

3D-RAS: A New Educational Simulation Tool for Kinematics Analysis of Anthropomorphic Robotic Arms*

T. J. MATEO SANGUINO and J. M. ANDÚJAR MÁRQUEZ

Universidad de Huelva, Dpto. de Ing. Electrónica, Sistemas Informáticos y Automática, Ctra. Huelva-La, Rábida s/n, 21819 Palos de la Frontera (Huelva), Spain. E-mail: tomas.mateo@diesia.uhu.es

The present work launches a new educational simulation tool aimed at making kinematic analysis of anthropomorphic robotic arms more attractive and practical. This work focuses on the development of a new educational simulation tool for the “Robotics” subject in the third year of the “Computer Engineering” and “Electronic Engineering” degrees at the University of Huelva (Spain). The developed tool (3D Robotic Arm Simulator, 3D-RAS) is aimed at engineering practices and demands students no programming knowledge. Its highly graphic interface allows easy and simple definition of both parameters and geometry in serial arms with up to five degrees of freedom (DOF). Therefore, this work focuses on an innovative graphic environment designed for teaching professionals which allows them studying D-H conventions, mapping both forward and joint workspaces, evaluating surface and volume trajectories applied to anthropomorphic robotic arms, and visual observation of the movement of serial robotic manipulators. This process is much simpler than the analytical interpretations of results.

Keywords: engineering education, kinematics, robotic arm, virtual instrument, trajectory planning

1. Introduction

Since practices with robotic arms are needed to understand fundamental theoretical concepts, students need to complete a full study on different kinematics-related aspects. The available bibliography includes numerous works on tools aimed at studying and simulating models for low-cost robotic arms, thus being highly appropriate for educational purposes [1]. For instance, ROBOTSIM was developed as a simulation tool for students working on robotic-manipulator programming based on animations and 3D graphs [2]. A computer-aided design package for multi-DOF manipulators allows computing mathematical equations to follow D-H kinematic convention through a GUI [3]. Regarding the study of robotic arms for industrial use (e.g. PUMA 560), work models can be defined by means of matrix equations through a robotics toolbox for MATLAB[®] [4]. ROBOLAB is a simulation toolbox based on this environment and aimed at solving students’ problems with robotic manipulators [5]. With this purpose, Cakir et al. developed an educational tool to solve kinematics for 6-DOF industrial robots [6]. Different methods have been studied so far for the solution of IK models for robotic arms. For instance, ROBOMOSP is a complete modelling and simulation environment for robotic manipulators through a 3D multiplatform CAD system. Its main innovation is that this package, specifically developed to train researchers, can be connected to remote robotic arms [7].

Obtaining the point cloud corresponding to the workspace of a given manipulator helps us to evaluate its physical limitations. For instance, the workspace of robotic arm PUMA 560 is described on the basis of discriminant polynomials (see Zhang et al. [8]). In this sense, Wang et al. [9] study workspace density in a hyper-redundant manipulator through an algorithm which takes ~40 s to solve workspaces. Thus, workspace boundaries are analysed by Abdel-Malek et al. [10] through the estimation of workspace volume.

The study of joint workspace in kinematics enables us to assure a continuous range of joint values when the end-effector is intended to reach a particular position [11]. According to the points in a trajectory, Lee-ElMaraghy [12] present a simulator to calculate the IK model of 5-DOF robotic manipulators. Based on this, Qin et al. [13] study IK workspaces applied to PUMA 560.

Different methods have been proposed for the implementation of trajectories applied to robotic arms. Tools based on the generation of linear trajectories between two given points through homogeneous transformation matrices are used in most studies (for instance, see references [4] and [14]). A trajectory planner through polynomial equations is also used for industrial applications [15]. Trajectory algorithms are applied to rigid robots with singularity restrictions (Robot A255) [16] and a 3D-trajectory pattern based on the IK model of 3-DOF manipulator arms is also specified [17]. Improving the ability of simulation and interactivity

with the user are educational features sought by the tool RobotiCad over the previous ones [18]. As main features to note, RobotiCad implements twelve types of different paths to simulate objects in the workspace of a robotic manipulator and interact with them.

The ability to test different kinematic techniques and complex algorithms, as mentioned above, on anthropomorphic robotic arms is possible through the use of educational simulation tools [19]. Thus, a tool integrated in a virtual laboratory that is being applied to robotics teaching is presented as a new learning experience [20]. In this sense, a remote and virtual laboratory that allows students to interact with real and simulated robots with 5-DOF is presented [21]. From the standpoint of impact on teaching and learning, a robot toolbox for kinematics simulation is also being applied with educational purposes [22]. The tool provides interactive real-time simulation and visualization of the industrial robot PUMA 560. Finally, an analysis about the advantages and disadvantages of using virtual environments and remote laboratories to teach practical courses in Robotics is discussed [23].

The present paper is organized as follows. The General Overview section presents and describes the main components of the developed simulation tool. The next section, Educational Tool for Kinematics, presents the graphic user interfaces which develop the equations based on the D-H method for the FK model and the geometrical method for the IK model. The subsections 3D Forward Workspace and 3D Joint Workspace describe the developed algorithms and simulation tools which allow calculating both forward and joint workspaces for different robotic arm configurations. The subsections Surface Trajectory and Volume Trajectory allow planning surface and volume trajectories through analytical geometry. The Experimentation section puts into practice the methodology developed so far, in which robotic arm Lynx6 is used as a platform. The section Pedagogical Assessment discusses the impact of the simulation tool on teaching and learning in Robotics. Finally, this paper contributes some conclusions, provides new features under current development and offers the developed set of tools to the scientific community for free download.

2. General overview

3D-RAS comprises functional subsystems which represent particular aspects of the kinematics of anthropomorphic arms. The simulation tool was programmed as a virtual instrument (VI) in LabVIEWTM and includes the following applications:

- Robotic Arm Simulator (FK & IK);
- 2D Joint Workspace;
- 3D FK Workspace;
- 3D IK Workspace;
- Surface Trajectory;
- Volume Trajectory.

Since this simulation tool is based on modules, the present methodology allows studying them and comparing the geometry of anthropomorphic arms through different subsystems (as long as their physical parameters and mechanical characteristics are known beforehand and regardless of their physical availability in our lab). Forward and inverse kinematics, together with forward and inverse workspace, helps students to understand kinematics-related aspects while using arm observations to debug discrepancies. Its different features—interactivity, robot type, input via programming or user commands, workspace typology, dynamic capabilities or remote control—are compared to some previously available tools to prove the novelty and possibilities contributed by this new simulation tool, as Table 1 shows.

3. Educational tool for kinematics

The solution to the problem of finding the FK model for a robotic manipulator is based on the matrix method proposed by D-H [24]. Equation (1) is the resulting transformation matrix which relates the reference system of segment $i-1$ to the reference system of the i -th segment of the arm:

$${}^{i-1}\mathbf{A}_i = \text{Rotz}(\theta_i) \cdot \mathbf{T}(0, 0, d_i) \cdot \mathbf{T}(a_i, 0, 0) \cdot \text{Rotx}(\alpha_i) \quad (1)$$

where ($i = 1, \dots, 6$); $\text{Rotz}(\theta_i)$ stands for rotation around axis Z_{i-1} with angle θ_i ; $\mathbf{T}(0, 0, d_i)$ is a translation of distance d_i along axis Z_{i-1} ; $\mathbf{T}(a_i, 0, 0)$ is a translation of distance a_i along axis X_i ; $\text{Rotx}(a_i)$ is a rotation around axis X_i with angle a_i ; and θ_i , d_i , a_i and α_i are the known values of joints and length in each coordinate system to be expressed in terms of the D-H convention.

Equation (2) shows the relation between transformation matrices, which forms the robotic arm's kinematic chain of joints and segments:

$$\mathbf{T} = {}^1\mathbf{A}_6 = {}^1\mathbf{A}_2 \cdot {}^2\mathbf{A}_3 \cdot {}^3\mathbf{A}_4 \cdot {}^4\mathbf{A}_5 \cdot {}^5\mathbf{A}_6 \quad (2)$$

where \mathbf{T} stands for the homogeneous transformation matrix we are looking for. As a practical case of the FK models developed and studied through 3D-RAS (see Fig. 1), Figures 2a–d show different simple 3D representations of 3-5 DOF serial manipulators. The results obtained exclusively depend on the robotic arms' geometrical characteristics—namely,

Table 1. Main features of some educational tools for teaching anthropomorphic robotic arms

Features	ROBOTSIM	Robotics Toolbox	ROBOLAB	Cakir et al.	ROBOMOSP	ROBOSIM	RobotiCad	3D-RAS
Language	C	MATLAB [®]	MATLAB [®]	MATLAB [®]	Open-Source	-	MATLAB [®]	LabVIEW [™]
Visualizing	3D-CAD	3D-Wire	3D-Solid	3D-Wire	3D-Solid	3D-CAD	3D-CAD	3D-Wire
Interactive	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Input	Commands	Commands	Parameters	Parameters	Commands	Commands	Commands	Parameters
FK & IK Simulator	Rhino XR-3	PUMA 560	16 Industrial Robots	Industrial Robots	Mitsubishi PA-10, PUMA	Microbot TeachMover	Cartesian and Serial Robots	Robots up to 5-DOF
DOF	5	6	6	6	7	5	8	5
Forward Workspace	No	No	Yes	No	No	Yes	No	Yes
2D Joint Workspace	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
3D Joint Workspace	No	No	Yes	No	No	No	No	Yes
Analytical Trajectories	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Dynamics	No	Yes	Yes	No	Yes	Yes	Yes	No
Real Remote Operation	Yes	No	No	No	Yes	No	No	Yes

physical parameters and joint values (θ_i , d_i , a_i and α_i).

The IK model allows arm positioning once specific coordinates (p_x, p_y, p_z) have been introduced for its end effector. A closed and feasible solution can be found for the inverse problem using a geometrical method which relates end-effector coordinates, joint angles and the robotic arm's physical parameters. The method is particularly feasible when only the four first degrees of freedom which define the position of the gripper are considered in order to minimize multiple solutions.

Equations (3–9) report a list of equations which allow determining the end effector's position and θ_i orientation angles in relation to the ground:

$$\theta_1 = \text{atan2}(p_y, p_x) \quad (3)$$

$$\theta_2 = \text{atan2}((x_b - (q \times z_b)), (z_b + (q \times x_b))) \quad (4)$$

$$\theta_3 = \theta_2 - \text{atan2}((x_b + (q \times z_b)), (z_b - (q \times x_b))) \quad (5)$$

$$\theta_4 = \theta_6 - \text{atan2}((x_b + (q \times z_b)), (z_b - (q \times x_b))) \quad (6)$$

where $\theta_1, \theta_2, \theta_3$ and θ_4 are the joint angles formed by the arm's kinematic chain; θ_5 stands for the values introduced for gripper opening/closure, and θ_6

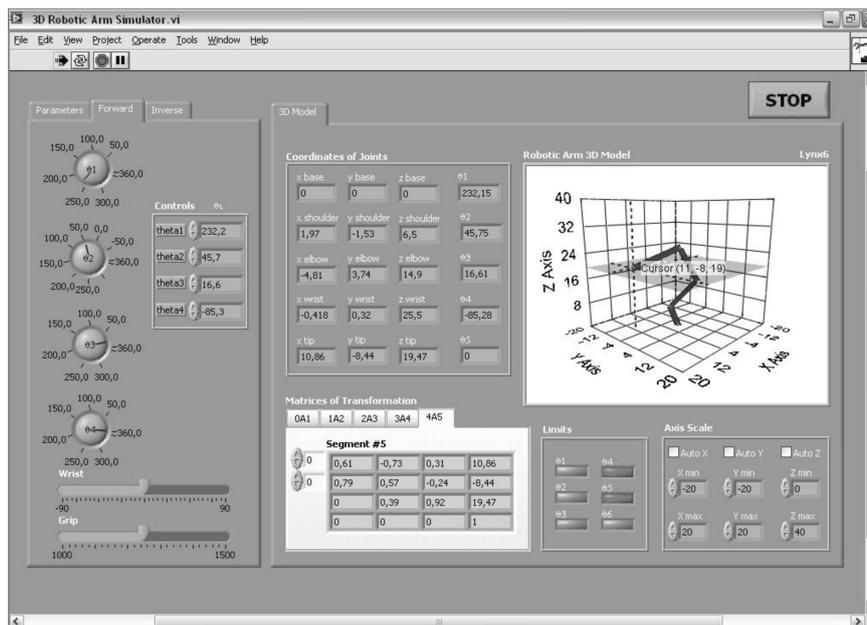


Fig. 1. GUI of Robotic Arm Simulator subsystem with 5-DOF arm Lynx6.

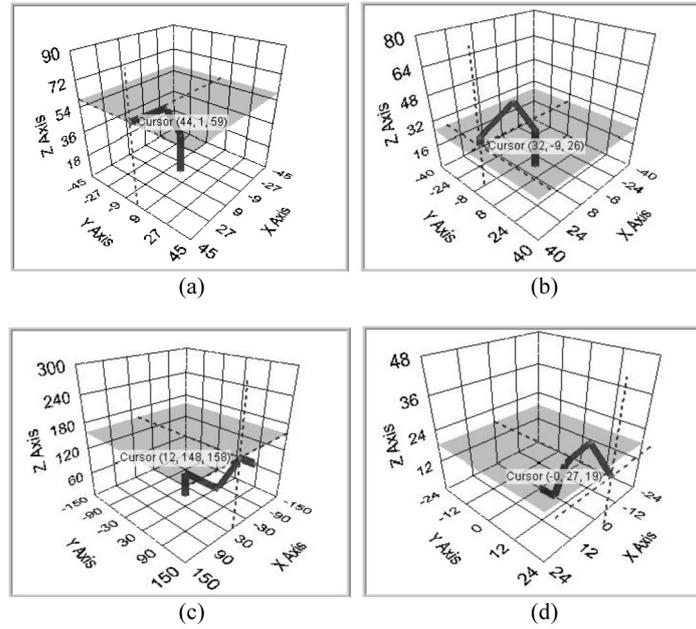


Fig. 2. Several models of anthropomorphic robotic arms with different DOF: (a) 3-axis for bionic arm, (b) 4-axis for A255, (c) 5-axis for IRB2400, (d) the first 5-axis for Pioneer 2.

stands for the wrist angle in relation to the ground (*wartg*):

$$x_b = \frac{(p_x - a_1 - (a_4 \times \cos\theta_6))}{2 \times a_2} \quad (7)$$

$$z_b = \frac{(p_z - d_1 - (a_4 \times \sin\theta_6))}{2 \times a_2} \quad (8)$$

$$q = \sqrt{\frac{1}{((x_b^2) + (z_b^2) - 1)}} \quad (9)$$

where x_b and z_b stand for the coordinates in axes X and Z at the end of the forearm; d_1 , a_1 , a_2 and a_4 stand for segment lengths, and q is a relation to facilitate compactness and equation solving.

The inverse tabs in Fig. 1 and 3 represent the IK and 2D Joint Workspace subsystems, respectively, which allow studying the IK model of anthropomorphic arms when the values of the end effector's coordinates (p_x , p_y , p_z) change. Figure 3 shows that kinematic motion can be studied for each joint separately or combined for all joints in the same chart.

4. 3D forward workspace

Obtaining a mathematical equation to calculate workspace in anthropomorphic arms is not an easy task. For this reason, the present paper launches a tool for the calculation and 3D computational representation of the forward workspace of

serial robotic arms by means of a point cloud. The study of forward workspace is focused on determining the reachable point cloud surrounding the studied robotic arm through the FK model. The algorithm of the developed tool creates an ActiveX container of parametric surfaces and configures the properties of the graphic window where results are displayed. Its graphic interface enables the user to configure D–H parameters and introduce the segment lengths of the robotic arm under study.

The implemented FK model performs a multiplication of transformation matrices through the kinematic chain defined in Equation (2). Thus, the developed algorithm calculates the set of coordinates (p_{xi} , p_{yi} , p_{zi}) for each segment of the robotic arm through Equations (10–12):

$$p_{xi} = \mathbf{T}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \quad (10)$$

$$p_{yi} = \mathbf{T}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \quad (11)$$

$$p_{zi} = \mathbf{T}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \quad (12)$$

As input values to the algorithm, the user must define a range of joint values for θ_1 , θ_2 , θ_3 and θ_4 to simulate robotic arm orientation in different positions:

$$\theta_1 = \theta_{1a_1}; \quad 0 \leq \theta_{1a_1} \leq 180^\circ \quad (13)$$

$$\theta_2 = \theta_{2a_2}; \quad 0 \leq \theta_{2a_2} \leq 90^\circ \quad (14)$$

$$\theta_3 = \theta_{3a_3}; \quad 0 \leq \theta_{3a_3} \leq 90^\circ \quad (15)$$

$$\theta_4 = \theta_{4a_4}; \quad -90 \leq \theta_{4a_4} \leq 90^\circ \quad (16)$$

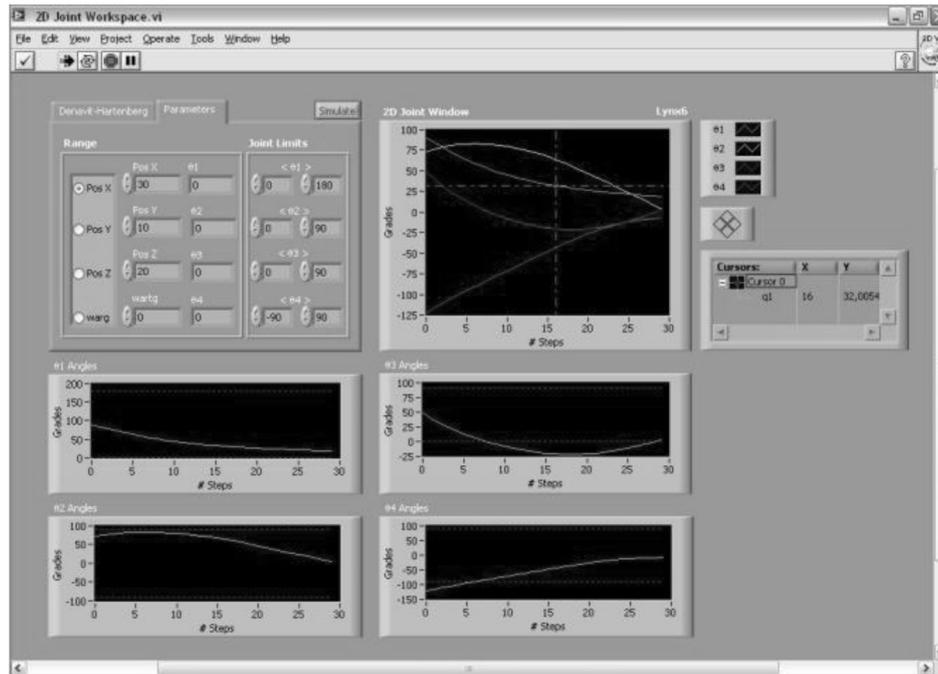


Fig. 3. GUI of 2D Joint Workspace subsystem. Charts represent joint values combined or separately.

the robotic arm's workspace linked to its kinematic chain. These series of coordinates depending on the FK model are calculated iteratively. The points of each robotic joint are passed from each iteration to the next and—once all points have been obtained—they are restructured and ordered for the same segment (namely base, shoulder, elbow, wrist and gripper) to facilitate point-cloud visualization, as points are grouped by colours into a 3D graph to perform the forward workspace.

In some graphical applications, robotic arm workspace is tested through manual variation of joint values for θ_i angles. As a practical example of the algorithm implemented in this subsection, workspace is computationally generated for different 3-5 DOF anthropomorphic arms (Fig. 4a–d).

The graphic user interface (see Fig. 5) is divided into three tabs: control panel of *D-H parameters*, simulation controls and time graphs. In the upper left-hand part, θ_i controls take the joint's range of continuous values defined by the user in Equations (13–16). The different arm's segments are identified by colours in its central part—base (orange), shoulder (yellow), elbow (red), wrist (magenta) and gripper (purple)—to determine the coordinate point-cloud (p_{xi} , p_{yi} , p_{zi}) to be drawn in the workspace. In its right-hand part, depending on the ongoing simulation, the tool allows varying the Scale to reduce the number of values for θ_i and represents the number of points composing the 3D map according to Equations (17–18):

Each value in joint angles leads to a new state of

$$Scale = \frac{\theta_{i1}}{f}; 1 \leq f \leq 90 \quad (17)$$

$$Number\ of\ points = \frac{\theta_1 \times \theta_2 \times \theta_3 \times \theta_4}{f^i}; 1 \leq i \leq 4 \quad (18)$$

where f is the user-defined *Factor* in the GUI and i is the number of considered joints. *Points* indicates the number of points calculated for each joint, while *Elapsed Time* displays the seconds elapsed in the representation of each point cloud. In the lower right-hand part, auto-scale can be selected depending on the simulated robotic manipulator. In the lower left-hand tab, *D-H parameters* specify the robotic arm's segment lengths, although these values can be loaded from a library or modified by the user when working with different robotic-arm lengths.

For instance, the panel in Fig. 5 shows the point cloud for the Lynx6 arm formed by the revolution of joint angles with the following values: $\theta_1 = 180$, $\theta_2 = 90$, $\theta_3 = 90$, $\theta_4 = 180$ and *Factor* = 15.

5. 3D joint workspace

This section puts forward the development of a tool which allows obtaining the range of joint values of anthropomorphic robotic arms and determining whether their physical limits are exceeded. The user provides the simulation tool with all the D–H parameters which define the robotic arm's physical model and the simulation's functioning parameters.

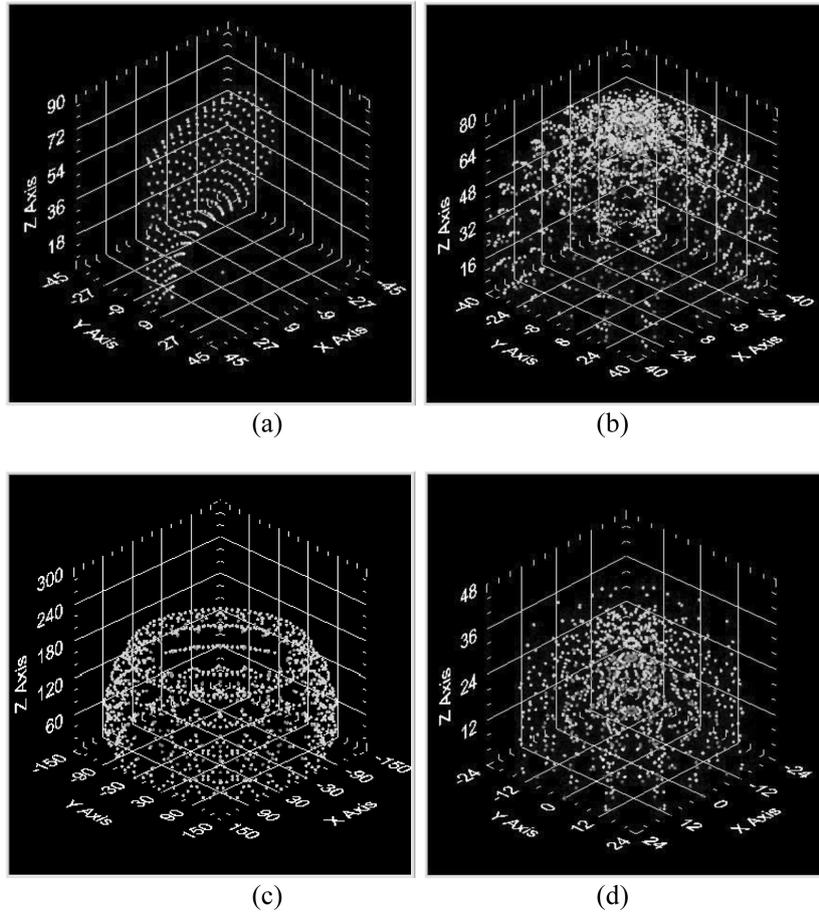


Fig. 4. Set of graphs to compare forward workspace of anthropomorphic robotic arms with different number of DOF: (a) 3-Axis for bionic arm, (b) 4-Axis for A-255, (c) 5-Axis for IRB2400, and (d) the first 5-Axis for Pioneer 2. Workspace has been performed for base (orange), shoulder (yellow), elbow (red), wrist (magenta) and gripper (purple).

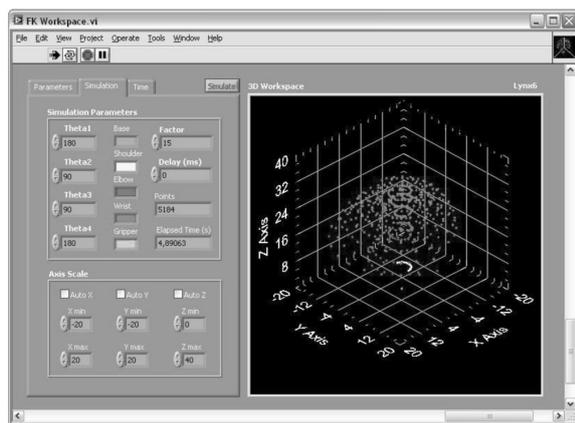


Fig. 5. GUI of 3D FK Workspace subsystem performing simulation.

The algorithm combines the range of values (p_x, p_y, p_z) and calculates all the possible joint values ($\theta_1, \theta_2, \theta_3, \theta_4$ and θ_5) which can be adopted by the manipulator. Once ordered, the array of joint values obtained by the IK model allows joint workspace visualization and study by means of independent

graphs. During joint angle calculation (θ_i), the algorithm calculates the robotic arm's FK model and obtains the coordinates (p_{xi}, p_{yi}, p_{zi}) for each arm segment.

To avoid erroneous angle values, joint variables are tested to prevent them from exceeding the threshold values. Likewise, the IK model guarantees that joint movements do not lead the robotic arm into a position exceeding its maximum physical length. Computationally, the physical limit of the robotic arm can be calculated through Equations (19–20):

$$\overrightarrow{[(S5), (S1)]_{x,z}} = [x_5 - x_1, z_5 - z_1] \quad (19)$$

where $\overrightarrow{[(S5), (S1)]_{x,z}}$ is the vector joining the base's reference system to the arm's end effector in plane XZ.

$$\overrightarrow{[(S5), (S1)]_{x,z}} \leq \sqrt{(d_1 + a_2 + a_3 + a_4)^2 + a_1^2} \quad (20)$$

where (a_1, d_1, a_2, a_3, a_4) stand for the segments' maximum length from the arm's base to its end effector.

The graphic user interface is divided into two main sections: application controls and joint-workspace graphs. The length of any robotic arm with up to 5 DOF can be configured through its D–H parameters. By default, it includes the parameters of the arm studied in the present work (see the “Experimentation” section). Through its GUI, the range of values followed by the end effector (p_x, p_y, p_z) can be defined by the user. Graphs are useful to study joint workspace ($\theta_1, \theta_2, \theta_3$ and θ_4) and prove that the values adopted by joint angles meet joint restrictions and conditions. In this user environment, indicators q_1, q_2, q_3 and q_4 are activated when the joint angles reach out of the range values. On the other hand, the performance step enables simulation speed variation of the simple model into a 3D graph. It helps to study the robotic arm’s slowed-down movements and the positions adopted by its segments.

6. Surface trajectories

The objective of the present subsection is describing a simulation module intended to study different surface equations through analytic geometry, which allows applying open curvilinear trajectories to anthropomorphic robotic arms with up to 5 DOF.

The user provides the algorithm of this module with the robotic arm’s D–H and operating parameters for simulation. Its formula menu enables the user to experiment with different equations as well as to compare and study aspects related to analytic

geometry. The model for a given surface is generically defined by the following explicit equation:

$$y_i = f_i(x_1, x_2) \quad (21)$$

where $f_i: \mathbb{R}^2 \rightarrow \mathbb{R}$, $x_1, x_2 \in [a, b]$ and $[a, b] \in \mathbb{R}$.

Programmatically, Function (21) renders an array of defined values (x_1, x_2) between the initial and final points in the arm’s real workspace $[a, b]$. The number of intermediate points (i) calculated by this function depends on a digital control (points) located at the GUI, which sets the number of curves composing the 3D surface. The user can select each of these curves individually in the 3D graph by means of a cursor (index row). The set of values (p_x, p_y, p_z) shaping the curvilinear trajectory followed by the robotic arm through its IK model is actually defined when one of the possible curves is extracted. The manipulator’s movement is considered constant along the curve when planning and applying 3D trajectories.

The GUI control panel (Fig. 6) is divided into two main sections, which are further divided into two subsections. All these subsections can be accessed through the view tabs: configuration of simulation and *D-H parameters* (a), and path planning and 3D modelling (b). The upper tab in (a) allows the user to observe the points (p_x, p_y, p_z) in the robotic arm’s trajectory as well as to control simulation steps. The running speed can be set according to research needs, thus facilitating the understanding of the manipulator’s movements along its curvilinear tra-

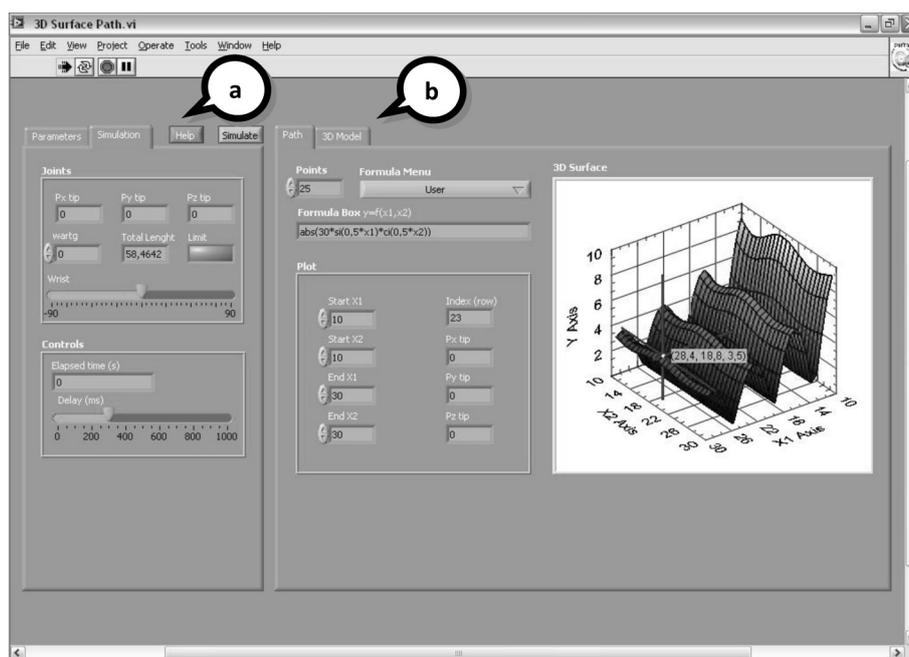


Fig. 6. GUI of surface trajectory subsystem performing a simulation.

jectory. The resulting array of data can be extracted and exported into a file for real robotic-arm control. The lower tab allows the user to configure D–H parameters and introduce the segment lengths of the studied robotic arm.

The upper tab in (b) displays the controls necessary for surface modelling through explicit formulae (Equation 21). These controls enable the user to select different predefined surfaces as well as to edit new formulae. The lower tab displays the evolution of the simple model of the robotic arm in 3D perspective along user-defined movement sequences. This tab also enables the user to study all the values of homogeneous transformation matrices for each robotic-arm segment.

7. Volume trajectories

This section explains the development of another 3D-RAS subsystem by means of a geometrical method of volume parametric equations. This system allows generating close curvilinear trajectories to study movement in anthropomorphic robotic arms with up to 5 DOF.

The algorithm implemented for this simulation module is an adaptation of the algorithm described in the previous subsection for volume parametric equations. Generically-defined, explicit equations such as (21) have the limitation of needing the curve to be a function of (x_1, x_2) in \mathbf{y} . That is, all values (x_1, x_2) have one only corresponding value in \mathbf{y} . Not all curves fulfil this condition, so—in order to be able to work with them as if they were functions—a domain (the manipulator’s workspace) must be chosen and (x_i, y_i, z_i) must be considered as dependant variables. Thus, the 3D-space parametric representation of any given object requires defining the three functions in Equations (22–24):

$$x_i = g_i(u, v) \quad (22)$$

$$y_i = h_i(u, v) \quad (23)$$

$$z_i = j_i(u, v) \quad (24)$$

where $u, v \in [a, b]$, $[a, b] \in \mathfrak{R}$ and $g_i, h_i, j_i : \mathfrak{R}^2 \rightarrow \mathfrak{R}$.

Functions (22–24) render an array of values (x_i, y_i, z_i) which represent the i -th coordinate of the point generated when independent variables (u, v) are assigned values within the interval $[a, b]$.

Like the module structure described in Fig. 6, the control panel is divided into two main sections. The left-hand panel (a) contains the controls and indicators which interact with the robotic arm’s IK model. The lower tab allows the user to configure the *D–H parameters*, segment length and DOF for any robotic manipulator.

The right-hand panel (b) contains 3D-trajectory

modelling (lower tab) and the robotic arm’s 3D modelling (upper tab). The lower tab displays the parameters which define the trajectory and volume functions applied to the manipulator.

The number of intermediate points (i) calculated by parametric equations is specified by the user by means of a digital control (*points*) at the GUI which defines 3D-object surface resolution. Once volume has been studied, the next step allows the user selecting a close curve, thus obtaining the values followed by the end effector and applying this trajectory by collecting the points (x_i, y_i, z_i) which compose it.

The upper right-hand tab displays a 3D perspective of the simulated robotic arm. The homogeneous transformation matrices and coordinates of each segment (p_{xi}, p_{yi}, p_{zi}) can be studied in the lower central area.

8. Experimentation

Lynx6 is used here for practical demonstration of the results obtained with the developed methodology. This robotic arm allows quick, accurate and low-size movements, thus being highly recommendable for educational practices. Lynx6 is a 5-DOF manipulator: its first three degrees of freedom correspond to its base, shoulder and elbow, respectively, while the remaining ones correspond to wrist movement and rotation. Although it does not affect end effector positioning or orientation, its functional clip acts as an additional DOF.

The time the software takes for forward workspace mapping is an appropriate measure of computational cost in terms of effort. Note that calculation of all possible coordinates (p_{xi}, p_{yi}, p_{zi}) for each robotic arm segment demands high computational effort due to the huge amount of processed data, as calculation complexity increases with the arm’s number of degrees of freedom. Completion and processing time necessary to calculate the coordinates (p_{xi}, p_{yi}, p_{zi}) within every loop in the algorithm depends on the scale *Factor* chosen. Equation (18) allows analysing the effect of the number of DOF (18), showing that, for equal DOF complexity, the range of angles (θ_i) can differ so as to provide different number of points. Therefore, the number of points calculated is inversely exponential to the *Factor*.

As a demonstrative example of the previously-presented “Volume Trajectory” subsystem, Fig. 7a–d shows the results obtained through the study of different parametric equations. Thus, to obtain a close trajectory through a hyperbolic paraboloid composed by $i=25$ curves (see Fig. 7a).

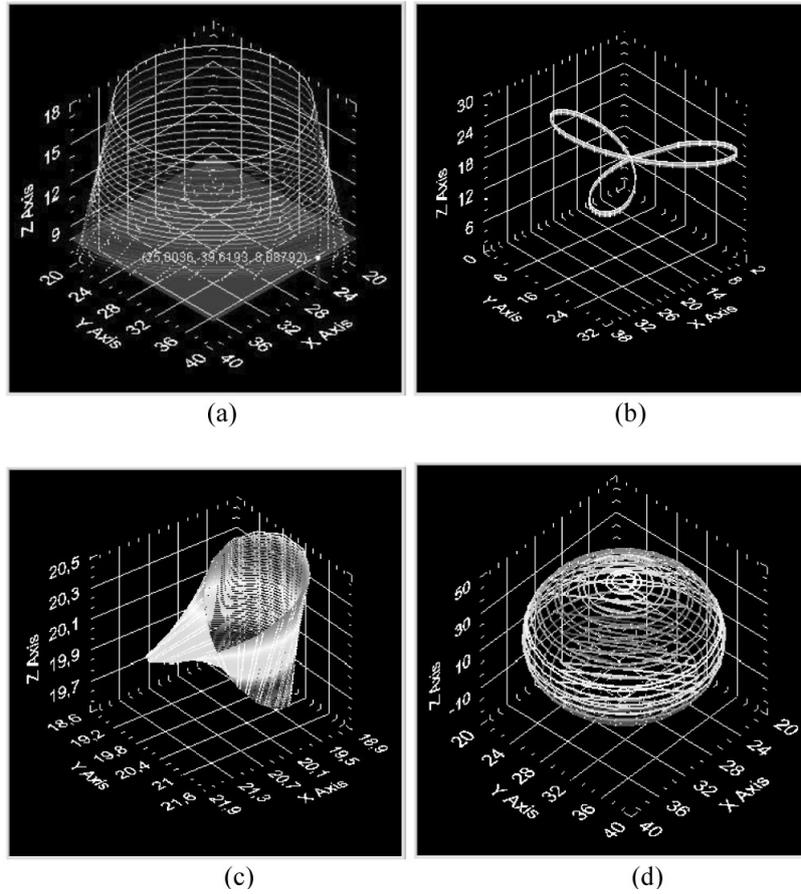


Fig. 7. Examples of planned close curvilinear trajectories: (a) hyperbolic paraboloid, (b) epicycloid, (c) Möbius belt, and (d) latitudinal sphere.

Equations (25–27) must be defined as follows:

$$x_i = 30 + 10 \cdot \sin u \quad (25)$$

$$y_i = 30 + 10 \cdot \cos(u + v) \quad (26)$$

$$z_i = 18 \cdot \exp(-v) \quad (27)$$

where $u \in [0, 6]$ and $v \in [0, 1]$. Equations (28–30) define a close trajectory obtained from an epicycloid composed by $i=70$ curves (see Fig. 7b):

$$x_i = 20 + 20 \cdot \sin u \cdot \cos u \quad (28)$$

$$y_i = 20 + 20 \cdot \sin 3u \cdot \sin u \quad (29)$$

$$z_i = 20 + v, \quad (30)$$

where $u \in [0, 30]$ and $v \in [0, 0]$. To obtain a close trajectory belonging to a Möbius belt (Fig. 7c), Equations (31–33) must be defined as follows:

$$x_i = 20 + (5 + 0,5v \cdot \cos(0,5u)) \cdot \cos u \quad (31)$$

$$x_i = 20 + (5 + 0,5v \cdot \cos(0,5u)) \cdot \sin u \quad (32)$$

$$z_i = 20 + 0,5v \cdot \sin(0,5u), \quad (33)$$

where $u \in [0, 360]$, $v \in [0, 360]$ and $i = 28$. If a latitudinal sphere is required instead of the previous

one (see Fig. 7d), Equations (34–36) allow generating the following curvilinear trajectory:

$$x_i = 20 + 10 \cdot \cos u \cdot \cos v \quad (34)$$

$$y_i = 20 + 10 \cdot \sin u \cdot \cos v \quad (35)$$

$$z_i = 15 + 15 \cdot \sin v \quad (36)$$

where $u \in [0, 360]$, $v \in [0, 360]$ and $i = 28$. For correct application of the path defined by parametric equations to the robotic arm's kinematic model, the range of independent variables (u, v) must fall within the manipulator's real workspace. With this purpose, it should be borne in mind that each different point in the curve within the interval $[a, b]$ must fulfil the following conditions: $u, v \in [a, b]$ and $[a, b] \in \mathfrak{R}$. In order to obtain a close trajectory, the coordinates of the points obtained when $(u, v) = a$ must coincide at least in one of its limits when performing $(u, v) = b$.

Since lines represent the intersection of two planes, the analytic description of curves defines the intersection of real objects. As a practical example, a welding cord in the union of two pipes (Fig. 8) is one of the possible applications of the Volume Trajectory subsystem. The orthogonal cylinders

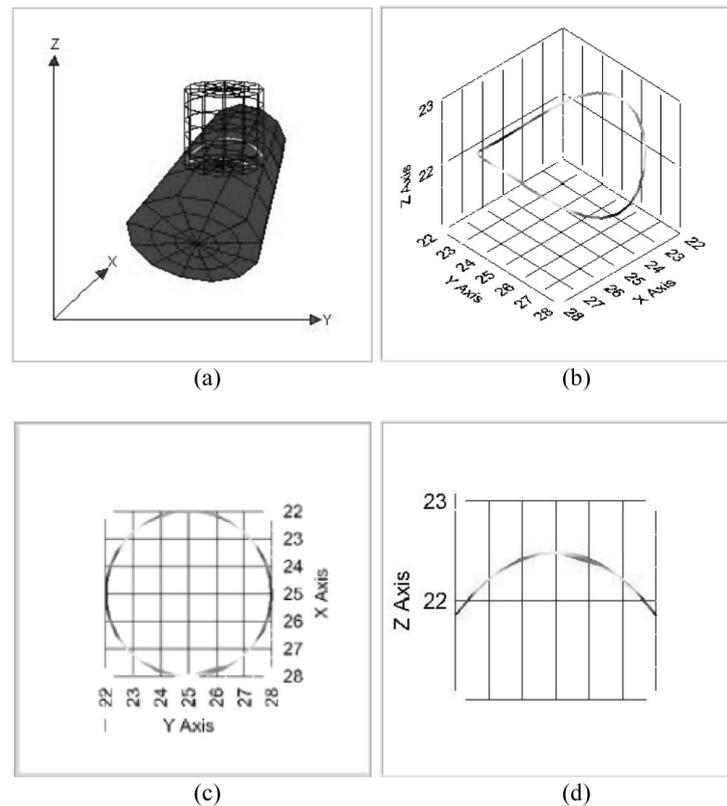


Fig. 8. (a) Intersection of two orthogonal cylinders, (b) trajectory obtained with Equations (37–39), (c) zenithal view of the curve, and (d) transversal perspective.

with radii $r_1=8$ and $r_2=9$ cm, respectively, produce an intersection whose function can be expressed with Equations (37–39). To produce the via-points used to guide the robotic arm, the cylinders must belong to the manipulator's workspace. For this example, their location must be defined with a relative displacement of (25,25,15) as it can be observed in the following equations:

$$x_i = d_x + r_1 \cdot \cos u \quad (37)$$

$$y_i = d_y + r_1 \cdot \sin u \quad (38)$$

$$z_i = d_z + \sqrt{r_2^2 - r_1^2 \cdot \sin^2 u} \quad (39)$$

where d_x , d_y and d_z are the offsets on axes (X,Y,Z), respectively, and $i = 32$.

9. Pedagogical assessment

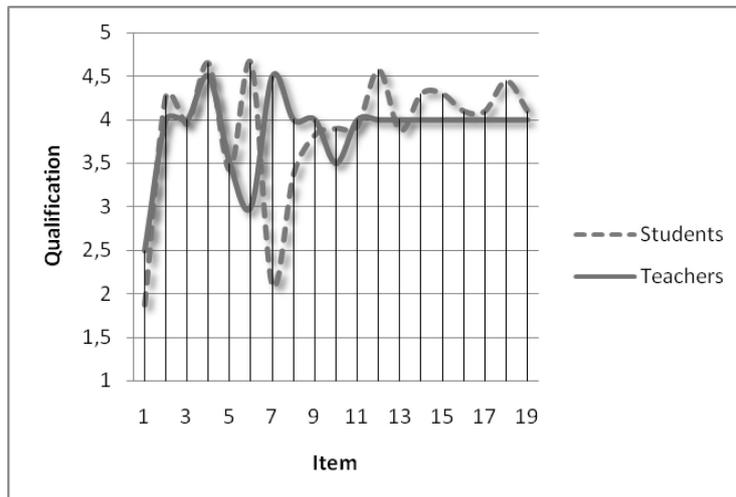
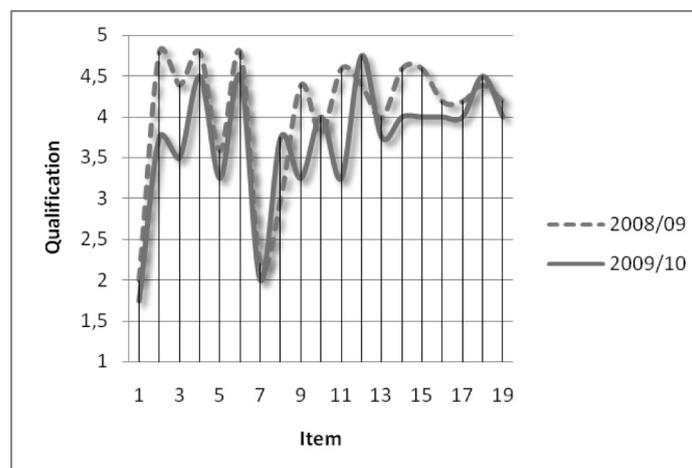
Robotics, whose teaching innovation is the driving force of this work, is an elective subject in the university degrees Electronic Engineering and Computer Engineering at the University of Huelva (Spain). With the aim of presenting the contributions and proving the capabilities of this educational simulation tool, Table 2 presents a statistical study

completed in 2008/09 and 2009/10 with a score ranging between 1 (very low valuation) and 5 (very high valuation). This assessment includes aspects referred to the tool's features and its acceptance and use. Two groups of users (38 students of this subject and 14 professionals from this university Department) were considered. Their initial knowledge level on Robotics was low (question 1). Fig. 9 shows that the students' opinion was similar (57.9%) to that of teaching professionals (42.1%). Regarding the teaching and learning of Robotics, the high scores given by the users prove that the use of the simulation tool reinforces and favours the acquisition of these concepts (questions 2–7). In this sense, students declare they disagree with the idea that theoretical concepts are to be learned by studying, without the need of computer simulations. Regarding the user graphic interface, its ease of use and interactivity stand out remarkably. Besides, students stress the way theoretical and practical concepts are related to one another, thus allowing quick learning (questions 8–12). Clear preference for the modules composing the 3D-RAS tool is also shown (questions 13–18), as the users' degree of satisfaction is rather high.

Figure 10 shows a comparison of the students' assessment along two academic years. In general,

Table 2. Evaluation questionnaire of the 3D-RAS simulation tool

Item	Requested Features and Capabilities	Teachers	Students	Deviation
1	Previous level on Robotics	2.5	1.88	3.83
2	The use of a graphic tool fosters the students' motivation and interest	4	4.28	2.28
3	The software application has a clear and intuitive structure	4	3.95	3.35
4	Putting the application into practice is feasible in the university context	4.5	4.65	3.19
5	The application allows learning new theoretical concepts	3.5	3.43	3.03
6	The application allows consolidating previous theoretical concepts	3	4.65	3.19
7	Theoretical concepts are learned through study, without need of computer simulations	4.5	2.1	1.79
8	Interface appearance	4	3.38	2.28
9	Usability	4	3.83	1.92
10	Graphic simulation and results	3.5	3.9	3.35
11	Theoretical-practical understanding	4	3.93	2.05
12	Application interactivity	4	4.58	3.03
13	2D Joint Workspace subsystem	4	3.88	3.35
14	3D FK Workspace subsystem	4	4.3	3.49
15	3D IK Workspace subsystem	4	4.3	3.49
16	Surface Trajectory subsystem	4	4.1	3.35
17	Volume Trajectory subsystem	4	4.1	3.35
18	Robotic Arm Simulator (FK & IK) subsystem	4	4.45	3.19
19	General assessment	4	4.1	4.38

**Fig. 9.** Average score of students and teachers about 3D-RAS.**Fig. 10.** Comparison of the students' opinion along two academic years.

the opinion of the two groups of students can be said to follow the same tendency, showing slight variations. It is still too early to obtain concluding results, although the developed analysis leads us to conclude that the generic capability of the tool allows graphical simulation of the movement of anthropomorphic robotic arms and helps studying kinematic models. In this sense, the general assessment was rather high (question 19), since practices were more pleasant and appealing. Regarding the place where this tool was used, 56.52% of users opted for completing practices in lab, counting on the teaching support of professionals and help of their classmates. However, a large number of students (43.47%) used the tool at home. The reason behind this decision is that they did not feel conditioned by the need to dispose of specific equipment to complete these practices in the classroom.

10. Conclusions

The main contribution of the present work is the proposal of a new educational tool (3D Robotic Arm Simulator) aimed at studying and evaluating the kinematics of anthropomorphic robotic arms. With this purpose, the present work develops the different subsystems which shape the 3D-RAS tool. This simulation tool is based on the D–H method; thus, the followed methodology is put forward in the present paper as well as proven capable of simulating different robot arm configurations. The developed GUIs—given their high versatility and visual nature—are highly intuitive and allow studying forward and joint workspaces in serial robotic arms with up to 5 DOF. Besides, a simulation module of trajectory planning has been developed. The capabilities of this module allow generating surface and volume trajectories applied to serial robotic arms. With the aim of teleoperating a robotic arm with educational purposes in lab practices, a remote control module was also developed for one of the arms available in our lab: Lynx6.

The download of a set of demonstrative videos and an installable version of this educational tool (including all its subsystems, which allow running simulations without connecting the tools to remote experiments) is available at <http://www.uhu.es/tomas.mateo/investigacion/3DRAS.htm> (3D Robotic Arm Simulator file). This tool is available as GNU license and the source code can be requested from the authors for educational purposes only. Both comments and critics are welcome.

Regarding future developments, which are to be presented in forthcoming papers, our current research activity is aimed at integrating new analysis tools and validating the proposed algorithms by

means of one of the manipulators available in our lab. This allows studying robotic arms according to different kinematic models (Quaternions, Jacobian, etc.) In this sense, the use of numeric analysis tools and other analytical solutions with feasible algorithmic implementation may improve the quality of the results. For instance, the workspace of a manipulator can also be obtained by considering its volume limits. The dynamic features of robotic manipulators shall also be considered with the aim of improving trajectory planning (interpolation, speed, etc.) Finally, among the tool's new features, GUI is currently being improved by means of a view allowing 3D CAD representation of simulated arms.

References

1. T. J. Mateo Sanguino and J. M. Andújar Márquez, Simulation Tool for Teaching and Learning 3D Kinematics Workspaces of Serial Robotic Arms with up to 5-DOF, *Comput. App. Eng. Educ.*, 2010, DOI: 10.1002/cae.20433.
2. J. Sheehan and A. M. Eydgahi, Computer Animated Simulation of Robotic Manipulators and Workcell design, in *Proc. 33rd Midwest Symposium on Circuits and Systems*, 2, 1990, pp. 1175–1178.
3. J. F. Nethery and M. W. Spong, Robotica: a Mathematica Package for Robot Analysis, *IEEE Robotics and Automation Magn.*, 1(1), 1994, pp. 13–20.
4. P. I. Corke, A Robotics Toolbox for Matlab, *IEEE Robotics and Automation Magn.*, 3, 1996, pp. 24–32. Available http://www.petercorke.com/Robotics_Toolbox.html
5. S. Kucuk and Z. Bingul, An Off-Line Robot Simulation Toolbox, *Comput. App. Eng. Educ.*, 2009, online.
6. M. Cakir and E. Butun, An Educational Tool for 6-DOF Industrial Robots with Quaternion Algebra, *Comput. App. Eng. Educ.*, 15(2), 2007, pp. 143–154.
7. A. Jaramillo-Botero, A. Matta-Gómez, J. F. Correa-Cacedo and W. Pera-Castro, ROBOMOSP: ROBOTics MOdelling and Simulation Platform, *IEEE Robotics and Automation Magn.*, 33(4), 2006, pp. 62–73.
8. H. Zhang, Efficient Evaluation of the Feasibility of Robot Displacement Trajectories, *IEEE Trans. Systems, Man and Cybernetics*, 23(1), 1993, pp. 324–330.
9. Y. Wang and G. S. Chirikjian, Workspace Generation of Hyper-Redundant Manipulators as a Diffusion Process on SE(N), *IEEE Trans. Robotics and Automation*, 20(3), 2004, pp. 399–408.
10. K. Abdel-Malek, H. J. Yeh and S. Othman, Interior and Exterior Boundaries to the Workspace of Mechanical Manipulators, *Robotics and Computer-Integrated Manufacturing*, 16, 2000, pp. 365–376.
11. J. Lenarčič and A. Umek, Simple Model of Human Arm Reachable Workspace, *IEEE Trans. Systems, Man and Cybernetics*, 24(8), 1994, pp. 1239–1246.
12. D. M. A. Lee and W. H. ElMaraghy, ROBOSIM: a CAD-based Off-line Programming and Analysis System for Robotic Manipulator, *Computer-Aided Engineering Journal*, 1990, pp. 141–148.
13. C. Qin and M. A. Carrera-Perpiñán, Trajectory Inverse Kinematics by Non Linear, Non Gaussian Tracking, *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2008, pp. 2057–2060.
14. J.I. Maza Alcañiz and A. Ollero Baturone, A Matlab-Simulink toolbox for robotics, *1st Workshop on Robotics, Education and Training*, Weingarten (Germany), 2001, pp. 43–50.
15. S. Macfarlane and E.A. Croft, Jerk-bounded manipulator trajectory planning: design for real-time applications, *IEEE Trans. Robotics and Automation*, 19(1), 2003, pp. 42–52.
16. D. Braganza, W. E. Dixon, D. M. Dawson, and B. Xian,

- Tracking control for robot manipulators with kinematic and dynamic uncertainty, *Int. J. Robotics and Automation*, **23**(8), 2008, pp. 117–126.
17. B. Fardanesh and J. Rastegar, A new model-based tracking controller for robot manipulators using trajectory pattern inverse dynamics, *IEEE Trans. Robotics and Automation*, **8**(2), 1992, pp. 279–285.
 18. R. Falconi and C. Melchiorri, RobotiCad: an Educational Tool for Robotics, in *Proc. 17th World Congress of the IFAC*, Seoul (Korea), 2008, pp. 9111–9116.
 19. R. Šafaric, M. Truntic, D. Hercog and G. Pacnik, Control and Robotics Remote Laboratory for Engineering Education, *Int. J. Online Engineering*, **1**(1), 2005, pp. 1–8.
 20. A. R. Sartorius Castellanos, L. Hernández Santana, E. Rubio and I. Santana Ching, Virtual and Remote Laboratory for Robot Manipulator Control Study, *Int. J. Eng. Ed.*, **22**(4), 2006, pp. 702–710.
 21. J.M. Andújar Márquez and T.J. Mateo Sanguino, Design of Virtual and/or Remote Laboratories. A Practical Case, *Revista Iberoamericana de Automática e Informática Industrial*, **7**(1), 2010, pp. 64–72.
 22. M. Toz and S. Kucuk, Dynamics Simulation Toolbox for Industrial Robot Manipulators, *Comput. App. Eng. Educ.*, 2008, online.
 23. F. Torres, F. A. Candelas, S. T. Puente, J. Pomares, P. Gil and F. G. Ortiz, Experiences with Virtual Environment and Remote Laboratory for Teaching and Learning Robotics at the University of Alicante, *Int. J. Eng. Ed.* **22**(4), 2006, pp. 766–776.
 24. J. Denavit and R. S. Hartenberg, A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices, *J. Applied Mechanics*, 1955, pp. 215–221.

Tomás de J. Mateo Sanguino is an Industrial Engineer, Electronic Engineer and Master in University Teaching. From 1998 to 2004, he was granted a scholarship at the National Institute of Aerospace Technology (INTA) and worked as a hired engineer at the Spanish National Research Council (CSIC). Since 2004, he has worked as a full-time lecturer with tenured position (not a civil servant) in the Dep. Electronic Engineering, Computer Systems and Automatics at the University of Huelva (Spain). Besides, he currently works as an instructor in the CCNA program at the CISCO Networking Academy. He earned his Ph.D. in Electronic Engineering in 2010, which deals with design and construction of virtual/remote laboratories in Robotics.

José Manuel Andújar Márquez is an Industrial Engineer, Bachelor of Physical Sciences and Engineer and Doctor. He is Full Professor in Systems and Control Engineering and Head of the Department of “Electronic Engineering, Computer Systems and Automatics” and Director of the Research Group “Control and Robotics”, both at the University of Huelva. His research lines focus on Control Engineering, Intelligent-control Systems, Non-linear Control, Modelling and Simulation, Artificial Intelligence applied to Engineering, Design, Monitoring and Control of Renewable-energy Systems, Systems for Signal Acquisition and Distribution, and Education within Engineering.