elitism	Best 10 (5%)

Table 2. Parameters for the genetic algorithm

The fitness functions for the three motions are given in equations 1, 2 and 3, where d is the distance between two joints, lh is the left hand, rh is the right hand, lk is the left knee, lf is the left foot, h is the head, bc is the body's centre and $lhip$ is the left hip. The joint layout of HOAP-2 is illustrated in Figure 6.

A single component subscript such as y in lh_y means the position of the left hand in the y -direction, while a multiple-component subscript like $lh_{x,y,z}$ means the three dimensional position of the left hand in the global coordinate system. The operation abs returns the absolute value of its argument. Figure 7 shows some important metrics of the fitness evaluation. The encircled labels show the Euclidean distance between critical joints.

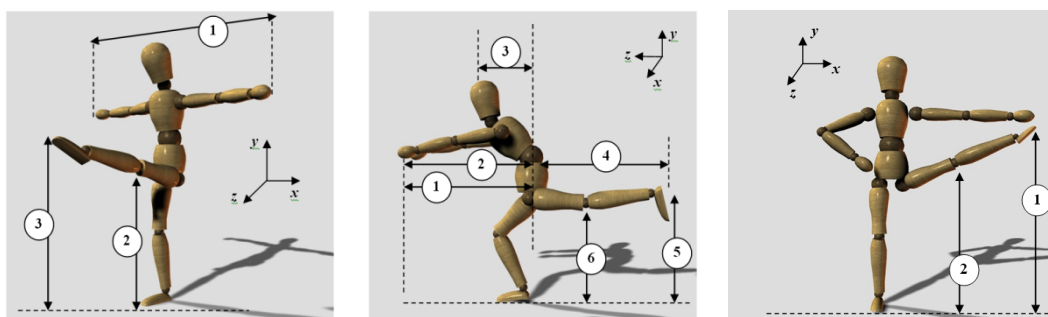


Figure 7. Metrics for fitness evaluation

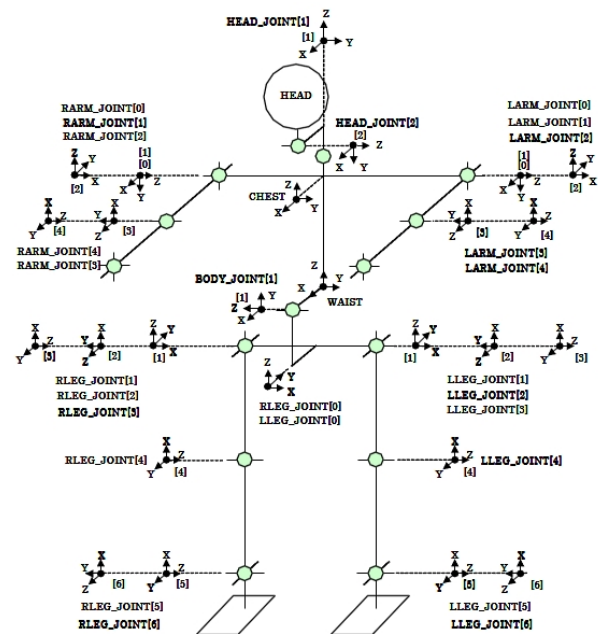


Figure 6. Joint layout of HOAP-2

$$\begin{aligned}
 \text{frontal_leg_raise} = & d(lh_{x,y,z}, rh_{x,y,z}) \\
 & + lk_y + lf_y - abs(d(lh_z, rh_z)) - abs(d(lh_y, rh_y)) \quad (1) \\
 & + d(lf_z, lhip_z)
 \end{aligned}$$

$$\begin{aligned}
 \text{warrior} = & d(lh_z, bc_z) + d(rh_z, bc_z) + d(h_z, bc_z) \\
 & + d(bc_z, lf_z) + lf_y + lk_y - abs(d(lh_x, rh_x)) \quad (2) \\
 & - abs(d(lh_y, rh_y)) - abs(d(lh_z, rh_z))
 \end{aligned}$$

$$\begin{aligned}
 \text{htot} = & lf_y + lk_y - d(rh_{x,y,z}, bc_{x,y,z}) - abs(d(lh_z, lf_z)) \\
 & - abs(d(re_z, rh_z)) - abs(d(lf_z, bc_z)) \quad (3)
 \end{aligned}$$

The conventional approach to calculating the distances based on forward dynamics and the mapping of local coordinates to global coordinates is both complex and computationally expensive. Therefore, we used a virtual GPS device - originally intended for the localization of wheeled-robots - for tracking the joint positions in the global coordinate system.

The search space for all yoga poses is inherently large. For example, we aim to optimize 11 parameters (angles) across 20 time steps for the first motion - i.e., a chromosome of length 220 - giving rise to an overwhelming 3^{220} possible combinations. Therefore, some form of prior knowledge is more than helpful in locating an acceptable solution. For instance, the mirroring of

movement for similarly behaving joints - such as arms or hips (just before raising a leg) - can reduce the required number of effective parameters. Moreover, the use of heuristics for the mutation operator gives a sense of direction for the search, practically reducing the complexity of the search space. The heuristics used for the three motions are shown in Table 3.

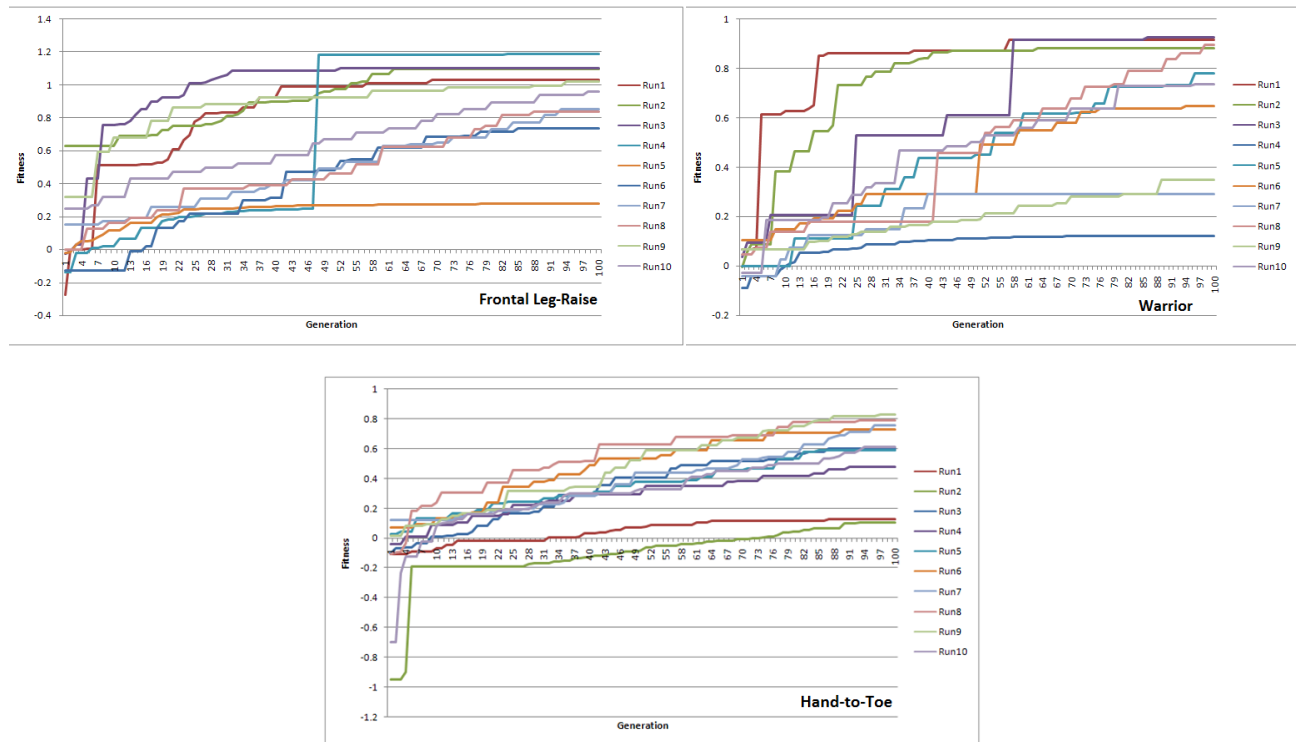


Figure 8(a). Fitness variation for ten independent runs for the three motions

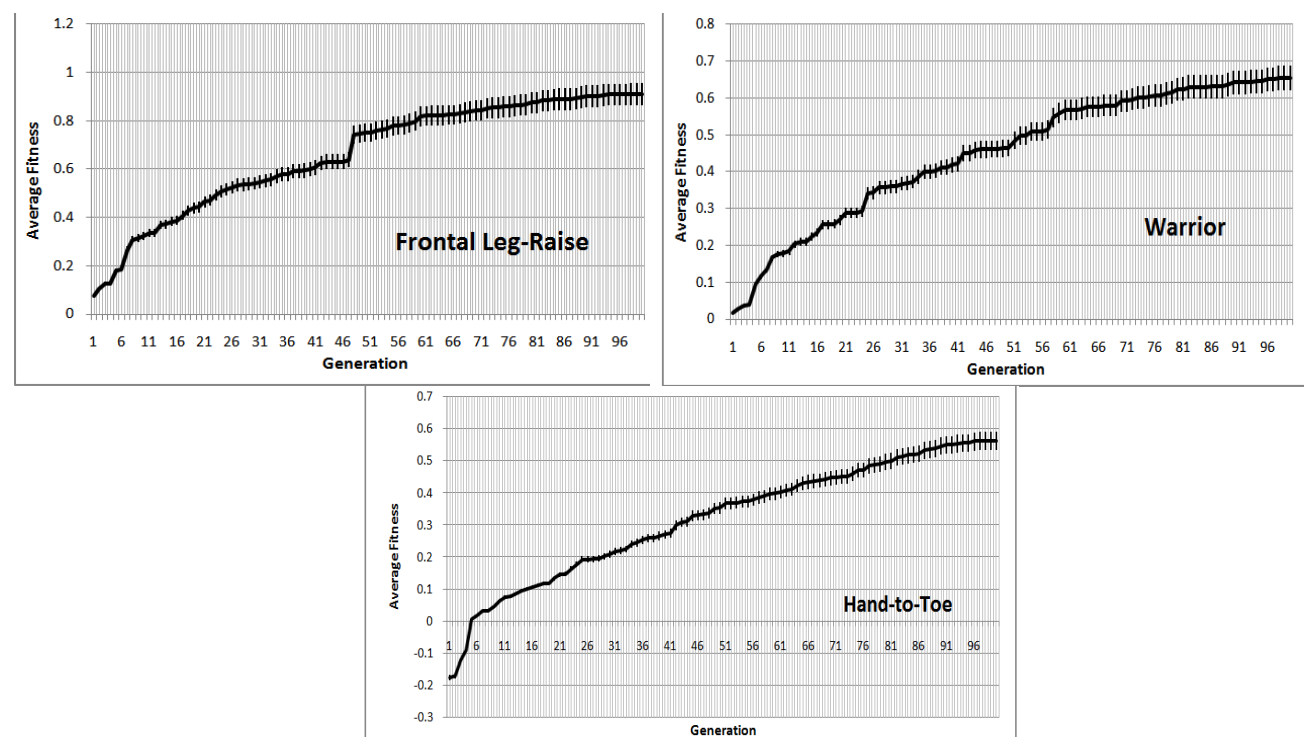


Figure 8(b). Mean fitness variation vs. generation

Frontal Leg-Raise	The behaviour of the left hand is mirrored for the other. For monotonously increasing or decreasing angles, mutated values are set as either +1 or -1.
Warrior and Hand-to-Toe	The above heuristics plus setting narrow ranges to low-varying angles. Tilting the upper body forward to maintain balance.

Table 3. Heuristics for the three Yoga motions

4. Results

To ascertain the effectiveness of our approach in finding a yoga posture as close as possible to the target motion, we performed ten independent evolutionary runs for each of the three postures. The fitness variation for these independent runs and the mean fitness for the three motions (with standard error) are illustrated in Figure 8(a) and 8(b), respectively.

The mean fitness for the frontal leg-raise converged to a value close to 0.9, while the warrior pose converges to a value close to 0.65 after one hundred generations. When independent runs are considered, the best run for the frontal leg-raise produced an individual with a fitness value of approximately 1.18, when the best run for the warrior produced an individual with a fitness of 0.92. The best run for the hand-to-toe motion could create a controller with a fitness of 0.8. However, the overall progress of this last motion was inferior compared to the first two.

The behaviour of the best individual in the best evolutionary run for each of the three trajectories is shown in Figure 9. All three motions proved stability while demonstrating satisfactory trajectories.

The best-evolved individuals in the simulator environment were applied on the real humanoid to ascertain real-world stability and robustness. Prior to applying to the real robot, the motion files having radian values for the joint servos had to be converted into motion files with pulse data. The formula for conversion is shown in equation (4):

$P = A \times \Theta_d$, where $A = \pm 209$ and

$$\Theta_d = \left(\frac{180}{\Pi} \right) \times \Theta_{rad} \quad (4)$$

Next, as the robot works in constant time steps of 2ms, the pulse file was interpolated before applying to the real robot. Linear interpolation to 10,000 time steps as well as 20,000 time steps was tried out and the 20,000 time step

interpolation demonstrated a smoother trajectory. Nevertheless, the 10,000 point interpolation could also show robustness in the real-world. Figure 10 shows how HOAP-2 performed when the best-evolved programs were applied on it.

Although the humanoid demonstrated real-world robustness, a peculiar anomaly could be observed in both motions. While trying to lift the leg, the heel of the lifting leg touched the ground and made the whole body rotate counterclockwise in all three yoga motions. However, HOAP-2 could maintain his balance despite this collision with the ground. This particular collision was not observed in the simulator environment. In order to steer clear of the ground, the lifting heel should ideally be controlled (leg joint 5), but this joint was not taken into consideration for the simulator-based evolution. In other words, this anomaly shows the gap between the simulator and the real-world environment.

An interesting point to analyse is to see whether a relationship exists between the concept of stability and the number of evolutionary cycles. Supposing there is a clear relationship, then the nature of the correlation should be analysed. This can be further extended to check whether there exists a correlation between stability and the increase in overall fitness. With this objective in mind, a regression analysis was performed on the evolved motions. The analysis revealed a linear regression relationship between the generation number (x) and the average number of stable solutions (y). The regression models for the frontal leg-raise, warrior and hand-to-toe motions are given by equations (5), (6) and (7), respectively:

$$y = 0.34x + 0.73 \quad (5)$$

$$y = 0.35x + 0.53 \quad (6)$$

$$y = -0.005x^2 + 0.87x - 0.65 \quad (7)$$

The fitness of the regression model was tested using R^2 - the square of the sample correlation coefficient between x and y . The calculated R^2 gives a value of 0.97, which suggests that the linear regression model in equation (5) fits the relationship between x and y very well. Given the generation number (x), the regression model suggested by equation (5) can be used to predict the number of stable solutions for that generation. Therefore, if required, the model can be utilized to predict the number of stable solutions for future generations as well. A similar relationship could be seen for the warrior motion as well. The regression model for the warrior pose is given by equation (6). The model of equation (7) suggests that the number of stable solutions increases with the number of generations for the hand-to-toe motion; however, the rate

of increase gradually decreases, unlike for the first two motions. Given the generation number (x), the regression model suggested by equation (7) can be used to predict

the number of stable solutions for that generation. Figure 11 shows the graphs for the average number of stable solutions vs. generation for the three yoga motions.

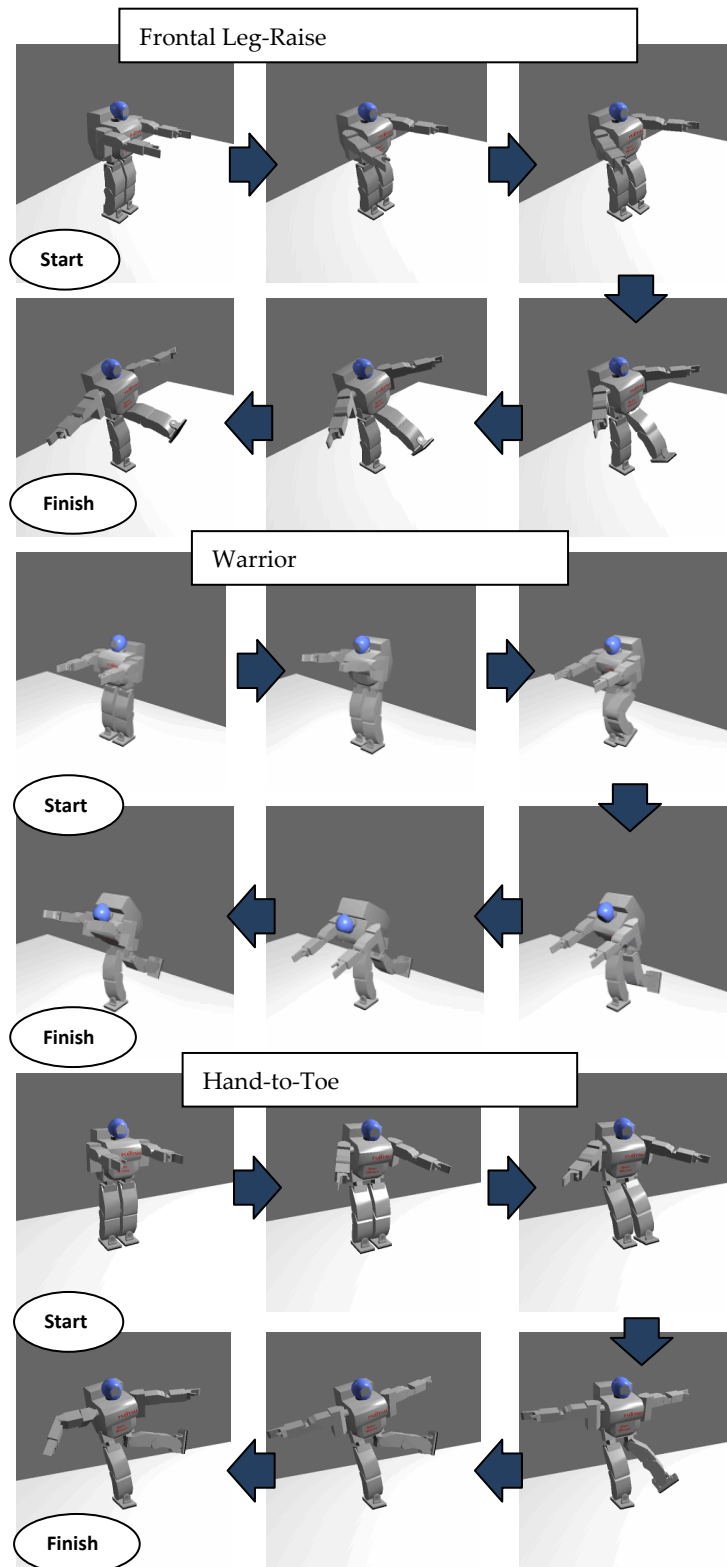


Figure 9. Behaviour of the best individuals in the simulator environment

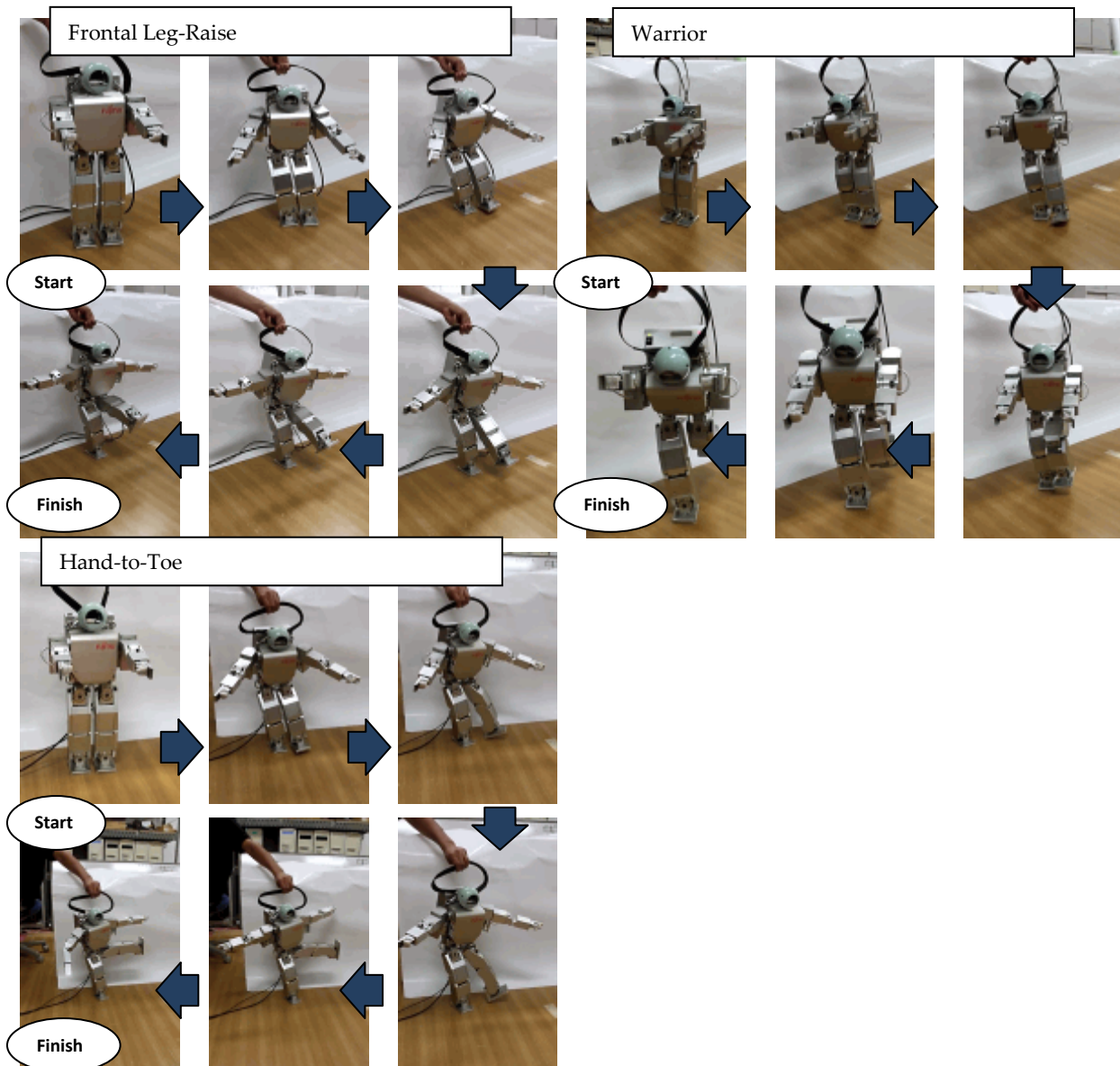


Figure 10. Applying evolved motions to the humanoid

A hill climbing search was also performed for investigating the feasibility of numerical optimization for asymmetric motion generation and to compare its performance with the GA-based approach. As such, hill climbing was applied to all three yoga motions for 20,000 iterations. The application of hill climbing to the first motion could produce an individual with a 0.42 fitness value. The improvement of fitness for hill climbing when compared to that of the GA-based optimization was limited. The maximum fitness that could be achieved with a hill climbing search for the second motion was approximately 0.57. Although this value is still less than the average fitness achieved by GA after one-hundred generations, the overall improvement was satisfactory. Although hill climbing could not acquire a stable posture for the frontal leg-raise, it could find a stable individual for the warrior pose. The fitness improvement for the hand-to-toe was not as progressive as the first two

motions. If not for the use of heuristic knowledge, the fitness improvement could have been worse. Figure 12 shows the fitness variation for the three yoga motions under the hill climbing search (best runs). The variations HC1, HC2 and HC3 refer to the frontal leg-raise, warrior and hand-to-toe motions respectively.

Figure 13 illustrates the best individual acquired through hill climbing for the warrior pose and Table 4 compares the GA-driven methodology with hill climbing for the three yoga postures. It should be noted that the best runs of several hill climbing searches were considered for this comparison, not the averaged performance of the hill climb. The Q-factor is defined as the ratio of the wasted number of generations when applying hill climbing to the total number of generations of the GA. In other words, the Q-factor is a measure of the efficiency of the GA against hill climbing.

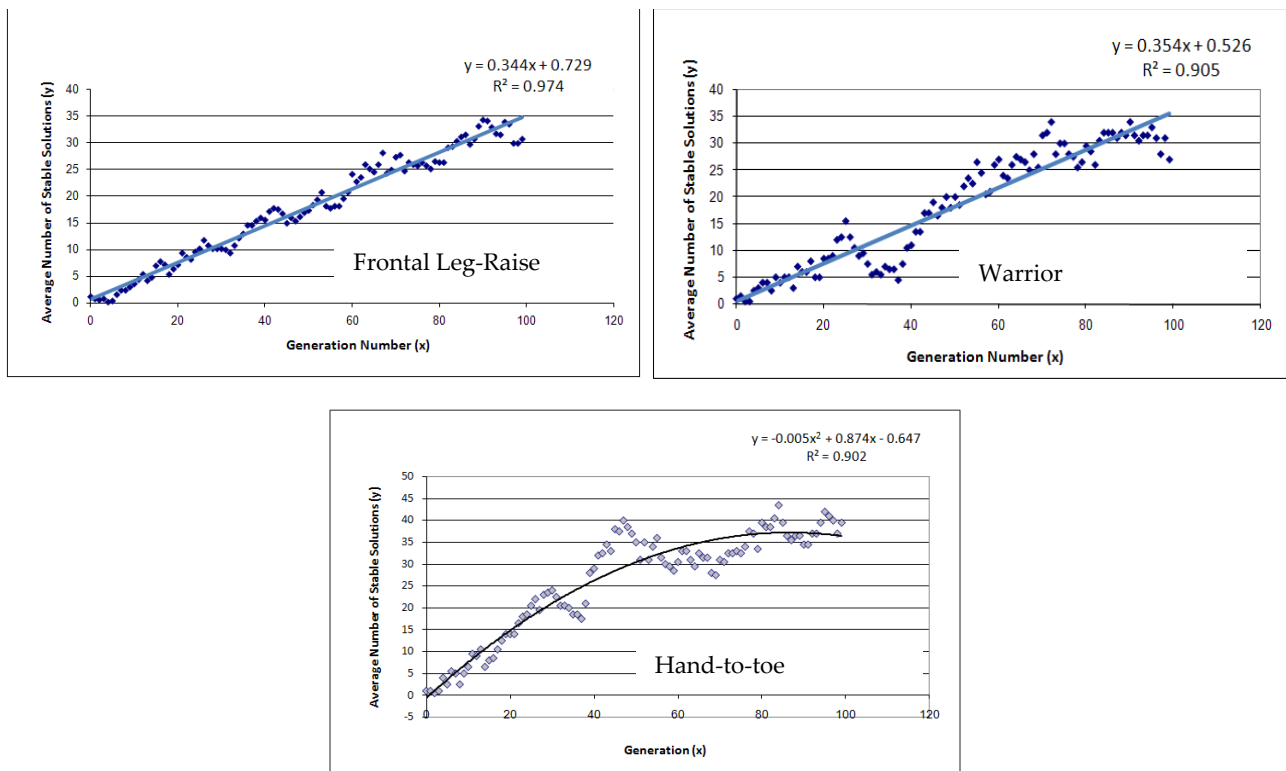


Figure 11. Regression models for the average number of stable solutions and generation

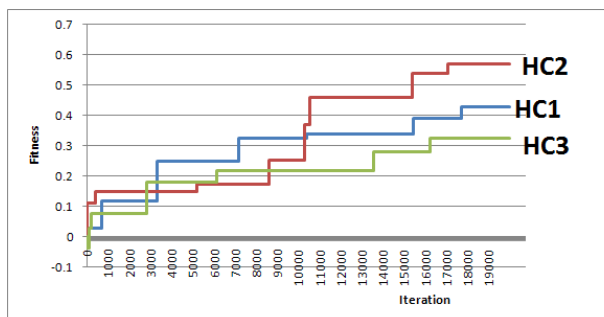


Figure 12. Hill climbing performance for the three motions

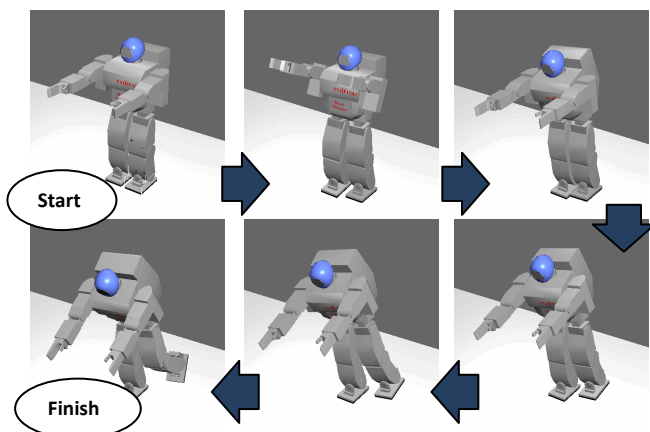


Figure 13. Best solution for the warrior pose given by the hill climb

Frontal Leg-Raise	Maximum Fitness of the Best Run	Maximum of the Average	Overtake Point	Q-Factor	Stability
GA	1.18	0.9	20 th Generation	~80%	Yes
Hill Climb	0.42	NA	NA	NA	No
Warrior	Maximum Fitness of the Best Run	Maximum of the Average	Overtake Point	Q-Factor	Stability
GA	0.92	0.65	65 th Generation	~35%	Yes
Hill Climb	0.57	NA	NA	NA	Yes
Hand-to-Toe	Maximum Fitness of the Best Run	Maximum of the Average	Overtake Point	Q-Factor	Stability
GA	0.82	0.56	45 th Generation	~55%	Yes
Hill Climb	0.32	NA	NA	NA	No

Table 4. GA vs. Hill Climb for the three trajectories

A notable point from the above GA-based evolution is the required computational time for each generation. Each generation consumed approximately 15 minutes on average (wall-clock) for processing on a PC equipped with a Core2 Quad CPU (2.5 GHz) and 4GB of memory, even after activating the fast simulation mode. Therefore,

we had to compromise on the maximum number of generations and the number of individuals permitted to exist in a given generation. It took approximately 25 hours to complete a single evolutionary run.

5. Future Work

Our study unveils a number of directions for future research. First, and foremost, the proposed method should be benchmarked by taking other competing techniques for motion generation as a baseline. For instance, neural networks (Central Pattern Generators based on Recurrent Neural Networks) have been known to deliver satisfactory results in humanoid motion planning [9, 11, 12] and it is another viable method for training the HOAP-2 to perform asymmetric motions. Then, the applicability of a variation of reinforcement learning should also be investigated, as we have identified the possibility of starting with a stable target motion and working backwards to reach a pre-defined initial state. This process may expedite learning, while helping find novel yoga motions. Another possible improvement is to treat an asymmetric trajectory as a modular configuration. A full motion trajectory can be broken down into manageable parts/modules that can be independently evolved. The aim of the evolution is to build modular controllers capable of transforming a starting posture to a target posture that is not too distant in space while maintaining balance. Due to the restricted search space of the modules, the evolution will be faster and at the same time those modules can be made reusable by other types of similar yoga motions. Moreover, Interactive Evolutionary Computation (IEC) has also shown success in many areas, including robotics [13]. IEC makes use of human judgment for evaluating individuals in contrast to conventional evaluation based on algorithmic methods. Although these human evaluations are subjective, the judgment of stability and posture cannot vary much from one evaluator to another as these can be clearly seen in graphical form. The disadvantage of using IEC in this research is, put simply, the large search space and the substantial amount of time required for evaluations. Naturally, users tend to get exhausted after several minutes of evaluation [14, 15, 16] and therefore a suitable approach is necessary to minimize the time for evolution and reduce user fatigue. A possible workaround would be a periodic user evaluation (IEC) coupled with the proposed evolutionary approach. A periodic evaluation may help direct the search and converge on a better solution than by the pure GA-based approach.

6. Conclusion

Asymmetric motion trajectories, unlike their symmetric counterparts, are complex, both in terms of search complexity and the difficulty of acquiring a stable motion, while balancing on a single limb. The motion

planning and balancing of robots, including that of humanoids, have been attempted by researchers using various techniques. As previously discussed, these methods could be divided into two main branches: conventional/traditional methods and computational intelligent techniques. These techniques have delivered results with varying degrees of success. One promising method that finds its position in the group of computational intelligent techniques is that of evolutionary robotics. In this study, we presented the findings of a set of experiments conducted in order to ascertain the feasibility of genetic algorithms in deriving asymmetric motion trajectories for humanoids. The stability and proximity of the evolved posture to the target motion was taken into consideration in determining successful motion trajectories. The results were satisfactory as they proved the ability of the approach to find highly fit solutions which were more or less similar to the desired asymmetric postures.

7. References

- [1] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, H. Inoue, "Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp", In the proc. of the IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, EPFL, pp. 2684-2689, 2002
- [2] S. Kajita, F. Kanehim, K. Kaneko, "Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point", In the proc. of ICR 4, pp. 1620-1626, 2003
- [3] D. Aydemir, H. Iba, "Evolutionary Behavior Acquisition for Humanoid Robots", In the proc. of the Parallel Problem Solving from Nature-PPSN IX Conference, pp. 651-660, 2006
- [4] T. Yanase, H. Iba, "Evolutionary Motion Design for Humanoid Robots", Frontiers in Evolutionary Robotics, pp. 567-586, 2008
- [5] W. Langdon, P. Nordin, "Evolving Hand-Eye Coordination for a Humanoid Robot with Machine Code Genetic Programming", Genetic Programming, Lecture Notes in Computer Science, Vol. 2038, pp. 313-324, 2001
- [6] O.H. Miller, N. Kaufman, "Essential Yoga: an illustrated guide to over 100 yoga poses and meditations", Chronicle Books, 2004
- [7] A. Ude, C.G. Atkeson, M. Riley, "Planning of Joint Trajectories for Humanoid Robots Using B-Spline Wavelets", In the proc. of the ICRA, pp. 2223-2228, 2000
- [8] P. Cominoli, "Development of a Physical Simulation of a Real Humanoid Robot", BIRG Logic Systems Lab, 2005
- [9] R. Zaier, F. Nagashima, "Motion Generation of Humanoid Robot based on Polynomials Generated by Recurrent Neural Network", In the proc. of the Asia Symposium on Mechatronics, 2004

- [10] HOAP-2 Instruction Manual, Fujitsu Automation Co., Ltd.
- [11] R. Zaier, F. Nagashima, "Recurrent Neural Network Language for Robot Learning", In the proc. of the 20th Annual Conference of the Robotics Society of Japan, 2002
- [12] N. Sadati, J. Taheri, "Solving robot motion planning problem using Hopfield neural network in a fuzzified environment", In the proc. of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'02), pp. 1144 - 1149, 2002
- [13] Yanase.T., Iba.H. 'Evolutionary Motion Design for Humanoid Robots', *Frontiers in Evolutionary Robotics*, pp.567-586, 2008
- [14] Raffi.K., Ying.Z., Hideyuki.T., Agogino.A.M. 'Reduced Human Fatigue Interactive Evolutionary Computation for Micromachine Design', In the proc. of the Fourth International Conference on Machine Learning and Cybernetics, pp. 5666-5671, 2005
- [15] Watanabe.A., Tanji.M., Iba.H. 'Creating Singing Vocal Expressions by means of Interactive Evolutionary Computation', In the proc. of the 5th International Workshop on Computational Intelligence & Applications, pp. 278-283, 2009
- [16] Raffi.K., Yeh.E., Zhang.Y., Agogino A.M.,Takagi.H. 'Reducing Human Fatigue in Interactive Evolutionary Computation through Fuzzy Systems and Machine Learning Systems', In the proc. of the IEEE International Conference on Fuzzy Systems, pp. 3295-3301, 2006