2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

An online trajectory planning method for visually guided assisted reaching through a rehabilitation robot

C. Loconsole, R. Bartalucci, A. Frisoli, M. Bergamasco

Abstract— **Several manipulators or exoskeleton are characterized by having a concave workspace in the operational space due to mechanical limits. This article proposes an online trajectory planning method for performing visually guided assisted reaching through a rehabilitation robotic exoskeleton, the L-Exos, in its concave workspace. To evalute the proposed methodology in a rehabilitation application, we set-up a computer vision based system that can automatically identify target objects in the workspace and generate a robot assisted movement to reach them through the L-Exos.**

I. INTRODUCTION

The objectives of this work is the development and testing of an online trajectory planning method for visually guided assisted reaching through a rehabilitation robot. Rehabilitation robotics is a growing field where robots are applied to assist the patient during motion recovery after a main neurological damage. The Light-Exoskeleton, or L-Exos for short [1], is a rehabilitative anthropomorfic robot for upper limbs that can actively guide the patient through the execution of upper limb movements. One of the important research issues in the field of rehabilitation or assistive robotics, is the development of novel user control interfaces that can understand the user intention of movement, employing for instance Brain Computer Interface (BCI), gaze and camera devices.

In our research, we are employing an upper limb exoskeleton robot to guide the movement of the patient towards objects that are presented to him by the therapist. The objects are automatically on-line recognized by a vision system through a bird's eye-view camera put on the top of the scene and on-line motion tracking of the object is established. But not all the points of the space that are visible to the user or to the camera are also reachable by the end effector of the manipulator, so the calculation of the closest reachable point to the target (proxy point) is needed. One of the main goals of this paper is the proposal and evaluation of a trajectory planning method that the manipulator must follow to reach the target seen by the camera or the proxy point.

Moreover, while the reachability of a point in 3-D space is easily verifiable with the inverse kinematics method, the planning of a minimum distance trajectory between two points is not trivial in concave manipulator workspaces. In fact, in this case, there is not always a straight line that links the two points. For example in Fig. 1 A and B can be connected with a straight line, while A and C cannot. The generation of a suitable trajectory that connects any two points of the workspace is required in this case.

Different numerical and analytical methods are available in literature for the exact computation and representation of a robot workspace. Numerical techniques are reported by Kumar and Waldron [2], Yang and Lee [3], and Tsai and Soni [4], among many others. The advantage of these schemes over analytical approaches is that kinematic constraints can be easily included. Among the analytical approaches to workspace characterization, a topological analysis of robot workspace is given by Gupta and Roth [5] and Gupta [6]. Further studies of workspace analysis were reported by Freudenstein and Primrose [7] and by Lin [8]. A complementary issue to the determination of workspace is the path planning in operative space with costraints. A large number of motion planning methods have been proposed over the last years, especially in the area of mobile robotics, generally based upon three main approaches to path planning [9]: the potential field approach, the roadway approach, and cell decomposition. Each of these methods can be performed in the configuration or operative space: in particular cell decomposition can be exact or approximate and can search the free-collision graph for a suitable path. To find a path in a decomposed region many algorithms have been developed. One of the most known and efficient optimum algorithm is the $A*$ [10]. Other suboptimal (but faster) algorithms for this purpose are breadth-first, depth-first and greedy algorithms [11].

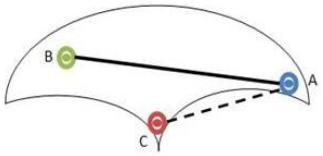


Fig. 1. A simplified example of concave workspace issue in 2D

This article proposes a method to on-line plan a "suboptimum" minimum distance trajectory for fixed and mobile targets in a particular concave manipulator workspace. The method can deal with manipulator redundancy, offers a model of the workspace and a way to easily visualize it. In the article the proposed method is evaluated through a computer vision based system and tested in a robotic-aided rehabilitation application of robot assisted reaching.

II. THE EXPERIMENTAL SETUP

The overall experimental set-up is made of three main components: the Light-Exos, the controller and the computer vision system (Fig. 2).

This work was developed in the context of BRAVO project funded by Project Seed of IIT (Italian Institute of Technology) http://www.iit.it/en/news/general/project-seed-results.html

C. Loconsole, A. Frisoli, M. Bergamasco are affiliated with the Perceptual Robotics Lab (PERCRO), Scuola Superiore SantAnna, Pisa, Italy, {c.loconsole, a.frisoli}@sssup.it

R. Bartalucci is with Scuola Superiore SantAnna, Pisa, Italy, r.bartalucci@sssup.it

Fig. 2. Experimental setup

A. The Light-EXOS

The L-Exos [12] was designed under biologically inspiration of a human-arm minimal configuration, see Fig. 3. It is composed of five degrees-of-freedom (DoF), four of them fully actuated and the last one is used for measuring the wrist pronation/supination motion. In this paper only the first four fully actuated DoFs will be used. The L-Exos has three contact points at the level of the user's body: the shoulder, the forearm, and the wrist (handpalm) [13]. Notice that the basic task resembles a human arm motion: (1) Adduction/ abduction; driven by joint q_1. (2) Flexion/extension; driven by joint q_2 and q_4. (3) Internal/external rotation;driven by joint q_3. (4) Pronation/supination; given by q_5, see Fig. 3. The joint motion span is: $-19° \leq q_1 \leq 90°$, $-90° \leq q_2 \leq 41°$, $-62° \leq q_3 \leq 62°$, $-94° \leq q_4 \leq 0°$. For more details on open kinematic chain of the L-Exos see [14]. The manipulator presents a position redundancy, since each point of the workspace can be reached with different poses of the shoulder of the manipulator.

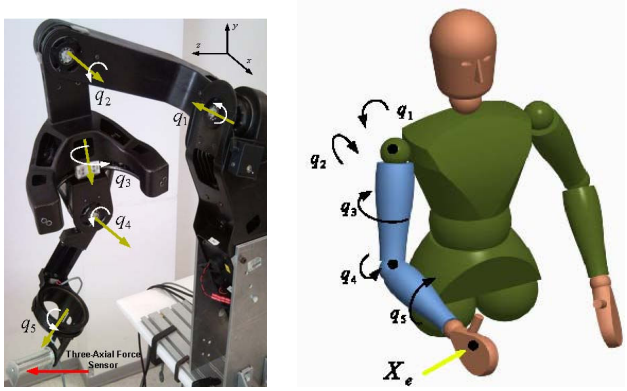


Fig. 3. L-Exos kinematics

B. The xPC-Target based controller

The control software was developed in Simulink and compiled for xPC-Target system. It receives the position of the target in 3-D space from the computer vision system and controls the L-Exos to reach it, or in the case of not reachability of the target, to reach the closest point to the target that belongs to the manipulator workspace (the proxy point).

C. The computer vision system

The computer vision system consists of two parts: an High Definition webcam and an OpenCV based software. The webcam is on the top of a rigid structure anchored to the L-Exos and can see the work area of the L-Exos from the top with a bird's eye-view. The OpenCV software is based on a color matching recognition of the target and provides via UDP connection to the controller the 3-D position of the target directly in the L-Exos reference framework.

III. A GREEDY ALGORITHM FOR ON-LINE PATH PLANNING TO MOBILE TARGETS

A. Workspace quantization and representation

In order to make possible in real-time the test of reachability of points in the operative space by the L-Exos, a discretized representation of the workspace was performed leading to the creation of a 3D look-up table. This was also a necessary requirement to deal in real-time with the inverse kinematic algorithm of the exoskeleton (exoskeleton presents one degree of freedom of redundancy). The algorithm used for the discretization was the cell decomposition method applied to the operative space. The cell decomposition algorithm was composed of four steps. In the first step the bounding-box (BB) of the overall workspace was determined by exploration of the reachable workspace through direct kinematics. For the L-exos the BB measured was $130\times120\times100$=1.560.000 cm^3. The second step was the subdivision of the BB in volumetric cells (voxels). Clearly the choice of the proper voxel dimension represents a compromise between precision and memory occupancy. A voxel dimension of 1 cm side was chosen, leading to a total memory occupancy of 1.5 MB. In the third step, a reachability-test (using inverse kinematics) was performed on the centroid of each voxel. Since the L-Exos is 4-joints redundant manipulator, a parametrized numerical inverse kinematics was found through geometrical and algebraic considerations. After a voxel labeling, the subdivision of the workspace in inner and boundary workspaces was possible. In this way during the path planning phase it is possible to avoid to work on the boundary of the workspace (that in this case has a thickness of 1 cm) and so kinematic singularities and loss of functionality issues are completely solved. In Fig. 4 some views of L-Exos workspace are shown: 3D look-up table values were plotted using isosurfaces on MATLAB software. As we can see the workspace is concave due to kinematic chain of the L-Exos and mechanical constraints on joint motion spans.

B. Trajectory planning method

One of the main goals of this paper is to plan a trajectory that the manipulator must follow to reach the target seen by a camera. If the target is out of the workspace, a proxy point is taken as new target: the proxy point is the nearest reachable point to the target, onto the straight line between current position and target itself. Through bisection method applied on this line, it can be proved that a proxy point

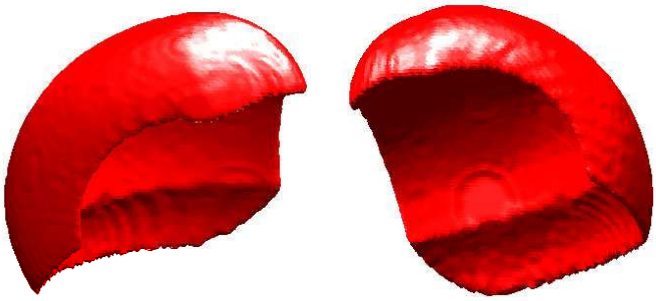


Fig. 4. L-Exos workspace from different views

can be found with 7 look-up table queries. It was chosen to perform a planning in operative space to have predictable trajectories: in fact in robotic aided rehabilitation tasks it's better to have minimum-length paths to approximate human behavior. Although easier, a joint space planning (joint space is convex) would generate unpredictable paths in operative space. Moreover, since in the future we would add the mobile obstacle avoidance feature, the planning in operative space using the space discretization permits to online plan the best trajectory that avoids the impact with one or more mobile obstacle. If current and goal points can be linked by a fully-reachable straight line path, path planning is very easy: the minimum-length path is the straight line.

The problem arises when the straight line is not completely inside the workspace, due to concavity. In this case starting from the previously built-in 3D look-up table, the path can be planned over the cells of the discretized workspace [9]. Workspace can be considered a graph, with voxels as nodes and mini-path between centroids of adjacent cells as arcs. Path planning problem is so reduced to a graph searching problem. One of the most known optimal shortest path searching algorithms for graphs is $A*$. However it's not very fast and not so suitable for a real-time application like ours. So in this paper $A*$ will be used as starting point for other algorithms.

The searching algorithm family it was chosen was the so-called "Greedy Algorithms". A greedy algorithm (GA) is any algorithm that makes the locally optimal choice at each stage with the hope of finding the global optimum. They find suboptimal (in this case longer) solutions, but their computation time is smaller. We'll try some versions of GAs in our configuration, investigating suboptimality magnitude and computation time saving.

1) Greedy algorithm: In general, a GA starts from the current node and moves on arcs until the goal node is reached, trying to minimize at each step a cost function f. The choice of this cost function characterizes the behavior of a GA. One of the issues of GAs, other than it's suboptimality, is that it does not guarantee to find a solution (for example in case of local minimum points). In this paper we try two different kinds of GAs, that here are called *hill* and *beam*. The *hill* algorithm is the classical hill-climbing algorithm: the f function is equal to the distance between the investigated node and the goal node:

$$f(\mathbf{x}) = ||\mathbf{x} - \mathbf{x}_g||,$$

where $\mathbf{x}$ and $\mathbf{x}_g$ is the position in the operative space of respectively investigated node and the goal node. The *beam* algorithm, instead, has the following function cost:

$$f(\mathbf{x}) = \left|\left|\mathbf{x}_c + k\frac{\mathbf{x}_g - \mathbf{x}_c}{||\mathbf{x}_g - \mathbf{x}||} - \mathbf{x}\right|\right|.$$

At each step, a vector of length k is generated starting from current node ($\mathbf{x}_c$) and going towards the goal node $\mathbf{x}_g$. The cost function is the distance between the end of this vector and the investigated node. Parameter k has to be chosen to minimize path lengths: it will be investigated later in paper. In convex spaces this two algorithms find length-equal paths, but the *beam* tries to follow the ideal straight line linking starting ($\mathbf{x}_s$) and goal ($\mathbf{x}_g$) nodes (Fig 5).

As mentioned before a comparison method is needed to compare different GAs and choose the one that best fit our case. So Montecarlo method was used for generating several numbers of random start and target points that can not be linked by trivial straight line path. For each couple of nodes, a path is generated by all algorithms that we are investigating. At the end, mean performance values are compared.

2) Determination of k parameter for beam *algorithm:* First of all k parameter for *beam* algorithm must be chosen. A bunch of *beam* algorithm with different k parameter are compared by Montecarlo method. It was chosen $k = 5$ cm because this value showed best performances.

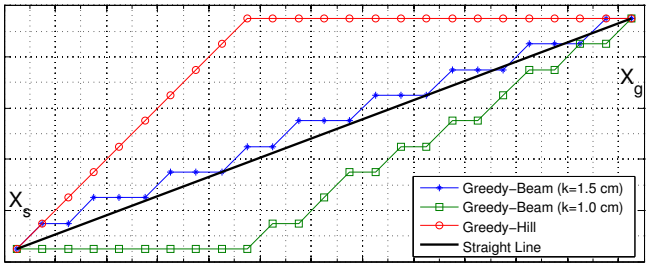


Fig. 5. Examples of path generation between x_s and x_g using *beam* and *hill* algorithms

C. Algorithm Comparison Results

Hill and *beam* algorithms are compared and then their suboptimality (comparing to optimal $A*$) will be quantified taking into account also their computation time. Four algorithms are compared: 1) *hill*; 2) *beam* ($k = 5$ cm); 3) optimal $A*$, with classical heuristic $h(\mathbf{x}) = ||\mathbf{x} - \mathbf{x}_g||$ [10]; 4) *suboptimal* $A*$, with amplified heuristic $h(\mathbf{x}) = 1.5||\mathbf{x} - \mathbf{x}_g||$ (quicker than the previous, [18]). Six hundreds (600) simulations have been performed: results are shown in Fig. 6. First of all it can be noted a huge computation time saving with suboptimal algorithms: $A*$ can not be used in real time context, while the others are suitable. Looking at suboptimality on decimated paths, *beam* and *suboptimal* $A*$ algorithms are the best. In particular, *beam* is far better than classical *hill* algorithm: that's because, trying to follow ideal straight line, it get more straight decimated paths. In conclusion *beam* algorithm represents a good compromise: it is quick and does not generate so longer paths than "A*" ($\approx 2\%$ longer). These paths are acceptable and the algorithm can be used for the paper's purpose.

1) Time saving queries considerations: Another relevant aspect of the proposed method is the straight line trajectory planning in concave workspace. Usually to generate this kind of trajectories in such a workspace some sample points on the joining line between two points can be taken and tested using inverse kinematics. However it is not completely assured that mini-path among these sample points are reachable. With the proposed method, instead, querying 3D look-up table it has been proved that if a sufficient number of sample points of the line are in the voxels that belong to the inner workspace, the line is fully reachable. Moreover under the time computation point of view, the proposed method is very faster than the application of the inverse kinematics to sample points. For example, in our case, if 100 sample points are taken on the line, with the application of the inverse kinematics the algorithm takes 1 s, while the proposed method takes 0.01 s.

2) Smoothing: After algorithm execution, we get a discretized path. It would not be very comfortable for the L-Exos users to follow this path. So a smoothing process is needed. It consists in two steps: 1) the path is "decimated", so only one point every n is taken; 2) a spline curve is generated from the path. Spline is composed by pieces of Bezier Curves: for every three decimated points a curve is generated and then all of them are connected, assuring second derivative continuity [15]. It has been proved that for a decimation with $n \leq 3$, resulting paths are fully reachable. In Fig. 7 an example of path generation and smoothing between two points that are not linkable by straight line path is reported.

3) Speed profile: Smoothed path is finally interpolated at equispaced points (1 mm). These points are sent by an operative space position scheduler to the control system, so that the motion will follow a trapezoidal speed profile.

4) Online path junction: If target moves or changes during a path execution, the system has to react and change its trajectory in real-time. In our solution, when the target changes, a new trajectory starting from current point is calculated. To avoid sharp trajectory changes, a gradual interpolation between old and new trajectory is performed (an example will be shown in a practical application later on this paper). To face synchronization issues related to trajectory change, a small buffer is used.

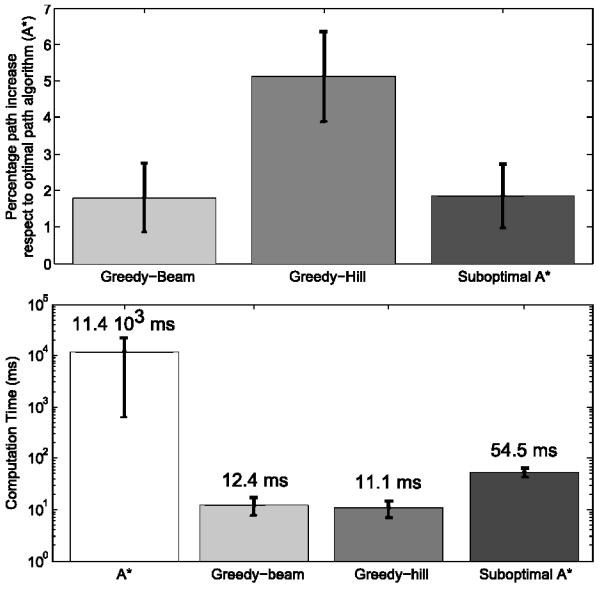


Fig. 6. Algorithm comparison results: decimated path lengths and computation times averages and standard deviations

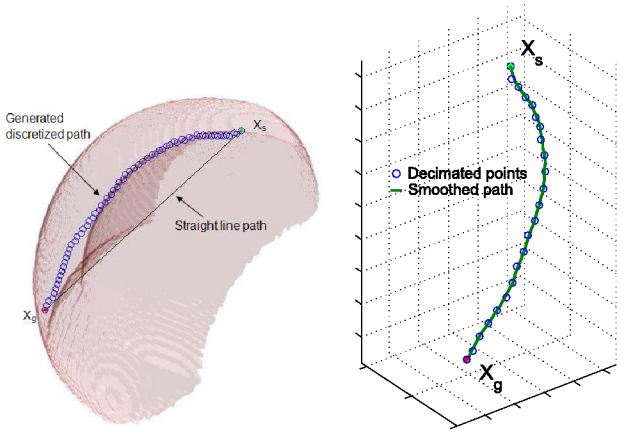


Fig. 7. Example of path generation and smoothing between two points that are not linkable by straight line path

IV. THE CONTROL SYSTEM

Due to the kinematic redundancy of the manipulator, the reference trajectory in operative space was converted in a joint trajectory through a classical pseudoinverse method [16]. In a few words, joint values $\mathbf{q}_r$ were obtained integrating:

$$\dot{\mathbf{q}}_r = \mathbf{J}^+(\dot{\mathbf{x}}_r + \mathbf{K}\mathbf{e}) + (\mathbf{I}_3 - \mathbf{J}^+\mathbf{J})\dot{\mathbf{q}}_0,$$

where $\mathbf{x}_r$ is reference trajectory, $\mathbf{e} = \mathbf{x}_r - \mathbf{x}$ is position error, $\mathbf{K}$ is a gain matrix, $\mathbf{I}_3$ is the 3x3 unit matrix and $\dot{\mathbf{q}}_0$ is a reference joint velocity. Pseudoinverse $\mathbf{J}^+ = \mathbf{W}^{-1}\mathbf{J}^T\left(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T\right)^{-1}$ is weighted by matrix $\mathbf{W}$ and $\dot{\mathbf{q}}_0 = -k_0\left(\frac{\partial w(q)}{\partial \mathbf{q}}\right)^T$ tries to minimize $w(\mathbf{q})$ function. It is defined as distance from joint end-points:

$$w(\mathbf{q}) = \sum_{i=1}^{4} \frac{(q_i - q_{c,i})^2}{(q_i^+ - q_i^-)^2},$$

where q_i^+ and q_i^- are joint constraints and $q_{c,i}$ is joint centered position. In this way cinematic redundancy is exploited to stay away from joint constraints. Reference joint values are sent to a PD controller to get torques needed by the manipulator:

$$\tau = \mathbf{K}_p(\mathbf{q}_r - \mathbf{q}) + \mathbf{K}_v(\dot{\mathbf{q}}_r - \dot{\mathbf{q}}),$$

where τ is motor torque, $\mathbf{K}_p$ and $\mathbf{K}_v$ are proportional and derivative matrix gain.

A. Computer vision based target recognition

The object recognition method is color based. More precisely, once the color of the target object (TC) is known, a set of colors "near" to TC is defined. All the pixels of the image captured from the webcam are scanned and if the pixel belongs to the defined color range, the corresponding pixel in a mask is set. After the scanning, the pixels corresponding to the centroid of the object is calculated through the mask.

For the next steps we use the camera model reported in [17]. If $\tilde{\mathbf{m}} = [u,\ v,\ 1]^T$ is a 2D point in homogeneous coordinates, $\tilde{\mathbf{M}} = [X,\ Y,\ Z,\ 1]^T$ is 3D point in homogeneous coordinates and the camera is modeled by the usual pinhole, the relationship between a 3D point $\tilde{\mathbf{M}}$ and its image projection $\tilde{\mathbf{m}}$ is given by:

$$s \cdot \tilde{\mathbf{m}} = \mathbf{A}[\mathbf{R}_c \ \ \mathbf{t}_c]\tilde{\mathbf{M}} \quad \text{with} \quad \mathbf{A} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where s is an arbitrary scale factor; $[\mathbf{R}_c \ \ \mathbf{t}_c]$, called the extrinsic parameters, is the rotation and translation which relates the world coordinate system to the camera coordinate system; $\mathbf{A}$ is called the camera intrinsic matrix, and u_0, v_0 are the coordinates of the principal point; α and β are the scale factors in image u and v axes. To determine the relationship between a 2D point $\tilde{\mathbf{m}}$ and its corresponding 3D point $\tilde{\mathbf{M}}$ we find a relation that depends on one parameter h that is the distance between the object and the camera:

$$\begin{cases} \mathbf{c} = \mathbf{A}_i^{-1}(\mathbf{m} - \mathbf{t}_i)h & \text{with} \quad \mathbf{A}_i = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \\ & \mathbf{t}_i = [u_0 \ \ v_0]^T \\ \mathbf{M} = \mathbf{R}_c^{-1}\left([\mathbf{c};\ \ h] - \mathbf{t}_c\right) \end{cases}$$

To estimate the value of the parameter h we use circle shaped objects. Once the diameter of the object is measured in pixels, we can interpolate the value of the h parameter and find the world coordinates of the centroid of the object. The last step of this module is the change of reference framework from that of the camera to the L-Exos one.

In Fig. 8 the block diagram of the entire system is reported.

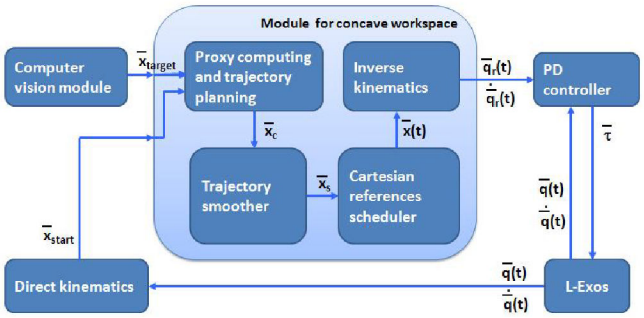


Fig. 8. Block diagram of the entire system

V. RESULTS AND EXPERIMENTAL EVALUATION OF PERFORMANCE

A practical case study is now presented to show the results obtained with the application of the proposed method on the entire system. In this robotic aided rehabilitation case study, a therapist can move a target that a patient with motor problems to the arm has to reach with the L-Exos support. In this way the therapist can easily tune instantaneously the rehabilitation therapy on the particular patient and follow more patients during the day because with the L-Exos support, the amount of the fatigue related to movement of the patient limbs is drastically decreased. To make clear the case study, the target position will not be continuously changed, but it will be moved "directly" in some default positions.

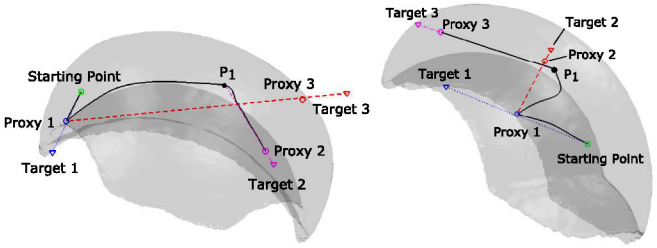


Fig. 9. End-effector trajectory on workspace during application example

The three chosen locations (*Target 1*, *Target 2* and *Target 3*) appear in Fig. 9 with the 3D representation of the workspace of the L-Exos. Note that all three targets are outside the workspace of the L-Exos, then it is necessary to calculate the corresponding proxy points inside the workspace (*Proxy 1*, *Proxy 2* and *Proxy 3* in Fig. 9). Moreover, as we can see in Figure 9, some paths are not straight due to the concavity of the workspace (e.g. from *Proxy 1* to *Proxy 2*).

In particular, the studied time line in the case study is:

- at time $t_1 = 4$ s, the target is in *Target 1* position and the end-effector of the L-Exos is in the *Starting Point*. *Proxy 1* point is calculated on the joining line between the *Starting Point* and the *Target 1* point;
- once *Proxy 1* is reached, at time $t_2 = 9.5$ s, the target moves to *Target 2* point. *Proxy 2* is calculated on *Proxy 1 - Target 2* joining line. The end effector of the L-Exos then moves towards *Proxy 2* position;
- during the *Proxy 2* point reaching phase, at time $t_3 = 14$ s, the end effector of the L-Exos has moved from *Proxy 2* to P_1 point. At this time the target moves from *Target 2* to *Target 3*. *Proxy 3* point is calculated on the P_1 (current position) and *Target 3* point joining line. Thanks to online path junction, a smooth trajectory calculated as gradual interpolation between the old and the new trajectory, allows to the end effector of the L-Exos to reach the *Proxy 3* point.

In Fig. 10 the end effector and the proxy positions in the three spatial coordinates are reported as function of time. Fig. 11 shows three images captured by the HD webcam at various moments of the case study. In particular in Fig. 11(a) is reported the system state at time t_1. It contains the reference framework and the contour of the workspace at the target height. Fig. 11(b) shows the state of the system in an instant between t_1 and t_2. Fig. 11(c) shows the state of the system in an instant between t_2 and t_3.

VI. CONCLUSIONS AND FUTURE WORKS

We presented a method for trajectory planning for overcoming all the issues related to the concavity of the workspace of the L-Exos. To achieve this goal we discretized and modeled the L-Exos workspace in an easy way. Moreover to show how the proposed method works and a possible use of it, for example in robotic-aided rehabilitation field, we presented a practical application with the computer vision help. With this application, a therapist can move a target that the patient has to reach with the help of the L-Exos. However also the use of the proposed method in other robotics field can be investigated. It is important to notice that the proposed method is suitable also to every kind of exoskeleton and

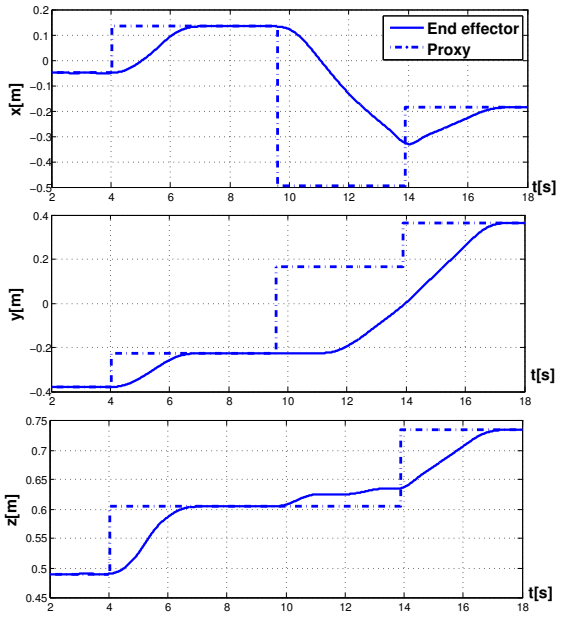


Fig. 10. End-effector coordinates during application example

manipulator, also those that have a convex workspace. In future work the proposed method should be extended with a gaze-tracking device in order to make possible the choice of the target to reach accordingly to the L-Exos user gaze.

REFERENCES

[1] A. Frisoli, A. Montagner, L. Borelli, F. Salsedo, M. Bergamasco, "'A force-feedback exoskeleton for upper limb rehabilitation in Virtual Reality"', *Applied Bionics and Biomechanics 6(2), 115-126*, 2009

[2] A. Kumar, K.J. Waldron, "'The workspaces of a mechanical manipulator"', *ASME J. Mech. Des. 103, 665672*, 1981

[3] D.C.H. Yang, T.W. Lee, "'On the workspace of mechanical manipulators"', *ASME J. Mech. Trans. Autom. Des. 105, 6269*, 1983

[4] Y.C. Tsai, A.H. Soni, "'An algorithm for the workspace of a general n-R robot"', *ASME J. Mech. Trans. Autom. Des. 105, 5257*, 1985

[5] K.C. Gupta, B. Roth, "'Design considerations for manipulator workspace"', *ASME J. Mech. Des. 104, 704711*, 1982

[6] K.C. Gupta, "'On the nature of robot workspace"', *Int. J. Robot. Res. 5(2), 112121*, 1986

[7] F. Freudenstein, E. Primrose: "'On the analysis and synthesis of the workspace of a three-link, turning-pair connected robot arm, *ASME J. Mech. Trans. Autom. Des. 106, 365370*, 1984

[8] C.C. Lin, F. Freudenstein, "'Optimization of the workspace of a three-link turning-pair connected robot arm"', *Int. J. Robot. Res. 5(2), 91103*, 1986

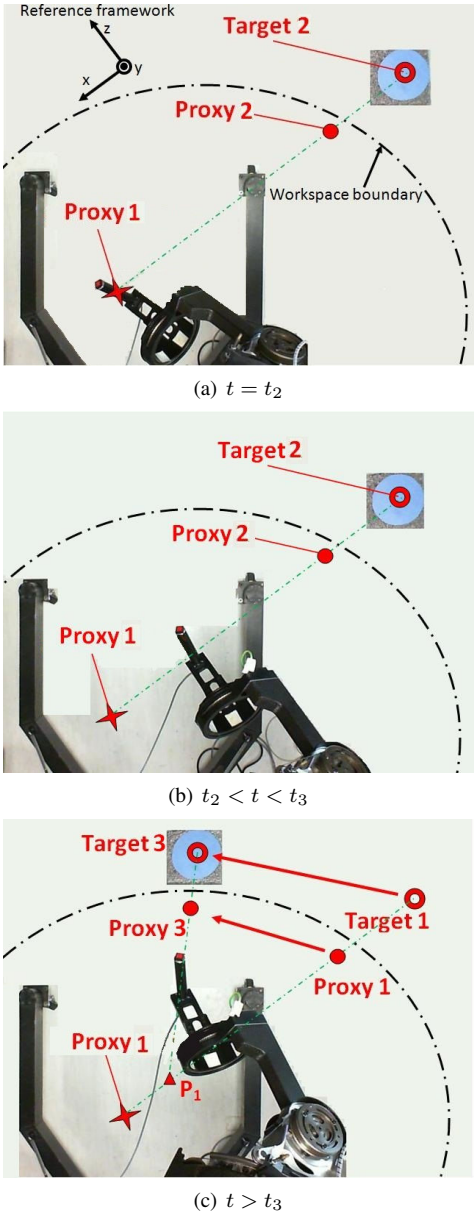
[9] J. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 1991

[10] Hart, P.E.; Nilsson, N.J.; Raphael, B.; , "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *Systems Science and Cybernetics, IEEE Transactions on* , vol.4, no.2, pp.100-107, July 1968

[11] Cormen, Leiserson, and Rivest, *Introduction to Algorithms* , MIT Press, 2001

[12] Frisoli, A., Rocchi, F., Marcheschi, S., Dettori, A., Salsedo, F., and Bergamasco, M. (2005), "'A new force- feedback arm exoskeleton for haptic interaction in virtual environments"', *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005, World Haptics 2005. First Joint* vol., no., pp. 195- 201, 18-20 March 2005

[13] Montagner, Alberto; Frisoli, Antonio; Borelli, Luigi; Procopio, Caterina; Bergamasco, Massimo; Carboncini, Maria C.; Rossi, Bruno; , "'A pilot clinical study on robotic assisted rehabilitation in VR with an arm exoskeleton device"', *Virtual Rehabilitation, 2007*, vol., no., pp.57-64, 27-29 Sept. 2007



(a) $t = t_2$

(b) $t_2 < t < t_3$

(c) $t > t_3$

Fig. 11. L-Exos seen by the HD camera in different times t during application example.

[14] Lugo-Villeda, L.I.; Frisoli, A.; Sandoval-Gonzalez, O.; Padilla, M.A.; Parra-Vega, V.; Avizzano, C.A.; Ruffaldi, E.; Bergamasco, M.; , "'Haptic guidance of Light-Exoskeleton for arm-rehabilitation tasks"', *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on* , vol., no., pp.903-908, Sept. 27 2009-Oct. 2 2009

[15] Kwangjin Yang; Sukkarieh, S.; , "An Analytical Continuous-Curvature Path-Smoothing Algorithm," *Robotics, IEEE Transactions on* , vol.26, no.3, pp.561-568, June 2010

[16] B. Siciliano; L. Sciavicco; L. Villani; G. Oriolo , *Robotics: Modeling, Planning and Control*, Springer, 2010

[17] Zhengyou Zhang; , "'Flexible camera calibration by viewing a plane from unknown orientations"', *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on* , vol.1, no., pp.666-673 vol.1, 1999

[18] Pearl, Judea; Kim, Jin H.; , "Studies in Semi-Admissible Heuristics," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.PAMI-4, no.4, pp.392-399, July 1982