

Sitting Pose Generation Using Genetic Algorithm for NAO Humanoid Robots

Mo'taz Al-Hami, Rolf Lakaemper

Department of Computer & Information Sciences

Temple University, Philadelphia, PA 19122

{motaz, lakamper}@temple.edu

Abstract—Humanoid robots are increasingly used to perform human mimicking tasks, such as walking, grasping, standing and sitting on objects. To generate poses interactively using a humanoid robot, the performed poses should be controlled to satisfy any potential interaction with the surrounding environment. In this paper, a simulated humanoid robot "NAO" is used to discover a fitness-based optimal sitting pose performed on various types of sittable-objects, varying in shape and height. Using an initial set of random valid sitting poses as the input generation, genetic algorithm (GA) is applied to construct the fitness-based optimal sitting pose for the robot to fit well on the sittable-object (i.e. box and ball). The used fitness criteria reflecting pose stability (i.e. how feasible the pose is based on real world physical limitation), converts poses into numerical stability level. The feasibility of the proposed approach is measured through a simulated environment using V-Rep simulator which shows how the GA is able to generate a fitness-based optimal sitting-pose. The real "NAO" robot is used to perform results generated by the simulation.

Index Terms—Humanoid robot, pose, genetic algorithm, fitness function, crossover, mutation, generations.

I. INTRODUCTION

Humans are able to get into a variety of whole-body static poses. These poses might interact with and get support from the surrounding environment. A pose is defined as the relative arrangement of body parts and limbs. Standing and sitting are the most important whole-body pose configurations. For example, in the traditional standing pose, the plumb line (line directed exactly toward the center of gravity) passes through the shoulders tip, spine, hips center, and the center of the ankles joints. To perform the standing pose correctly, the body pose should get some external support (i.e. flat surface) to be in contact with the body's feet. When sitting, the body hips play a vital role since they get the major external support (chair, floor, or any other object valid to sit on). In the case of a sitting pose, the plumb line does not pass through the lower body parts.

Humanoid robots consist of a set of joints organized in a kinematic hierarchy, where each joint has a limited space of movement. The pose of a humanoid robot is usually defined by the servo-angles, affecting the robot's joints. A valid humanoid robot pose (set of joints angles) is shaped to satisfy specific tasks or behavior requirements [3].

In this work, the main goal is to improve the ability of the human robot to mimic the human whole-body sitting pose.

More precisely, given a few examples of valid sitting poses, and using some unknown objects to sit on, our task is to find a fitness-based optimal sitting pose that satisfies the sittable-object's parameters (shape and height).

Since the humanoid robot has a large number of joints (NAO consists of 25 degrees of freedom), the search space has a high dimensional space, making it extremely hard to explore. In situations like this, heuristic approaches can guide the search process efficiently while producing competitive results in a relatively sufficient time. Genetic Algorithms (GA) play a major role in optimization problems through the use of a heuristic guiding approach inspired by the evolution theory. So far, GA has been used as a tuning tool for humanoid robot pose control [14].

In this paper, we describe a GA evolutionary approach for finding a fitness-based optimal sitting pose for a given sittable-object. For space consideration and some simulator limitations, this work is not intended on how to place the humanoid robot in front of sittable-objects. For details regarding this direction there are many related works [6], [12]. The paper is organized as follows: Problem formulation is introduced in Section II. Related work is described in Section III. The sittable-object's height discovery process is discussed in Section IV. Section V describes the GA. Evaluation is discussed in Section VI. Finally, conclusion remarks appear in Section VII.

II. PROBLEM DEFINITION

A. NAO Robot Specifications

In this paper, we used NAO humanoid robot simulation offered by V-Rep for experimentations and evaluation. The original NAO robot was developed by Aldebaran Robotics. NAO has actuated 25 degrees of freedom (DOF, see Fig. 1). NAO has a group of sensors including cameras, microphones, sonar rangefinder, pressure sensors, tactile sensors, and IR emitters and receivers [1]. In this paper, the use of NAO capabilities are limited to collision detection, distances estimation (i.e. Euclidian distance), and physics engine. There are three different kinds of distances, which we used in the simulation throughout this work:

- The distance between the robot's head and the floor surface, determining whether the robot falls down or not using a threshold distance γ cm.

- The distance between the robot's hips and the sittable-object's surface, determining whether the robot is on the sittable-object's surface or not. Monitoring this distance is important to make sure hips are in contact with the sittable-object. In some cases, the robot might be in valid sitting pose and does not fall, but uses contact points with the sittable-objects other than hips.
- The distance between the robot's hips and the floor surface, measuring whether the robot is sitting on the sittable-object surface or not.

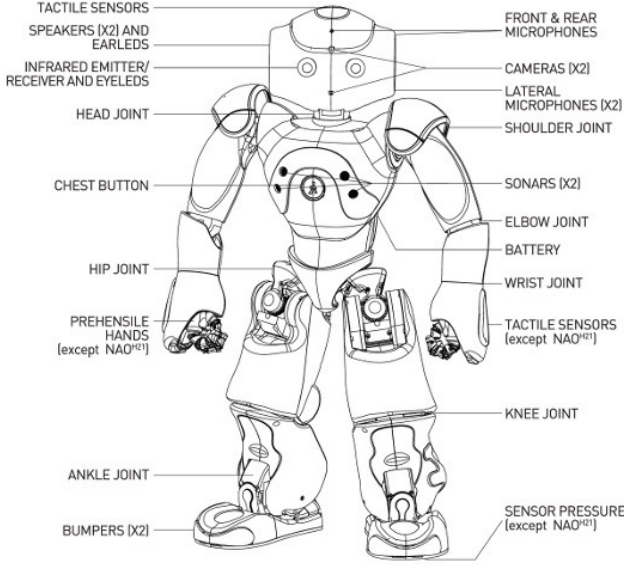


Fig. 1: NAO humanoid robot model developed by Aldebaran Robotics [1].

B. Problem Formulation

Let the robot model \mathcal{M} consists of a finite collection set of joints $\mathcal{J} = \{j_1, j_2, \dots, j_s\}$, where $|\mathcal{J}| = s$. These joints are organized and placed in a kinematic hierarchy model that satisfies a humanoid robot model. For these joints, there is a transformation $\mathcal{T} = \{t_1, t_2, \dots, t_s\} \subset \mathbb{R}^s$ related to the joints in the robot structure. Each t_i in the transformation is defined according to its parent joint (except t_1 , which is defined to some relative world position). We define a pose $p_i = (t_1, t_2, \dots, t_s) \in \mathcal{T}$ as the transformation related to the robot model rigid joints.

For such robot model, there is a space $\mathcal{S} \subset \mathcal{T}$ that is assumed, which consists of all possible sitting poses. We classify the pose as a sitting pose based on the hips joints and its angles judgment (whether the angle is larger than δ or not). We also define a set of sittable-objects $\mathcal{O} = \{o_1, o_2, \dots, o_k\}$, where $|\mathcal{O}| = k$. In order to measure a pose p stability, we define a fitness function $\mathcal{F}(\cdot)$, measuring a pose p fitness level using a fixed set \mathcal{U} of six monotonically increasing levels ($\mathcal{U} = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$). Using a set $\mathcal{I} = \{p_1, p_2, \dots, p_n\} \subset \mathcal{S}$ of initial poses which represent sitting poses, **the question that arises using the set \mathcal{I} , is how to find a stable pose p in the space \mathcal{S} , accurately fitting on the sittable-object.** Mathematically, the problem

can be formulated as an optimization problem, with the objective function $\mathcal{F} : \mathcal{S} \rightarrow \mathcal{L} \in \mathcal{U}$. The goal here is to find the global optimal statically-stable pose \hat{p} :

$$\hat{p} = \underset{p \in \mathcal{S}}{\operatorname{argmax}} \mathcal{F}(p) \quad (1)$$

If there are multiple global optimal poses, the optimal pose \hat{p} is selected randomly from those poses. In this paper, to construct a representative chromosome, we used all the related joints in the robot, which are used in the sitting pose confirmation. The joints are described in Table I.

III. RELATED WORK

As far as we know, this is the first work that uses the GA evolutionary approach to construct an optimal pose, taking into account the sittable-object's parameters (i.e. shape and height). However, there are many previous studies on how to control the humanoid robot motion [8], [17], [13], [2], [10]. The work in [8] focuses on asymmetric motion generation for some predefined yoga poses and finds stable trajectory between random initial poses set and a goal pose posture using the evolutionary genetic algorithm approach (GA). The number of active joints in GA individuals to create the goal posture varies according to the goal posture shape. In [17] an interactive evolutionary computation approach (IEC) is proposed to generate a stable simulated motion to perform tasks such as kicking a ball. The work depends on observing visually displayed motion of a robot, and then segment this motion into a set of keyframes using 3D computer graphics (3D-CG) method [16]. Based on these keyframes in which express the motion as a combination of a pose and a time frame, genetic algorithm can be applied in order to generate the optimal stable motion. Machine learning has been used in [10] for teaching humanoid robot on biped dynamic walking. This work invests the basic primitives in human behavior and use them as constraints in the learning process in order to generate human-like motion.

IV. DISCOVERING THE SITTABLE-OBJECT HEIGHT

The discovery of the sittable-object height requires applying a stable motion in order to place the robot's hips on the sittable-object. To perform this task dexterously, center of mass (COM), internal forces behavior, and the contact points with the surrounding environment should be managed appropriately [15], [4]. For the used NAO robot model, the default robot's joints COM settings parameters which were set in the V-Rep simulation are also kept in this work. The discovery process focuses on two main goals. First, since the sittable-object's height is unknown, the robot should sense any potential collision with the sittable-object to determine the sittable-object's height. Second, the robot's hips should be placed on the sittable-object such that the robot stays stable. In order to establish a stable discovery process, a predefined stable motion path is used to keep COM for the whole robot within its supporting polygon (the convex hull formed within all points of contact between the robot model \mathcal{M} and 3D surrounding environment objects). Starting from robot initial

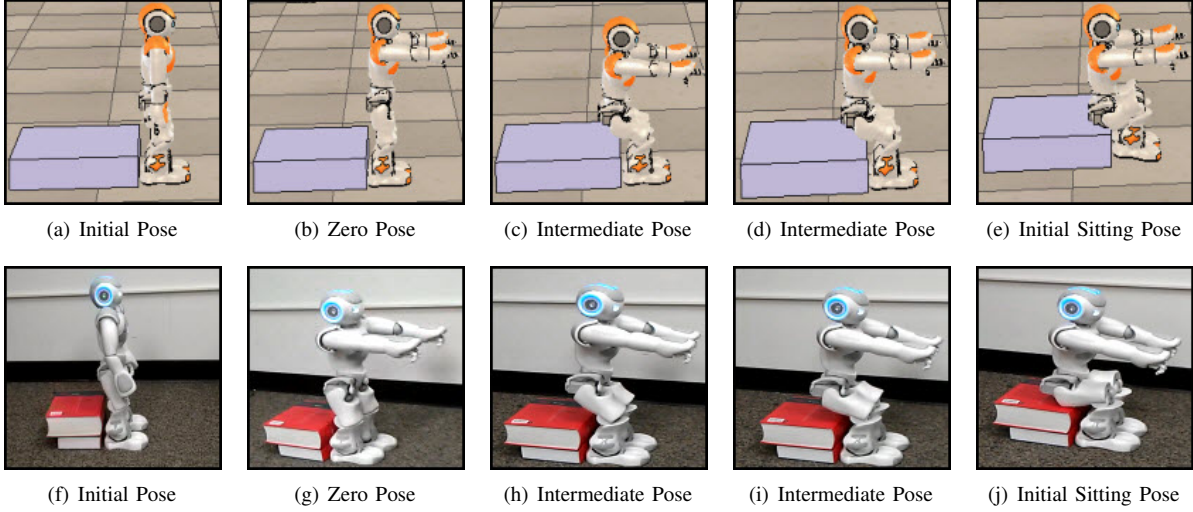


Fig. 2: Object height discovery process. The discovery process starts from adjusting the NAO robot’s hips on the cube surface and occurs in a sequence, a) a simulated initial pose for the NAO, b) a simulated zero pose (i.e. standing posture with arms pointing forward), c, d) some simulated intermediate stages, and e) a simulated initial sitting pose. f) - j) a real path discovery sequence related to the simulated sequence. Link to a video file shows the discovery process is provided.

pose (zero pose and assuming robot is placed on front of a sittable-object) shown in Fig. 2(b), the robot begins to lean in a motion which going towards a sitting pose gradually (see Fig. 2(c) - 2(e)). The discovery process adjusts specific joints in the pose p transformation: $LKneePitch$, $RKneePitch$, $LHipPitch$, $RHipPitch$ joints (see Table I).

V. GENETIC ALGORITHM

The proposed approach in Fig. 3 focuses on generating the fitness-based optimal sitting pose using an initial set ($\mathcal{I} = \{p_1, p_2, \dots, p_n\}$) of valid sitting poses. The approach starts with discovering the sittable-object’s height (see Section IV), followed by using the GA to find the fitness-based optimal sitting pose \hat{p} , which then satisfies the sittable-object’s parameters (shape and height). The used robot’s joints that participate in the GA are classified into four blocks: right arm, left arm, right leg, and left leg shown in Fig. 4, and explained in more details in Table I. The blocks and the order of these blocks are important for the crossover process to accomplish similar blocks in the parents chromosomes.

Algorithm 1 shows the GA structure. This approach starts with \mathcal{G}_1 as an initial generation (line 2), followed by fitness checking (line 5). Based on the generation \mathcal{G}_t fitness values, we use the elitist approach by moving the best $E\%$ of chromosomes, which have the highest $\mathcal{F}(p)$ values in each generation directly to next generation (line 7). Parents selection (line 12) is based on tournament strategy, where for each parent two candidates are randomly selected from \mathcal{G}_t and the one with the higher $\mathcal{F}(\cdot)$ value is assigned to that parent. Crossover (line 13) and mutation (line 16) are the main GA operators, that were used to generate an increasingly developed hypothesis regarding the optimal sitting pose \hat{p} . For each generation, we create a population (line 18) of chromosomes with a predefined size (line 10) such that,

Algorithm 1 Evolutionary GA Approach for Optimal Sitting Pose Generation

```

1:  $t \leftarrow 1$  {Generation counter}
2:  $\mathcal{G}_t \leftarrow \mathcal{I}$ 
3: repeat
4:   for all  $p \in \mathcal{G}_t$  do
5:     Compute  $\mathcal{F}(p)$  fitness
6:   end for
7:   Elitist  $\leftarrow$  From  $\mathcal{G}_t$  Select  $E\%$  of highest  $\mathcal{F}(p) \quad \forall p \in \mathcal{G}_t$  { $E$  is the elitist strategy percent}
8:    $\mathcal{C} \leftarrow 1$  {Poses counter in each generation}
9:   Population  $\leftarrow []$ 
10:  while  $\mathcal{C} \leq \text{Samples Size}$  do
11:    Parents  $\leftarrow$  tournament selection
12:    if (rand < CrossRate) then
13:      Chromosome  $\leftarrow$  Crossover(Parents) {Crossover at block level}
14:    end if
15:    if (rand < MutRate) then
16:      Chromosome  $\leftarrow$  Mutation(Chromosome)
17:    end if
18:    Population  $\leftarrow$  Population + Chromosome
19:     $\mathcal{C} \leftarrow \mathcal{C} + 1$ 
20:  end while
21:   $t \leftarrow t + 1$ 
22:   $\mathcal{G}_t \leftarrow$  Elitist + Select from Population using residual sub sampling
23: until (Generations limit or optimal pose  $\hat{p}$  is reached)

```

the population is a feasible distribution about the evolution process for \mathcal{G}_t . For this purpose, the sample size is selected to be four times the number of poses in the initial set \mathcal{I} .

In (line 22), we construct the next generation \mathcal{G}_{t+1} poses by choosing the elitist part from the current generation \mathcal{G}_t .

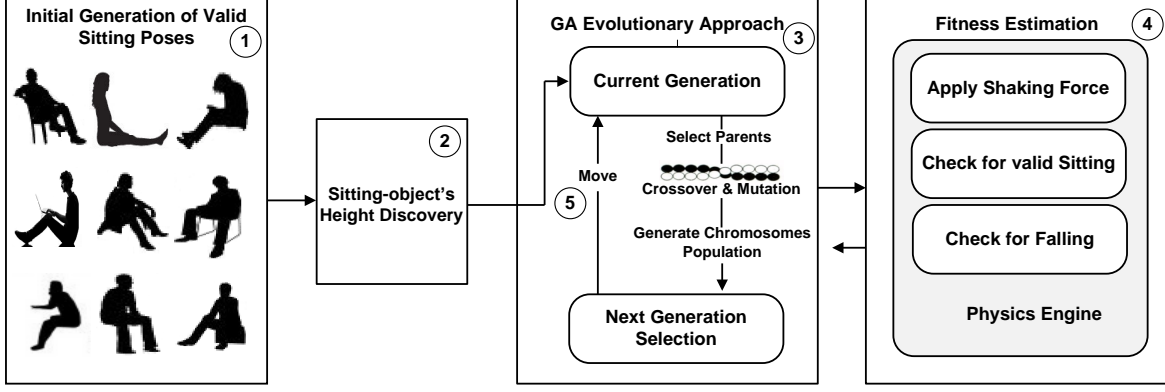


Fig. 3: Optimal sitting pose \hat{p} finding approach. (1) Initial hypothesis set \mathcal{I} . (2) A predefined path of motion to place the robot's hips on the sittable-object. (3) The generation of poses \mathcal{G}_t is used to apply GA in attempt to produce a more promising and increasingly developed hypothesis. (4) The used fitness function $\mathcal{F}(\cdot)$ structure. (5) The used GA operators and selection strategy. (6) Swapping process in order to repeat the same process for the next generation \mathcal{G}_{t+1} .

The other poses are selected from the constructed population distribution. The population distribution forms a non-deterministic process for choosing the poses that will survive in the next generation \mathcal{G}_{t+1} . The stop criteria for the GA (line 23) depends on the fitness value of the poses in the generation, or the number of generations limit. If the optimal fitness value has been reached (i.e. pose p , such that $\mathcal{F}(p) = 1$), the process stops and returns the optimal pose \hat{p} .

A. Fitness Function

To judge the stability of each pose p in a generation \mathcal{G}_t , and how well the pose fits on the sittable-object o_j , we designed a suitable fitness function. The fitness function $\mathcal{F}(\cdot)$ measures the pose p stability based on its resistance to a sided shaking force (both sides in the box case and directed towards the robot hips) initiated from the sittable-object o_j . The more the robot resists the shaking force without falling, a promotion to a better stability level for the pose is recorded. The shaking force consists of monotonically increasing six different levels. These levels are translated to six discrete values $\in \mathcal{U}$, in which were used as the fitness level.

B. Crossover and Mutation Process

A GA uses crossover and mutation operators to vary the chromosomes (the used poses in the generation \mathcal{G}_t) from generation \mathcal{G}_t to the next generation \mathcal{G}_{t+1} . The building block hypothesis [9] describes the existence of some important building blocks in the chromosome sequence. Crossover process works to produce highly fitted chromosomes in the reproduction process based on important building blocks in the chromosome sequence. In this work, a structured chromosome is used to support the building blocks hypothesis. For any pose p in a generation \mathcal{G}_t , the transformation sequence for this pose starts with the right arm joints angles (5 joints), followed by the left arm joints angles (5 joints), then the right leg joints angles (6 joints), and finally the left leg joints angles (6 joints). The crossover process occurs at block level, so

the crossover cut happens between parents left arms, parents right arms, parents left legs, and parents right legs. This means, only entire groups are exchanged in the crossover process. The mutation and crossover operators are intended to maintain a non-deterministic operation to assure diversity in the generation \mathcal{G}_t poses, and to avoid premature convergence (falling in a local minima). By this, the reproduction process keeps going toward the global optimal pose \hat{p} . A uniform distribution random number were used for mutation value β , such that $|\beta| \leq 5$ (i.e. the mutation amount is at most 5 degrees) to generate the mutation amount in the mutation process. The GA operators, CrossRate, and MutRate (see lines 12, and 15 in Algorithm 1) represent the crossover rate and mutation rate. Based on empirical tests we set fixed rates of CrossRate = 0.9, and the MutRate = 0.6 for the experiments. Table II shows an example on how some parents poses are used and block level crossover is applied, followed by joint level mutation in order to generate a new child pose chromosome. The crossover cutting points are chosen randomly with a uniform distribution from the associated building blocks in the parents chromosomes, so the new child chromosome gains a mixture between the parents building blocks.

C. New Generation \mathcal{G}_{t+1} Selection Strategy

In order to provide a non-deterministic and a distribution-based survival principle for the chromosomes in the next generation \mathcal{G}_{t+1} , we created a population of processed chromosomes (line 20 in Algorithm 1) to produce samples that represent a distribution for new generated chromosomes. The distribution is based on the fitness value $\mathcal{F}(\cdot)$ of the chromosomes. We used residual sub-sampling [11] in order to select the next generation \mathcal{G}_{t+1} chromosomes. The residual sub sampling keeps the number of poses chromosomes of each generation $|\mathcal{I}|$ constant.

Block	Used Joints	Number of Joints
Left Arm	LShoulderRoll, LShoulderPitch, LElbowYaw, LWristYaw, LElbowRoll	5
Right Arm	RShoulderRoll, RShoulderPitch, RElbowYaw, RWristYaw, RElbowRoll	5
Left Leg	LHipYawPitch, LHipPitch, LKneePitch, LAnklePitch, LAnkleRoll, LHipRoll	6
Right Leg	RHipYawPitch, RHipPitch, RKneePitch, RAnklePitch, RAnkleRoll, RHipRoll	6

TABLE I: The used joints in the GA and their classification into blocks.

	Right Arm					Left Arm					Right Leg					Left Leg						
<i>Parent₁</i>	-28	58	84	0	89	28	58	-86	0	-89	-1	-100	0	4	0	0	-1	-100	0	4	0	0
<i>Parent₂</i>	-51	18	84	101	42	44	-2	-82	-101	-38	-38	-67	82	9	-9	13	-38	-80	87	9	-13	20
Crossover	-28	58	84	0	89	44	-2	-82	-101	-38	-1	-100	0	4	0	0	-1	-100	0	4	0	0
Mutation	0	0	0	+3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Child	-28	58	84	+3	89	44	-2	-82	-101	-38	-1	-100	0	4	0	0	-1	-100	0	4	0	0

TABLE II: Generating a new child chromosome using crossover and mutation (all angles are in degree system). Using parents selected with tournament-based selection from the previous generation \mathcal{G}_{t-1} , we apply crossover block based, then we apply mutation joint based in order to generate a new child chromosome.

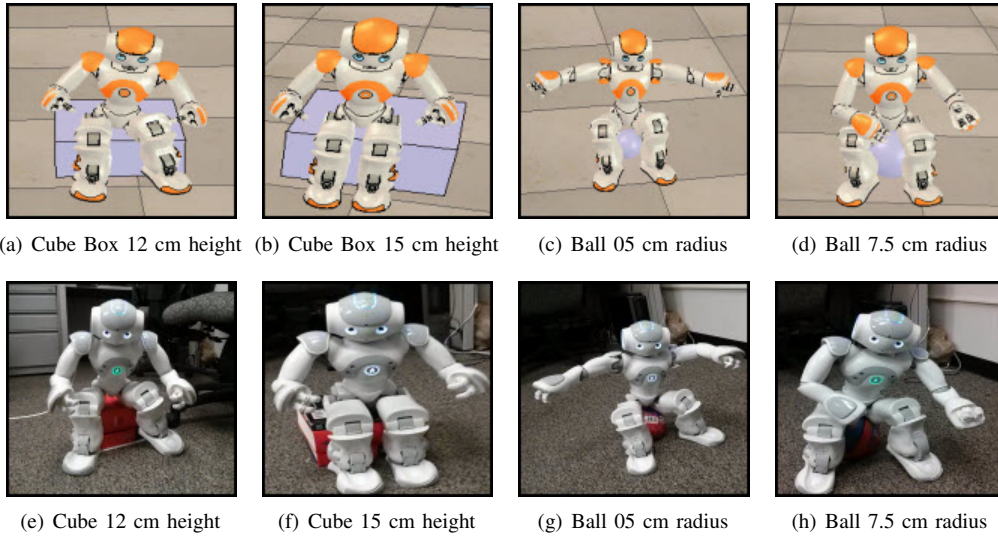


Fig. 5: Various fitness-based optimal poses generated using GA evolutionary approach for different sittable-objects varying in shape and height. Links to video files for real robot which show implementing the results on the real robot are provided by clicking on the labels (e - h).



Fig. 4: Joints blocks in the evolutionary approach. For crossover process, block level crossover is used from the parents poses in the generation \mathcal{G}_t to generate new springs in the generation \mathcal{G}_{t+1} , while single joint mutation level is used for the mutation process.

VI. EVALUATION

V-Rep is a 3D robot modeling simulator, that provides some concurrent capabilities such as actuating, sensing, and monitoring. The simulator also supports many calculation

modules such as forward and inverse kinematics, physics engines, path planning, and collision detection [5]. Therefore, we used the built-in simulated NAO robot in V-Rep, approximating the original NAO design in the Aldebaran original manufacturer. We used the V-Rep's simulation since applying the experiment on a real robot is difficult due to many obstacles, the major one being the difficulty to apply the fitness function in the real world.

The communication with V-Rep's simulator is programmed using the Python remote application programming interface (remote API) via socket communication. The client's side consists of genetic algorithm unit and fitness function unit, both implemented using Python programming language. At the opposite side, the server's side (V-Rep's simulation side) consists of a customized simulated environment involving the NAO robot and the sittable-object. The internal structure of the simulation is supported with physics engines (bullet, and ODE) replicating the real world physics in the simulation environment. The supported physics engines are able to detect any potential collision, approximate the gravity force, and assign the robot joints masses such that

the relation between COM of the NAO robot and supporting polygon can be mathematically estimated. In our evaluation, we chose bullet physics engine to measure pose stability and its ability to be performed on the sittable-object.

To assure pose validity (joint positions being within their limits), the output from crossover and mutation operator has to be checked, since the mutation process mutate the pose randomly, not guaranteeing the generated pose joints to be within their predefined limits. The used NAO model in V-Rep can discover these kinds of fault poses, and exclude them from the experiment. In the evaluation, we focus on two different sittable-objects types (box and ball). For each sittable-object, the experiment is executed on two different heights, and the best sitting pose chromosome is generated for each experiment using the GA. The objects are selected based on their availability in the V-Rep's simulator. For experimental purposes, we used randomly selected 12 sitting poses (not relating to a specific sittable-object) as initial generation \mathcal{G}_1 , and based on these initial poses we performed the GA algorithm to produce an increasingly developed hypothesis.

Fig. 5 shows the final optimal poses for each sittable-object case. For the box sittable-objects in Fig. 5(a) and Fig. 5(b), we used two different heights (12cm and 15cm) in the evaluation experiments. Since the generated pose from the experiment should assure the pose stability on the sittable-object, the generated pose might have a different number of contacts with the surrounding environment in order to keep the pose stable. In the case of ball sittable-object, the situation is slightly different. The sittable-object (ball) tends to roll easily, unlike the box sittable-object. The used shake force, which is applied in the fitness function $\mathcal{F}(\cdot)$ plays a major role in drawing the shape of the required adjustment to achieve the optimal pose. The crossover and mutation operators have directed the parents poses towards a more fitted pose that can achieve a higher fitness level value.

The computation time depends on the generation size and generations limit, however to speed up the GA, we applied the elitist selection strategy such that the best $E\%$ poses in every generation were kept in the next generation (i.e. in our case, we kept the best 20% of the poses in each generation). At the same time, the parents selection in the generation chromosomes depends on a tournament selection. For each parent, two randomly chromosomes are selected from the generation. Then, based on the fitness value for these chromosomes we choose the one with the higher fitness level. Throughout the experiment, we noticed slight differences between the simulation results, and how these results appear using the actual real robot. Since we are working through an open loop, these slightly differences are not recovered in the current work, thus might be considered as future work.

VII. CONCLUSION

In this paper, we presented a Genetic Algorithm (GA) evolutionary approach for a fitness-based optimal sitting pose generation, that is able to be performed on a sittable-object in a stable manner. We described how to construct a

fitness function that assigns a stability level score to a sitting pose. To discover object height, we applied a predefined motion, which keeps the COM within its supporting polygon, while simultaneously sensing any potential collision with the sittable-object. We also discussed how to apply the block-based crossover operator complying with the building blocks hypothesis [9], as well as how to use the process of mutation in a pose chromosome sequence.

REFERENCES

- [1] Nao robot key features. <http://www.aldebaran-robotics.com/en/Discover-NAO/Key-Features/hardware-platform.html>. [Online; accessed on July-28-2013].
- [2] A. D. Ames, E. A. Cousineau, and M. J. Powell. Dynamically stable bipedal robotic walking with nao via human-inspired hybrid zero dynamics. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pages 135–144. ACM, 2012.
- [3] C. G. Atkeson, J. G. Hale, F. Pollick, M. Riley, S. Kotosaka, S. Schaal, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar, et al. Using humanoid robots to study human behavior. *Intelligent Systems and their Applications, IEEE*, 15(4):46–56, 2000.
- [4] S. Cotton, A. P. Murray, and P. Fraitse. Estimation of the center of mass: from humanoid robots to human beings. *Mechatronics, IEEE/ASME Transactions on*, 14(6):707–712, 2009.
- [5] M. Freese, S. Singh, F. Ozaki, and N. Matsuhira. Virtual robot experimentation platform v-rep: a versatile 3d robot simulator. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 51–62. Springer, 2010.
- [6] J. Gutmann, M. Fukuchi, and M. Fujita. Real-time path planning for humanoid robot navigation. In *international joint conference on artificial intelligence*, volume 19, page 1232. LAWRENCE ERLBAUM ASSOCIATES LTD, 2005.
- [7] T. Hester, M. Quinlan, and P. Stone. Generalized model learning for reinforcement learning on a humanoid robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2369–2374. IEEE, 2010.
- [8] D. S. Hettiarachchi and H. Iba. An evolutionary computational approach to humanoid motion planning. *Int J Adv Robotic Sy*, 9(167), 2012.
- [9] J. H. Holland. Adaption in natural and artificial systems. 1975.
- [10] M. Lapeyre, O. Ly, and P. Oudeyer. Maturational constraints for motor learning in high-dimensions: the case of biped walking. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 707–714. IEEE, 2011.
- [11] J. S. Liu, R. Chen, and T. Logvinenko. A theoretical framework for sequential importance sampling with resampling. In *Sequential Monte Carlo methods in practice*, pages 225–246. Springer, 2001.
- [12] K. Okada, M. Inaba, and H. Inoue. Walking navigation system of humanoid robot using stereo vision based floor recognition and path planning with multi-layered body image. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2155–2160. IEEE, 2003.
- [13] S. Ra, G. Park, C. Kim, and B. You. Pca-based genetic operator for evolving movements of humanoid robot. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 1219–1225. IEEE, 2008.
- [14] J. Roberts, D. Kee, and G. Wyeth. Improved joint control using a genetic algorithm for a humanoid robot. In *Proceedings of the 2003 Australasian Conference on Robotics and Automation*. Australian Robotics and Automation Association Inc, 2003.
- [15] L. Sentis, J. Park, and O. Khatib. Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *Robotics, IEEE Transactions on*, 26(3):483–501, 2010.
- [16] K. Yamane and Y. Nakamura. Synergetic cg choreography through constraining and deconstraining at will. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 855–862. IEEE, 2002.
- [17] T. Yanase and H. Iba. Evolutionary motion design for humanoid robots. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1825–1832. ACM, 2006.