

Motions Planning for Virtual Character

Kalouache Saida, Cherif Foudil

Biskra University, LESIA Laboratory,
Algeria

kalouache.saida19@gmail.com, foud_cherif@yahoo.fr

Abstract. The existence of a virtual character in an environment requires its interaction within it. This interaction involves a combination of two skills, which are locomotion and manipulation. Our method treats this type of full-body motion planning problems. Where locomotion and manipulation are planned separately and coordinated with each other. We use a locomotion planner based on motion graph that searches for convenient body position for performing the manipulation motions successfully, and combined with an iterative inverse kinematic method. The IK method adjusts the upper-body end effectors in real time to new constraints for planning manipulation motions. As a result, our method is able to automatically coordinate whole-body motions for planning virtual character locomotion and manipulation.

Keywords. Motion graph, inverse kinematic, virtual characters, locomotion, manipulation.

1 Introduction

With the developments of interactive applications such as games, film animations or training applications. Interactions and moves of virtual characters have to be realistic, well animated and controllable by the simulation to meet the needs of these applications. In a situation of virtual character locomotion and manipulation, and inspired from how humans may solve real motion planning problems. An algorithm must generate a realistic path to the object, position the character in a convenient location near object and generate a realistic manipulation motion.

In the literature to synthesis a virtual characters motion, different approaches were proposed like model-based approaches, which limit the space of possible motion and give it a flexible representation.

Manipulation motion synthesis were well-treated using model-based approaches, contrast to motions with low energy such as locomotion, which may suffer from the difficulty of construction and control [1].

On the other hand, data-driven approaches are based on the idea of the reuse of existing 3D motions data to generate new motions. The adaptation of these motions to new situations will typically be difficult. Because the motion could be extremely dynamic or contain constrained interactions with the environment. Several researches have been proposed to solve these issues like motion graphs approach, where locomotion synthesis was actively studied. Motion graphs allow an exploration of solutions in complex environments and preserve the naturalness of the captured motions.

Motion graphs have improved their performance to synthesis a natural locomotion in contrast to the manipulation motions, because motion graphs have problem to deal with addressing exact position of end-effectors. Also the inclusion of both locomotion and manipulation motions in the same graph will explode the graph size and complicate the search function.

The main contribution of this paper is the proposition of an approach, which guarantees a whole-body motion planning. By enjoying the benefits of data driven and inverse kinematics approaches for synthesizing realistic locomotion and manipulation motions. The idea is based on planning locomotion motion using a locomotion planner based motion graphs. The locomotion planner reaches different locations in the environment and searches for suitable position for the virtual characters near objects.

The locomotion planner is coordinated with a manipulation planner until a total whole-body motion is found for performing the manipulation action.

The manipulation planner is based on an iterative IK method to control the trajectories of the end-effectors. Our approach could be suitable for interactive application and Avoids problems related to the inclusion of both locomotion and upper-body actions in the same graph.

2 Related Work

In the literature locomotion synthesis has been an active area of study, especially for Data-driven approach where new motions are generated from a library of motion clips for producing realistic motions. Blending has been used to synthesize virtual character locomotion animation in many works as [2, 3, 4, 5]. The planification of motion around obstacles was presented a difficulty for these approaches. The synthesis of locomotion motions among obstacles was treated by [6] by proposing feature-based motion graphs with a channel search method. The motion graph was proposed the first time by [7] this structure is based on the idea if connecting the frames of high similarity in a database of motion capture examples. A graph search is performed once a motion graph is available to extract motions with desired properties [7, 8, 9]. Motion graphs have been used mostly for Planning locomotion, in contrast to other motion, because motion graph have the inconvenient of not precisizing the end effectors positions, which make the generation of motions that needs tight coupling to the environment very hard. Except exactly those motions are in the database. Whenever more constraints are added, fewer paths will become available [10].

The Inverse Kinematics (IK) and collision avoidance are used to solve the problem of synthesis of hyper-body motions, which implicates object manipulation like reaching, grasp and moving objects. The Inverse Kinematic is a method used to control the trajectories of the end-effectors in order to reach given targets [11]. To deal with simple kinematic chain, the best choice is analytical methods because these methods are

fast [12]. With complex articulated chain Numerical methods are proposed, these methods suffer from the computational time, this inconvenient is resolved by the heuristic method [13, 14, 15]. FABRIK (Forward and Backward Reaching Inverse Kinematics) proposed by [16, 17] handle with multiple chains with multiple end effectors and based on geometric algebra. This method produces visually realistic poses with low computational cost.

Works which aim is Whole-Body Coordination are based on the coordination of primitive motion skills to reach a complex motion. [18] planed locomotion sequences among obstacles using a discrete search to determine the best sequence of pre-computed leg motions. This approach was modified as a multimode RRT problem in [19] where the pre-computed motions were not necessary. [20] proposed the coordination of walking and pushing. [21] proposed the Coordination of walking and grasping using a manipulation planner, which synchronizes with motion capture sequences. [22] proposed coupling locomotion and hyper-body action where a locomotion planner based motion graph and the upper-body actions planner based blending method. [23] proposed an animation system that integrates different controllers to achieve a reaching task by utilizing path planner, motion blending and inverse kinematics.

The virtual character is required to interact with different objects in different locations in the surrounding environment, which involve the coordination of two skills locomotion and manipulation. The manipulation of objects requires posing the character in suitable position near objects. To solve this type of problem we use an inverse kinematic method to planning the manipulation motions coupling with a motion graphs planner to pose the character near objects.

3 Character Modeling

A character is represented by his skeletal structure and modeled as a chain composed of rigid links attached at their ends by rotating joints. The chains can be formalized as (Fig. 1). All joints with no children are labeled as end-effectors.

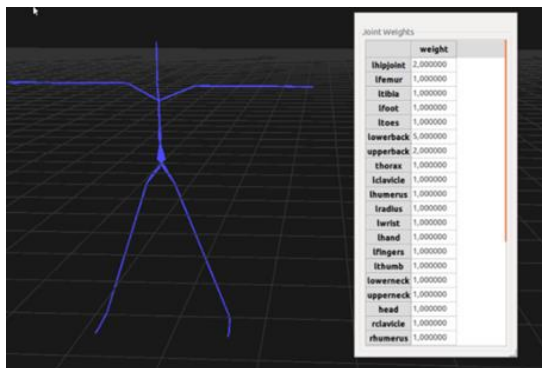


Fig. 1. ASF skeleton with list joints

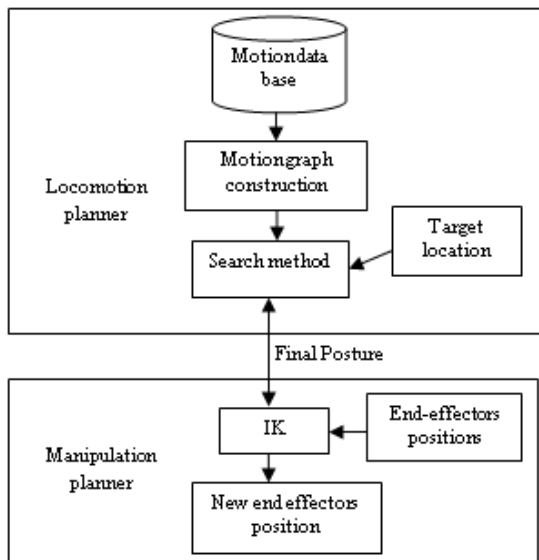


Fig. 2. System Overview

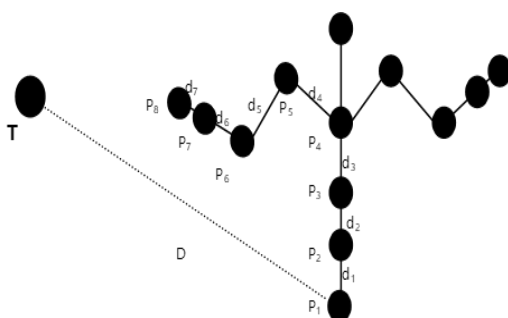


Fig. 3. Virtual character hyper body Part involved in the IK arm control

4 System Overview

The generation of a virtual character manipulation motions requires a correct body positioning. This is provided by the locomotion planner that search for suitable body placements nearby the action target location.

This combination is inspired from real human motion. If a virtual character wants to manipulate an object, he walks toward the object in order to help his reaching action.

Our approach is based on a locomotion planner able to place the character in a suitable location near target location and coordinated with a manipulation planner for planning manipulation motions. An overview of the proposed method is illustrated in (Fig.2).

4.1 Locomotion Planner

The locomotion graph is constructed similar to [7] starting by finding the candidate transition points then selecting from them the good enough for transitions, using a sampled 2D error or distance function. Straight forward blending will produce valid transitions. Pruning Graph phase is applied on the graph for eliminate problematic of dead and sinks nodes by calculating the largest strongly connected component of the graph. The task of the locomotion planner is to explore suitable body position in coordination with the upper-body inverse kinematic method until an overall whole-body motion for enabling the manipulation action is found.

The goal location of a virtual character may be a target location presented by a 2D path planner on the floor plan. The search method is used to explores the motion graph and generate the graph walks. We use a branch and bound algorithm [7] in our method to cull any branch incapable of giving an optima. Starts from the node in the motion graph containing the initial character pose. The algorithm explores the motion graph and generates multiple graph walks simultaneously. Each graph walk w has an error value $f(w)$ associated with it.

As the graph is explored, each edge is evaluated using a optimization criteria $g(w;e)$ to evaluates the additional error accrued by appending an edge e to the existing path w .

The total error $f(w)$ of a graph walk is:

$$f(w) = f([e_1, \dots, e_n]) \\ = \sum_{i=1}^n g([e_1, \dots, e_{i-1}], e_i). \quad (1)$$

By appending an edge to the graph walk, its $f(w)$ increases. If any graph walk has worse $f(w)$ than the current best graph walk w_{opt} , it is discarded. Because it is important to compute a w_{opt} with a low value of $f(w_{opt})$ as quickly as possible. Edges are chosen using a simple greedy heuristic. The search ends when a user-specified halting condition is satisfied. The $g(w; e)$ is defined to give some sort of guidance throughout the motion to find the desired motion. The branching factor of the search must be limited to only allow expansions nearby this path; the cost function $g(w; e)$ is defined as follows:

$$g(w, e) = \sum_{i=1}^n \|p'(s(e_i)) - p(s(e_i))\|, \quad (2)$$

p is the target given by the user, p' is the path estimated from motion graph, and $p(s)$ is the point on p whose arc-distance from the start p is s .

4.2 Manipulation Planner

For planning the manipulation action, a suitable body placement is needed. Our overall planning method based on finding a full body motion composed of two parts: locomotion for body positioning, and then manipulation action execution satisfying a given end-effector target location. Motion graph generate several possible trajectories nearby the action target that insures that the target is in the reaching range. A suitable proximity threshold is selected for a suitable distance, which will be the length of the character's arm from the root to the hand. Every time one branch of the locomotion graph expansion reaches a character position that is close enough to the action target, becomes a candidate initial posture for the manipulation planner. The manipulation planner use an iterative inverse kinematic FABRIK [17] method that has as input data the starting posture reached by the motion graph and the desired end effectors positions constraints.

The focus tends to be on arms control see (Fig. 3). Where T is the target position. $P_1, P_2, P_3, \dots, P_8$ are the joints positions. $d_1, d_2, d_3, \dots, d_7$ are the distances between each joint. The same thing for the left arm.

To reach the desired end effector position constraint. Fabrik uses points and lines instead of rotational angles to solve the IK, it divides the problem into two steps Forward and Backward. New joint position is calculated by keeping constant the length between joint i and joint $i-1$. The method starts from the last joint of the chain, which is the hand, and works forwards, adjusting each joint along the way.

Thereafter, in order to complete a full iteration the same procedure is repeated but this time it works backward starting from the root joint and moving outwards to the end effector. Because the new position of the root should not be different from its initial position. If the manipulation planner is not successful after a fixed number of iterations, because the distance between the target and the first joint (root) is higher than the summation of the length of each joints. The locomotion planner continues to expand towards additional candidate body placements near target location until the upper-body adjustment to the desired end effectors positions can be executed.

Whenever the locomotion planner generates a character position that is close enough to the target the forward and backward stages are applied. (Fig. 4) illustrates how the manipulation planner works, d is the distance between the target and the first joint (root), the initial pose of end effectors and target position and each point is expressed by the coordinate of Cartesian space.

5 Experimental Results

For the demonstration of our overall method, we investigate the results of the performed experiments. (Fig. 5(a)) shows the motion clips list and the weight of each joints used in the construction of the locomotion graph. (Fig. 5(b)) shows the generated point cloud from the distance metric.

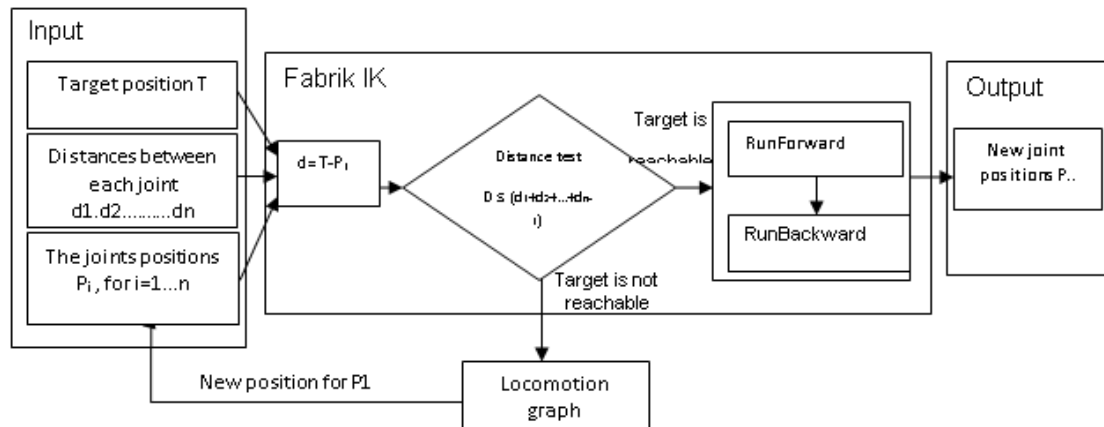


Fig. 4. Illustration of the manipulation planner

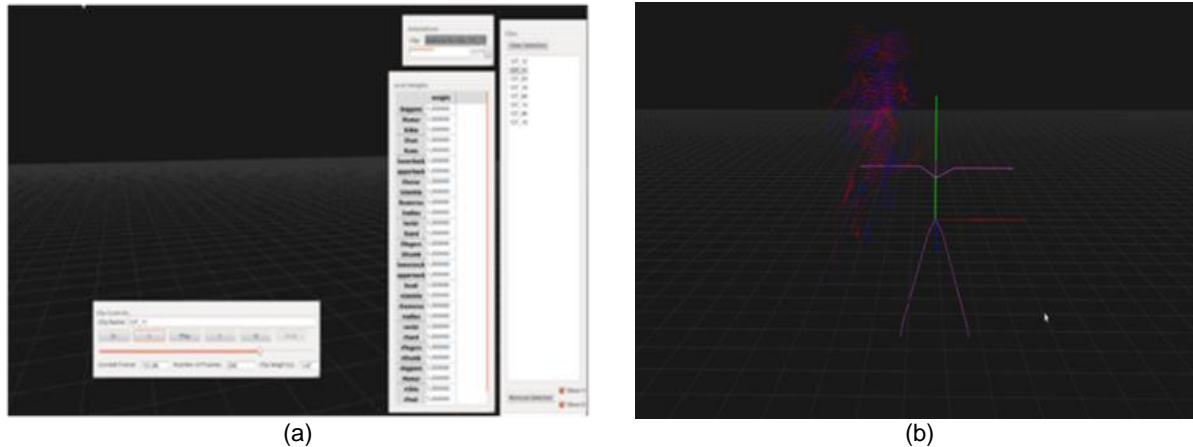


Fig. 5. (a) Motion clips list and joint weights, (b) Point cloud distance metric

For an example of a motion graph formed over 8 motion clips the construction time was 40 seconds. (Fig. 6) illustrates the generation of the graph walks, the locomotion graph is explored using a branch and bound search algorithm.

For a 2D path planned on the floor plan, the branching factor of the search method can be constrained to allow only expansions nearby this path.

Other results demonstrate the hyper-body motions adjustment to reach different end-effectors goal location using the inverse kinematic method Fabrik (Fig. 7) illustrates right and left hand adjustment using the IK method.

6. Conclusion and Discussion

In this paper, we present an approach for generating realistic whole body planning motion. The manipulation planner based Fabrik method is applied to the chain started from the root to the hand to reach given end-effectors goal location. The manipulation planner is coordinated with a locomotion planner based motion graph that search to pose the character in suitable position near objects. If the execution of the manipulation planner fails, the locomotion planner explores new positions for the character. Our hybrid approach allows a coordination of manipulation actions with locomotion.

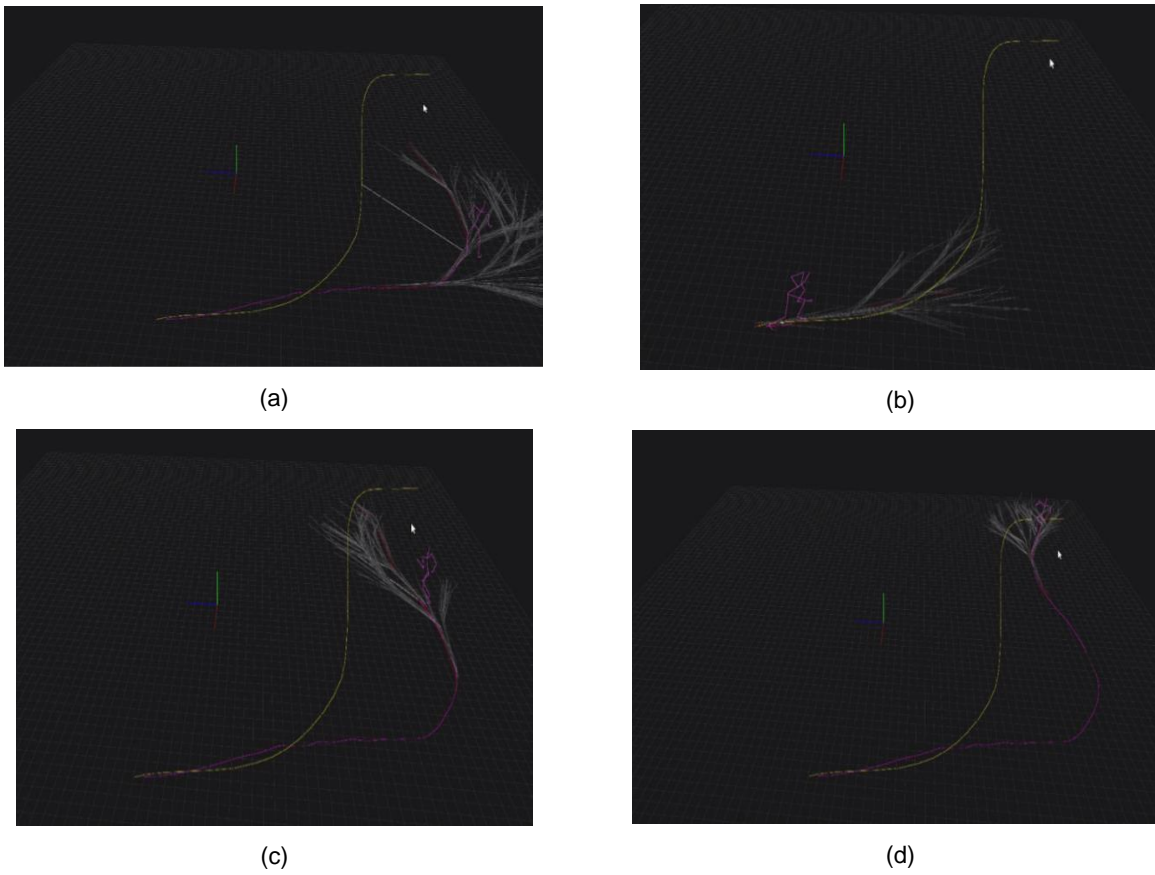


Fig. 6. (a, b, c, d) Examples of graph walks extracted from the motion graph

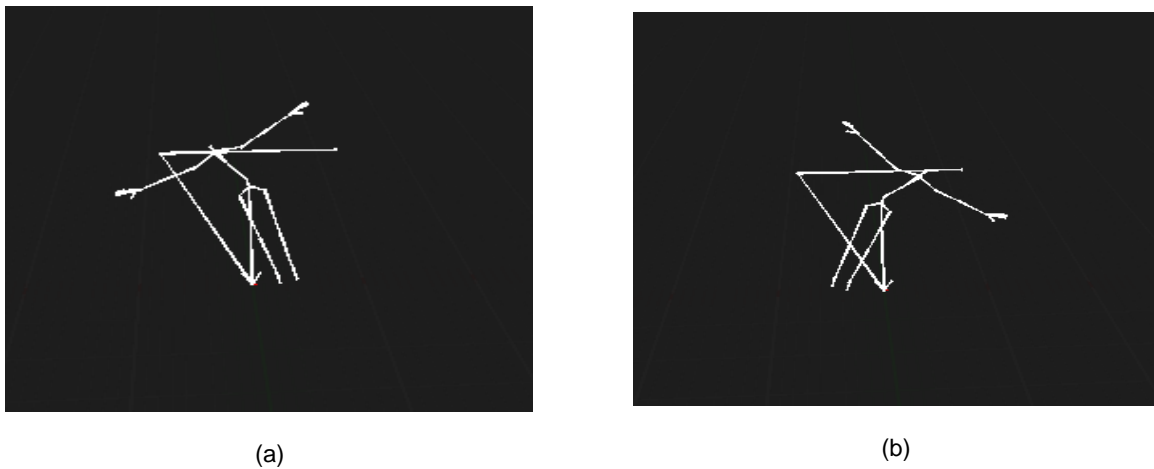


Fig. 7. (a, b) Right and left hand adjustment using the IK method

It proves be very efficient for planning object manipulation in different locations in the environment.

Avoids problems related to the inclusion of both locomotion and upper-body actions in the same graph.

Which are the explosion of the motion graph size and the complexity of the search method that may fail to find the right motion in the right time. Our approach could be suitable for interactive application.

References

1. **Yamane, K., Kuffner, J.J., & Hodgins, J.K. (2004).** Synthesizing animations of human manipulation tasks. *ACM Transactions on Graphics (TOG)*, Vol. 23, No. 3, pp. 532–539. DOI:10.1145/1186562.1015756.
2. **Pettré, J. & Laumond, J.P. (2006).** A motion capture-based control-space approach for walking mannequins. *Computer Animation and Virtual Worlds*, Vol. 17, No. 2, pp. 109–126. DOI: 10.1002/cav.76.
3. **Johansen, R.S. (2009).** *Automated semi-procedural animation for character locomotion*. Aarhus University.
4. **Park, S.I., Shin, H.J., & Shin, S.Y. (2002).** On-line locomotion generation based on motion blending. *Proceedings ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 105–111. DOI: 10.1145/545261.545279.
5. **Kwon, T. & Shin, S.Y. (2005).** Motion modeling for on-line locomotion synthesis. *Proceedings ACM SIGGRAPH/Eurographics symposium on Computer animation* pp. 29–38. DOI: 10.1145/1073368.1073373.
6. **Mahmudi, M. & Kallmann, M. (2013).** Analyzing locomotion synthesis with feature-based motion graphs. *IEEE transactions on visualization and computer graphics*, Vol. 19, No. 5, pp. 774–786. DOI: 10.1109/TVCG.2012.149.
7. **Kovar, L., Gleicher, M., & Pighin, F. (2008).** Motion graphs. *ACM SIGGRAPH 2008 classes*, pp. 1–10. DOI: 10.1145/1401132.1401202.
8. **Kovar, L. & Gleicher, M. (2004).** Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics (TOG)*, Vol. 23, No. 3, pp. 559–568. DOI: 10.1145/1015706.1015760.
9. **Heck, R. & Gleicher, M. (2007).** Parametric motion graphs. *Proceedings symposium on Interactive 3D graphics and games*, pp. 129–136. DOI: 10.1145/1230100.1230123.
10. **Van-Welbergen, H., Van-Basten, B.J., Egges, A., Ruttkay, Z.M., & Overmars, M.H. (2010).** Real Time Animation of Virtual Humans: A Trade-off Between Naturalness and Control. *Computer Graphics Forum*, Vol. 29, No. 8, pp. 2530–2554. DOI: 10.1111/j.1467-8659.2010.01822.x.
11. **Kallmann, M. (2008).** Autonomous object manipulation for virtual humans. *Motion Planning and Autonomy for Virtual Humans (SIGGRAPH'08)*, pp. 1–47.
12. **Tolani, D., Goswami, A., & Badler, N.I. (2000).** Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, Vol. 62, No. 5, pp. 353–388. DOI: 10.1006/gmod.2000.0528.
13. **Welman, C. (1993).** *Inverse kinematics and geometric constraints for articulated figure manipulation*. Simon Fraser University.
14. **Wang, L.C. & Chen, C.C. (1991).** A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, pp. 489–499. DOI: 10.1109/70.86079.
15. **Kulpa, R. & Multon, F. (2005).** Fast inverse kinematics and kinetics solver for human-like figures. *Proceeding International Conference on Humanoid Robots, (5th IEEE-RAS)*, pp. 38–43. DOI: 10.1109/ICHR.2005.1573542.
16. **Aristidou, A. & Lasenby, J. (2011).** FABRIK: a fast, iterative solver for the inverse kinematics problem. *Graphical Models*, Vol. 73, No. 5, pp. 243–260. DOI: 10.1016/j.gmod.2011.05.003.
17. **Aristidou, A., Chrysanthou, Y., & Lasenby, J. (2016).** Extending FABRIK with model constraints. *Computer Animation and Virtual Worlds*, Vol. 27, No. 1, pp. 35–57. DOI: 10.1002/cav.1630.
18. **Kuffner, J.J., Nishiwaki, K., Kagami, S., Inaba, M., & Inoue, H. (2001).** Footstep planning among obstacles for biped robots. *Proceedings. Intelligent Robots and Systems, IEEE/RSJ International Conference*, pp. 500–505. DOI: 10.1109/IROS.2001.973406.
19. **Kallmann, M., Bargmann, R., & Mataric, M. (2004).** Planning the sequencing of movement primitives. *Proceedings of the international conference on simulation of adaptive behavior (SAB)*, pp. 193–200.
20. **Hauser, K., Ng-Thow-Hin, V., & Gonzalez-Baños, H. (2010).** Multi-modal motion planning for a

humanoid robot manipulation task. *Robotics Research*, pp. 307–317. DOI: 10.1007/978-3-642-14743-2_26.

21. **Shapiro, A., Kallmann, M., & Faloutsos, P. (2008).** Interactive motion correction and object manipulation. *Proceedings symposium on Interactive 3D graphics and games*, pp. 137–144. DOI: 10.1145/1401132.1401208.
22. **Huang, Y., Mahmudi, M., & Kallmann, M. (2011).** Planning humanlike actions in blending spaces. *International Conference on Intelligent Robots and*

Systems, IEEE/RSJ, pp. 2653–2659. DOI: 10.1109/IROS.2011.6095133.

23. **Feng, A.W., Xu, Y., & Shapiro, A. (2012).** An example-based motion synthesis technique for locomotion and object manipulation. *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 95–102. DOI: 10.1145/2159616.2159632.

*Article received on 11/04/2018; accepted on 29/10/2019.
Corresponding author is Kalouache Saida.*