International Journal of Engineering & Technology, 7 (4.27) (2018) 153-157

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Assistive Robot Simulator for Multi-Objective Evolutionary Algorithm Application

Z. Mohamed[1]*, M.A Ayub[2], M.H.M Ramli[3], M.S.B Shaari[4], S. Khusairi[5]

[1]Faculty of Mechanical Engineering Universiti Teknologi MARA, Shah Alam Selangor, Malaysia
[2] Sports Engineering & Artificial Intelligence Centre, Faculty of Mechanical Engineering Universiti Teknologi MARA, Shah Alam Selangor, Malaysia
**Corresponding author E-mail: zulkifli127@salam.uitm.edu.my*

Abstract

This paper presents a new assistive robot simulator for multi-objective optimization application. The main function of the simulator is to simulate the trajectory of the robot arm when it moves from initial to a goal position in optimized manner. A multi-objective evolutionary algorithm (MOEA) is utilized to generate the robot arm motion optimizing three different objective function; optimum time, distance, and high stability. The generated neuron will be selected from the Pareto optimal based on the required objectives function. The robot will intelligently choose the best neuron for a specific task. For example, to move a glass of water required higher stability compare to move an empty mineral water bottle. The simulator will be connected to the real robot to test the performance in real environment. The kinematics, mechatronics and the real robot specification are utilized in the simulator. The performance of the simulator is presented in this paper.

Keywords: *Robot arm simulator, Optimization, Multi-objective evolutionary algorithm, Genetic algorithm, Neural network.*

1. Introduction

A robot simulator is very important for a researcher to simulate path and trajectory of a real robot. It will show the expected outcome before a robot can be developed physically. The will reduce the error and cost when building a full-scale robot. It will also speed up the development process. Normally, robot simulator development will be divided into two main section; the upper body and the lower limb.

A dynamics simulator for Compliant Humanoid Robot (COMAN) had been proposed by [1]. In this research the main advantage is an efficient symbolic dynamics equation is generated with high degrees of freedom. The developed simulator has a user define actuator dynamics model, ground model and fall detection. An imitation type of simulator had been proposed by [2]. A NAO humanoid robot is used in this research. The simulator equipped with Microsoft Kinect will capture the motion of the instructor and mapped to NAO robot. The communication between the human and humanoid robot used Robot Operating System (ROS) framework.

In other work by [3], a joint trajectory optimized controller for a humanoid robot simulator had been proposed. The proposed simulator is based on Open Dynamics Engine and GLScene graphics library which visualized the feedback. In this research, a realistic dynamics approach allows the testing without accessing the real hardware. iCub simulator had been design by [4], [5] to reduce the learning process for exploration and computation. The simulator consider the confidence function in order for the robot autonomously adapted the environment.

Ogura had proposed an integration of robot motion environment and the developed dynamics simulator. The simulation specification can be included to the robot motion. This research proposed a new method, where a new module can be added easily to the simulator. In the experiment the simulation embedded brain changes the motion planning of block moving problem is illustrated.

A virtual model simulator had been proposed by [7] adapting Danevit-Hartenberg formulation. The robot dynamics can be visualized, and the control algorithm can be embedded to the simulator. The proposed simulator is integrated to a robot system and the produced good performance. Lening proposed a software framework called BiRRTOpt which is divided into two main section; Guess and Optimization. The simulator shows good performance and have the ability to compute a collision free and optimized trajectory

A virtual simulator for a Mitsubishi Movemaster RV-M1 Robot had been proposed by [9]. The developed simulator system focuses on education in the university. The main objective is to increase the student's interactivity with the robot by using the Unity and the Oculus Rift headset (for gaming) to increase the visualization process. The system will allow the student to do off line programming of the real robot

A different type of simulator based on tendon driven system had been introduced by [10]. The developed simulator has the ability to realize the human motion naturally. The performance of the simulator is good but facing difficulties on the real hardware. An industrial type of robot simulator developed by [11] to simulate the production line environment and to test the feasibility of work process. The proposed simulator try to reduce the human error when working in a poor environment.

Other type of robot simulator for robot research also had been proposed by [12] with internet control robot simulator, [13] with 3D collision avoidance robot simulator, [14] with wireless robot simulator, [15] with trajectory optimization simulator and humanoid robot soccer simulator by [16].

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, the generated neural controller for the robot arm motion optimizing three different objective functions: shortest time, shortest distance and optimum stability had been proposed. These criteria cover a wide range of robot motion required during the execution of task. The generated neural controllers are implemented in the simulated platform to test and compare the performance. The advantage of the proposed method is that a single neural controller is employed to generate the robot arm motion in a wide range of initial and goal locations. The robot simulator selects the neural controller based on the task required to perform.

2. Kinematics Analysis

In performing a task such as moving an object, pick and place the robot arm motion and energy utilization must be carefully selected to complete the task successfully. In the earlier stage of this research, the simulator is developed in MATLAB environment and the 3-dimensional drawing had been design using Solidwork. The visualisation is much easier, and it allow the motion to be simulated before the fabrication process. This is also to make sure the mechanism, mechanical and electronic parts are all in its best position. In the next stage of the research, the 3D solid modelling will be embedded to the simulator for better visualization. MATLAB is used to interact with the real robot due to its flexibility and ability to interface with controller board via USB connector.

2.1. Forward Kinematics and D-H Parameters

The forward kinematics of the simulator can be determined via Denavit-Hartenberg (D-H) analysis [17], [18]. By using this method, homogeneous transformation matrix is determined, which specifies the position and orientation of the robot with respect to the base as in Figure 1(a) and (b) [17, 18]. D-H parameters for the robot left hand for assigned frames are shown in Table 1.

Table 1. D-H Parameter for the Robot Simulator

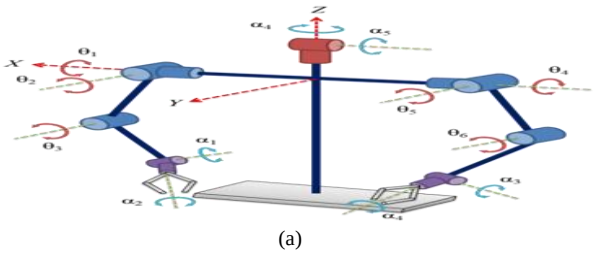
Joint, i	α_i	a_i	d_i	θ_i
OA	90^0	0	d_1	θ_1
B	0	a_1	0	θ_2
C	0	a_2	0	θ_3
D	0	0	0	Gripper

By substituting these parameters, the transformation matrices T1 to T6 can be shown in equation (1). The end effector or gripper position can be easily determine with given join angle and it can be generated from the transformation matrix below [19].

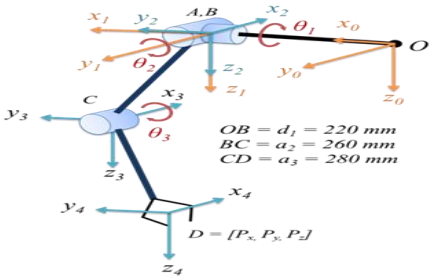
$$T_4^0 = A_1A_2A_3A_4 = \begin{bmatrix} n_x & o_x & a_x & d_x \\ n_y & o_y & a_y & d_y \\ n_z & o_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where;

$n_x = C_1C_{23}$ $\quad n_y = S_1C_{23}$ $\quad n_z = S_{23}$
$o_x = -C_1S_{23}$ $\quad o_y = -S_1S_{23}$ $\quad o_z = C_{23}$
$a_x = S_1$ $\quad a_y = -C_1$ $\quad a_z = 0$
$d_x = C_1(a_3C_{23} + a_2C_2)$
$d_y = S_1(a_3C_{23} + a_2C_2)$
$d_z = a_3S_{23} + a_2S_2 + d_1$



(a)



(b)

Fig. 1: (a) Kinematics Model of the Robot (b) Coordinate Frame of the Robot

The orientation of the end effector is shown in the first three columns whereas the last column represents the position of end effector.

2.2. Inverse Kinematics Analysis

Inverse kinematics analysis is very important in robot research. It is the vice versa of forward kinematics where the joint angle can be determined if the end effector position is known. The solution is more complex than direct solution since there is no unique analytic solution. Geometric approach (Figure 2) is utilized in the simulator and the inverse kinematics of the mobile robot is determined as in equations below [20], [21].

$$\theta_1 = \tan^{-1}\left(\frac{P_y}{P_z}\right) \quad (2)$$

$$\theta_3 = \cos^{-1}\left(\frac{\left(P_d^{\,2} + (P_x - d_1)^2 - \left(a_2^{\,2} + a_3^{\,2}\right)\right)}{2a_2a_3}\right) \quad (3)$$

$$\theta_2 = \tan^{-1}\left(\frac{(P_x - d_1)}{P_d}\right) \pm \cos^{-1}\left(\frac{a_2 + a_3\cos\theta_3}{s}\right) \quad (4)$$

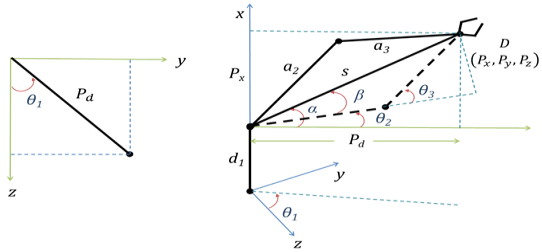


Fig. 2: Inverse kinematics analysis of the robot simulator

3. Simulator Design

Figure 3 shows the 3D modelling of the robot arm developed using Solidwork software. The developed model will be embedded to the simulator in the next stage of the simulator development.

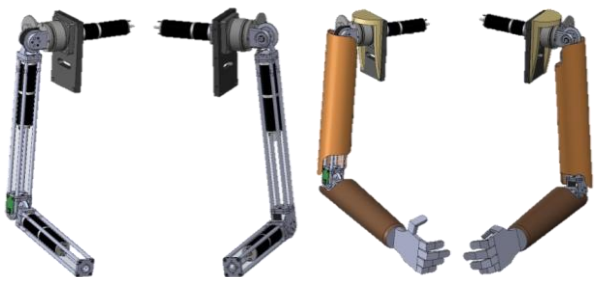


Fig. 3: 3D Modelling of the robot arm

The complete simulator is shown in Figure 4 and it is developed in MATLAB environment. The simulator will be connected to the real robot via Graphical User Interface (GUI) as in Figure 5. The real robot and simulator can be controlled automatically and manually using the developed GUI. In the first stage, the generated neural controller using multi-objective optimization will be implemented in this simulator to verify the performance before it can be tested on the real robot. Robustness, capability to solve complicated calculation, data acquisition and manipulation are the characteristics MATLAB is utilized to develop the simulator. Other advantage of MATLAB is the capability to interact with laser range finder, camera and robot controller. The key specifications of the robot simulator are shown below:

Arm length – 54 cm
Total height – 134 cm
Robot width – 52 cm
Upper body weight – 14 kg

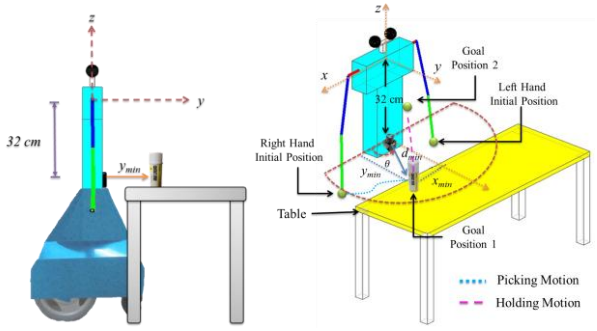


Fig. 4: Robot Arm Simulator

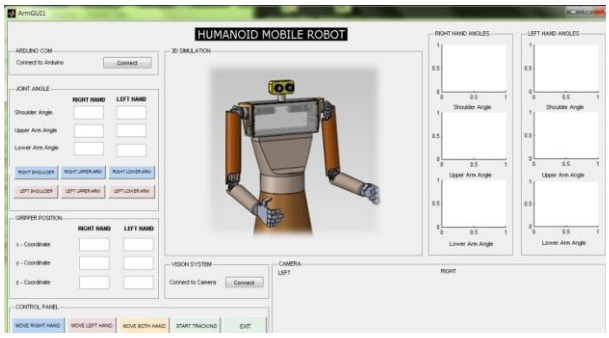


Fig. 5: Robot arm simulator GUI

4. Robot Arm Motion via MOEA

4.1. Artificial Neural Networks (ANN)

A feed forward neural network (FFNN) with single hidden layer is utilized to generate the optimum path of the robot arm as in Figure 6 [22]. FFNN is chosen due to its simplicity and robustness compared to back propagation NN [23]
A set of generated FFNN for the robot arm to move from starting to a goal position receives three inputs which is the differences between the robot hand start and goal positions in Cartesian coordinate system. Three output units for each FFNN represent shoulder (θ_1), upper arm (θ_2) and lower arm angles (θ_3). The weight connections between the neural controllers are optimized using genetic algorithm.

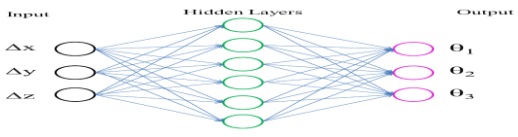


Fig. 6: Feed forward neural network

4.2. Genetic Algorithm (GA)

In this simulator an extended multi-population genetic algorithm is utilized, where the subpopulations apply different evolutionary strategies [24]–[26]. The summary of GA parameters is shown in Table 2.

Table 2: Summary of genetic algorithm parameter.

Number of Subpopulations	**3**
Number of Individuals	500, 500, 400
Maximum Generations	80

4.3. Fitness Function

Three different objective functions had been considered in this work namely optimum time (OT), optimum distance (OD) and optimum energy (OE). These objective functions are considered based on a wide range of robot motion characteristics to execute every day task.
The first fitness function is the shortest time for the robot hand simulator to move from its starting to the goal position. Thus the fitness function is to optimize the number of step for the robot to reach the final position.

$$f_1 = n_{step} \quad (5)$$

The second fitness function is shortest distance is as follows:

$$f_2 = abs\left(\sum rt_i - sd\right) \quad (6)$$

where,
$\sum rt_i$ - summation of robot hand distance
sd - shortest distance.

If the robot hand simulator is required to move with high stability, a constant acceleration fitness function had been introduced. A gradually increasing velocity at the beginning of the motion and gradually decreasing motion toward the goal position had been considered. Therefore, the stability fitness function is as follows:

$$f_3 = \sum a_{hand} + (v_{hand_end} * w) + (n_{vc} * w) \quad (7)$$

where,
$\sum a_{hand}$ - summation of robot hand acceleration
v_{hand_end} - simulator velocity toward the goal position
w - weight function
n_{vc} - number of velocity changes

5. Results

In order to test the performance of the robot simulator, the generated neural controller are tested with two different set of initial and goal position as tabulated in Table 3. In this experiment, the behaviour of generated neural controller will be compared and discussed.
The Pareto front of the 80th generation simultaneously optimizing all three objective function for the left hand simulator is shown in Figure 7. The Pareto front has 12 neural controllers and the best neural controller had been chosen from the generated Pareto front, *NC1*. The chosen neural controller had been implemented in the simulated environment as in Figure 8.
Utilizing the same neural controller (NC1), the robot simulator had been tested its performance by selecting two different goal position (P2) as in Table 3. The neural controller shows good performance optimizing all three-objective function, shortest time, shortest distance and high stability. It can be observed from the motion the speed of the robot arm is constant and produce the optimum distance and time (Figure 9). The same generated neural controller is further tested in a different goal position in *x-axis*

direction. The goal position is shifted 10 cm from its original position (Figure 10) and the robot arm manage to execute the task successfully.

Table 3 Simulation parameters

Experimental Setup		***Start Position***			***End Position***		
		x_{init}	y_{init}	z_{init}	x_g	y_g	z_g
Neural Controller 1 (NC1)	Position 1 (P1)	22	2	-50	15	38	-36
	Position 2 (P2)	22	2	-50	5	38	-36

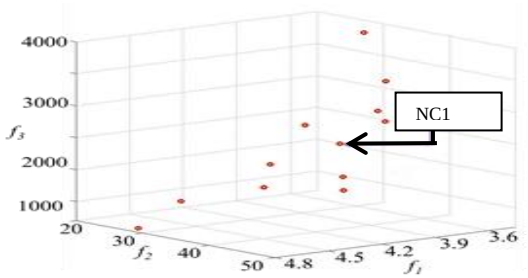


Fig. 7: Selected neural controller

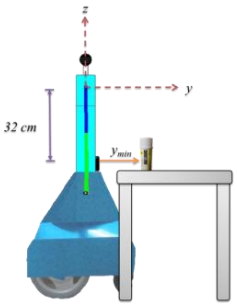


Fig. 8: Fully Developed Robot Simulator

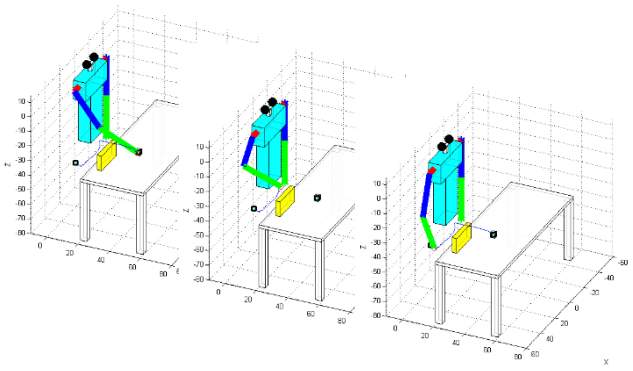


Fig. 9: Robot arm motion for position 1 (P1), initial *(20, 2,-50)* and goal *(15, 38,-36)*

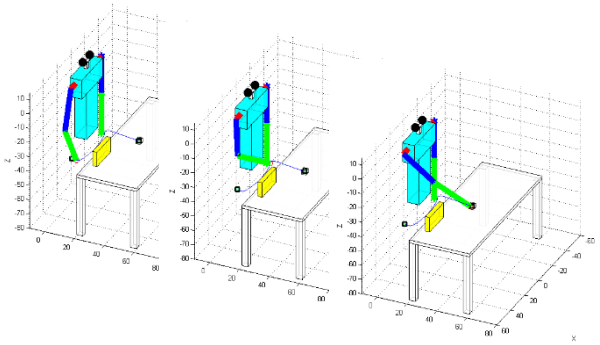
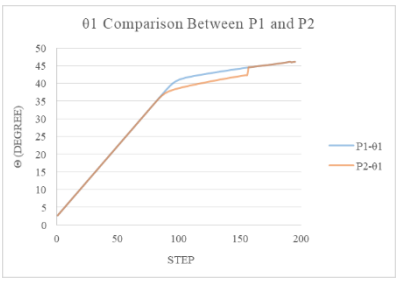


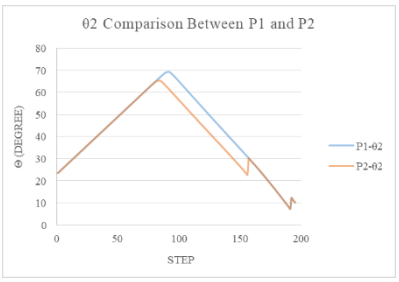
Fig. 10: Robot arm motion for position 2 (P2), initial *(20, 2,-50)* and goal *(5, 38,-36)*

The shoulder, upper arm and lower arm angle comparison is shown in Figure 11. At the initial stage of the motion, the robot simulator shows the same angle for each joint of P1 and P2. The slight change only occurs half of the trajectory due to different goal position. A similar trajectory shown in Figure 12, when the end effector move from its initial to the goal position in x, y and z-axis.

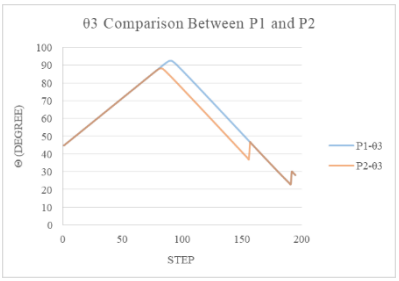
The velocity and acceleration comparison (Figure 13) show more significant analysis. A similar velocity profile is shown for both P1 and P2 but a different performance for acceleration comparison. The acceleration is increasing when the robot try to move from the bottom of the table and avoid the acceleration. Toward the end of the trajectory, the acceleration is not gradually decreased thus the fitness function need to be further improved.



(a)

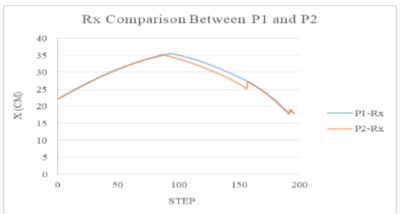


(b)

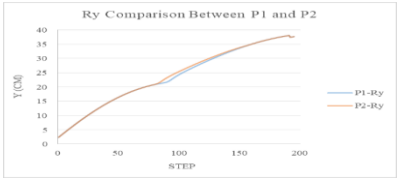


(c)

Fig. 11: Comparison between P1 and P2 for (a) Shoulder angle (θ_1) (b) Upper arm angle (θ_2) (c) Lower Arm angle (θ_3)



(a)

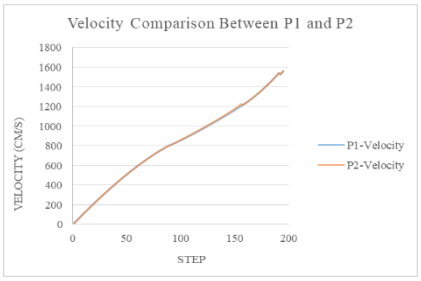


(b)

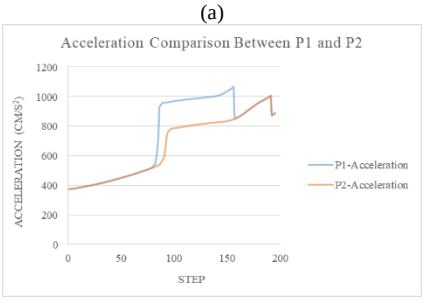


(c)

Fig. 12: Comparison between P1 and P2 for (a) x-axis (R_x) (b) y-axis (R_y) (c) z-axis (R_z)



(a)



(b)

Fig. 13: (a) Velocity and (b) Acceleration profile comparison of the robot simulator end effector

6. Conclusion

In this paper a MATLAB GUI Robot arm simulator had been developed and tested. The performance of the simulator is tested by implementing the generated neural controller to the system and the results show optimized trajectory of the robot arm. Two different initial and goal positions of the robot arm had been tested on the generated neural controller and the arm motion show good performance. The robot arm is successfully reach the goal position utilizing only one neural controller and simultaneously optimizing distance, speed and energy.

Acknowledgement

Research reported in this International Journal of Engineering Technology was supported by Ministry of Higher Education (MOHE) and Universiti Teknologi MARA of the Fundamental Research Grants under award number 600-RMI/FRGS 5/3 (0114/2016).

References

[1] H. Dallali *et al.*, (2013) "Development of a dynamic simulator for a compliant humanoid robot based on a symbolic multibody approach," *2013 IEEE Int. Conf. Mechatronics, ICM 2013*, pp. 598–603.

[2] P. Shahverdi and M. T. Masouleh, (2017) "A simple and fast geometric kinematic solution for imitation of human arms by a NAO humanoid robot," *4th RSI Int. Conf. Robot. Mechatronics, ICRoM 2016*, pp. 572–577.

[3] J. L. Lima, J. C. Gonçalves, P. G. Costa, and A. P. Moreira, (2008) "Humanoid robot simulation with a joint trajectory optimized controller," *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, pp. 986–993.

[4] R. Saegusa, G. Metta, and G. Sandini, (2009) "Active Learning for Multiple Sensorimotor Coordination Based on State Con fi dence," pp. 2598–2603.

[5] B. Akgün, N. Dağ, T. Bilal, I. Atil, and E. Şahin, (2009) "Unsupervised learning of affordance relations on a humanoid robot," *2009 24th Int. Symp. Comput. Inf. Sci. Isc. 2009*, pp. 254–259.

[6] T. Ogura, K. Okada, and M. Inaba, (2007) "Realization of dynamics simulator embedded robot brain for humanoid robots," *Proc. - IEEE Int. Conf. Robot. Autom.*, no. April, pp. 2175–2180.

[7] X. Deng, J. Wang, and W. Zhu, "Study on Simulation for Humanoid Robot," *2008 Int. Conf. Intell. Comput. Technol. Autom.*, pp. 732–735, 2008.

[8] L. Li, X. Long, and M. A. Gennert, (2016) "BiRRTOpt: A combined sampling and optimizing motion planner for humanoid robots," *IEEE-RAS Int. Conf. Humanoid Robot.*, pp. 469–476.

[9] R. Crespo, R. García, and S. Quiroz, (2016) "Virtual reality simulator for robotics learning," *Proc. 2015 Int. Conf. Interact. Collab. Blended Learn. ICBL 2015*, no. December, pp. 61–65.

[10] Y. Nakanishi, K. Hongo, J. Urata, I. Mizuuchi, and M. Inaba, (2010) "Muscle geometric topology estimation based on muscle length - Joint angle nonlinearity in tendon-driven robot systems," *2010 10th IEEE-RAS Int. Conf. Humanoid Robot. Humanoids 2010*, pp. 263–268.

[11] T. Choi, D. Park, K. Chung, D. H. Kim, K. T. Park, and J. Kyung, (2012) "Industrial dual-arm robot simulator implementation," *2012 9th Int. Conf. Ubiquitous Robot. Ambient Intell. URAI 2012*, no. Urai, pp. 553–554.

[12] W. M. H. W. Kadir, R. E. Samin, and B. S. K. Ibrahim, (2012) "Internet controlled robotic arm," *Procedia Eng.*, vol. 41, no. Iris, pp. 1065–1071.

[13] S. Charoenseang, A. Srikaew, D. M. Wilkes, and K. Kawamura, (1998) "Integrating visual feedback and force feedback in 3-D collision avoidance for a dual-arm humanoid robot," *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 4, pp. 3406–3411.

[14] M. A. K. Yusoff, R. E. Samin, and B. S. K. Ibrahim, (2012) "Wireless mobile robotic arm," *Procedia Eng.*, vol. 41, no. Iris, pp. 1072–1078.

[15] I. W. Park, Y. D. Hong, B. J. Lee, and J. H. Kim, (2012) "Generating optimal trajectory of humanoid arm that minimizes torque variation using differential dynamic programming," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 1316–1321.

[16] G. Tong, Z. Shao, H. He, and X. Chen, (2008) "A HiFi 3D simulator for humanoid robot soccer," *Proc. 2008 IEEE Int. Conf. Inf. Autom. ICIA 2008*, pp. 804–808.

[17] L. Sciavicco and B. Siciliano, (2001) *Modelling and Control of Robot Manipulators*. London: Springer.

[18] R. N. Jazar, (2007) *Theory of Applied Robotics : Kinematics, Dynamics and Control*. New York: Springer.

[19] Z. Mohamed and G. Capi, (2012) "Development of a New Mobile Humanoid Robot for Assisting Elderly People," *Procedia Eng.*, vol. 41, no. Iris, pp. 345–351.

[20] D. Xu, C. a. Acosta Calderon, J. Q. Gan, H. Hu, and M. Tan, (2005) "An analysis of the inverse kinematics for a 5-DOF manipulator," *Int. J. Autom. Comput.*, vol. 2, no. 2, pp. 114–124.

[21] M. Spong, F. Lewis, and C. Abdallah, (1993) "*Robot Control : Dynamics, Motion Planning, and Analysys*" New York: IEEE Press.

[22] Z. Mohamed, M. Kitani, and G. Capi, (2013) "Humanoid Robot Arm Performance Optimization using Multi Objective Evolutionary Algorithm," *Int. J. Control Autom. Syst.*, vol. 10, no. 2, pp. 1–8.

[23] D. J. Montana and L. Davis, (1989) "Training Feedforward Neural Networks Using Genetic Algorithms," in *Eleventh International Joint Conferences on Artificial Intelligence*, pp. 762–767.

[24] G. Capi and K. Doya, (2006) "Application of evolutionary computation for efficient reinforcement learning," *Appl. Artif. Intell.*, vol. 20, no. 1, pp. 1–20.

[25] T. Belding, (1995) "The Distributed Genetic Algorithm Revisited," in *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 114–121.

[26] E. Cantu-paz, (2000) "Topologies , Migration Rates , and Multi-Population Parallel Genetic Algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 91–98.