A simulator for assembly practices of robotic entities based on plug and play building blocks

L A Portilla-Meneses¹, E G Puerto-Cuadros¹, and F H Vera-Rivera¹,²,³

¹ Grupo de Investigación en Inteligencia Artificial, Universidad Francisco de Paula
Santander, San José de Cúcuta, Colombia
² Grupo de Estudios Doctorales en Informática GEDI, Universidad del Valle, Santiago
de Cali, Colombia
³ Materials Science and Technology Research Group, Foundation of Researchers in
Science and Technology of Materials, Colombia

E-mail: eduardpuerto@ufps.edu.co, fredyhumbertovera@ufps.edu.co

Abstract. This paper describes a simulator of physical robotic entities. A robotic entity simulator includes new functionalities in the simulation: flexibility by allowing the creation of many types of robots such as arms, legs and hands. Multiplatform and multi-user by allowing multiple users to enter from multiple systems. Scalability to be developed as a web application being this type of technology which is currently at the forefront and therefore the possibility of developing new modules and continuous improvements. Three test cases were developed including the design of a hand, leg and a model to represent movement through servos in a real model of a planar type pantograph robot. This simulator can be used as a pedagogical and innovative strategy for training and learning in the field of robotics. Useful for teaching and laboratory practices of research in artificial intelligence and related in an individual, group and online way.

1. Introduction

In the context of engineering, especially in the last two decades, the implementation of modeling and simulation has become important as an indispensable and transversal tool to solve scientific and technological problems raised from systems engineering, civil, chemical, industrial, biomedical, mechanics, education and others. The simulation allows us two fundamental tasks, the virtual representation of the robotic entity and the transfer of commands to the physical entity so apart from simulating it will also allow us to perform control actions without the need for intermediate software. The importance of simulations is that developers can detect errors and correct them before the robot manufacturing process. Also, users can use the robotic entities to previously verify if the orders that will be sent will have the desired result. Knowing (simulating) in advance the actions that the robot will perform, not only increases the efficiency of the process, cost reduction, ubiquity, among other aspects, but also helps to protect the safety of users and the robot itself.

A robotic entity is the creation and virtual reproduction of a physical robot. A robotic entity can therefore be concrete, abstract, particular or universal. Robotic entities are not only physical robots such as chairs or people, but also virtual entities or software programs. Robotics constitutes knowledge and doing about robots, this implies the use of knowledge of various areas for the design, construction, assembly and commissioning of a robot for a specific purpose [1]. In this paper we introduce a “simulador de entidades robóticas (SER)”; it was developed in order to teach the concepts of robotics.

Published under licence by IOP Publishing Ltd
In the field of training in the area of robotics, new paradigms based on the construction of robotic entities have been developed in order to make students interested in working in robotics, such as the culture change project in robotics by Carnegie Mellon (CCRC), a project that develops robotic education lessons that involve students in computational thinking practices (CTP) and then measures their effectiveness in teaching concepts traditionally taught in computer science and other science, technology, engineering and mathematics (STEM) classrooms [2]. On the other hand, in [3, 4, 5] a simulator that allows you to simulate certain robot arms that are already previously designed within the system and that cannot be modified is proposed. The difference with our simulator is that SER allows design and simulation of almost any system. Another difference is that the simulator does not allow a collaborative environment. Work [6] proposed an open source Java based 3D robot visualization environment called SIMBAD. It does not support any physics calculations, only simple collision detection with objects placed in a flat world. The goal of this simulator is to provide a simple environment to test robot controllers and AI algorithms. In [7] a simulator of 4 degrees of freedom based on virtual reality is proposed. A new concept of robot, which stands for selective compliance assembly robot arm (SCARA). This simulation model only has a robot already pre-designed so the SER to design several types of robots stands out with respect to this project. Microsoft Robotics Studio [8] provides a “networkable”, service-based architecture framework for developing real robots. The package includes a simulation run-time that is similar to a game engine in terms of physics and visualization capabilities. Robotics Studio uses the PhysX physics engine, which is one of the highest fidelity engines available to date.

Other projects of interest have to do with the creation and/or simulation of robots of human parts. The author of [9] propose to learn a robot controller in simulation, with the potential of then transferring this to a real robot. Uses 3D simulations to train a 7-DOF robotic arm in a control task without any prior knowledge. The controller accepts images of the environment as its only input, and outputs motor actions for the task of locating and grasping a cube, over a range of initial configurations. In [10] a simple and effective method that combines 3D printing and digitization techniques to create a 3D printable cable-driven robotic hand from scanning a physical hand is proposed. The method involves segmenting the 3D scanned hand model, adding joints, and converting it into a 3D printable model.

At the level of software applied to robotics, the technological advances of recent years such as web applications, virtual reality and 3D simulation frameworks such as WEBGL and THREEJS, are allowing both physical and software agents to robotic entities. For example, in the entertainment industry, applications such as fourth-generation simulators have been designed at Universal Studios where you can see how immersive virtual reality manages to cause real sensations in a virtual world in viewers.

This paper is organized as follows. Section 2 presents the methods used in this research work, section 3 presents the characterization 3. Architecture of SER, then in section 4 the case study is presented, finally in section 5 we summarize the conclusion of this work.

2. Methods

In order to reach this objective, a preliminary analysis of the system and a design of the modules necessary to develop the simulator were carried out, as well as a basic communication interface with the physical models. The Figure 1 shows the research model carried out in this work.

![Figure 1. Research model.](image-url)

2.1. Phase 1. Analysis

The activities related to the analysis of requirements for the development of SER were executed, which consisted of analyzing each of the elements that make up the system, taking into account that the primary objective is to achieve a very detailed understanding of the system and its operation model. The
requirements for software development were compiled, examined and verified. In addition, there was an activity to examine simulator restrictions.

2.2. Phase 2. Design
Include an entry and registration module, part modeling module that allows the user to model parts, designer module that allows joining various pre-designed pieces, maintenance module that will allow maintenance and updating of the parts or the models and a 3D simulator applying the WEBGL specifications [11,12]. The user entry and registration module, the part modeler, the assembler module, the simulator module and the maintainer.

2.3. Phase 3. Communication model
Implement a basic communications model that allows the simulator to be linked to electronic elements, allowing the possibility of creating the protocol within said module, handling the concept of data stream. The implementation of the communications model allows the transmission of appropriate data between the simulator and the physical model or with projects developed by students from other areas such as electronic engineering.

2.4. Phase 4. Case study, evaluation and conclusions
In the testing phase, four cases were studied to apply within the simulator and where the proper functioning can be verified and where it was verified that the simulator meets the specifications set at the beginning of the project. This phase is complemented by the realization of a model and its respective communication with the simulator. Two case studies demonstrate the utility that this will provide to different users of the system in this way it will also be verified that the simulator responds and correctly performs the tasks indicated in the specifications.

3. Architecture
A methodology called IDEF [13] was used, which allows us to show the interaction between each of the modules defined for SER. In addition, IDEF allows us to visualize how they interact with the controls and methods that influence the system. Although several levels of detail of the SER systems were defined with IDEF, we will only show the second level called IDEF 0, where the interaction between the three main modules is broken down in addition to the interaction of the external elements with the simulator (see Figure 2).

![Figure 2. Robotic entities simulator model.](image)

The modeling of the pieces to be used in the simulations within the system is carried out considering that it is a plug and play system and therefore it is only drag and drop. The assembly of these pieces is done by joining links as if it were a chain and being a piece, this chain must be closed at the start point
with the end point. Being a piece that will be joined with other pieces, it must have at least two points of union, one its father and the other, its son, therefore said that each of these points can be treated in two different systems depending on the needs of the robotic entity, movements can be made in XY axes or XYZ axes.

The assembly process is carried out by joining the pieces that are designed in the modeler. The pieces are basically joined when two joints of the same type overlap, when this intersection is achieved a small black circle appears indicating that the two pieces are already unit at that point so that pieces can be joined until the system is completed, which We will call Entity from now on. When several pieces are already joined, you can get very personalized results and with different variations. Always keep in mind that to be able to simulate an assembly there must be a base piece. The base piece is the one that within the system is the only one that does not move by default. It is the point of support of the whole entity and that will remain fixed in time and is established simply by right clicking on said piece and selecting the option.

The simulation can store profiles that, in addition to storing selected positions, can be used to give continuous movement over time or a Timeline. When using the timelines an additional value is given to the simulator that will allow users to analyze the possible variations, errors, automation and better configurations that only when observing a repetitive timeline could be revealed to the end user. The simulation can be done without connecting the model or connecting it, so that if you want the movements to be physically represented, only the model you represent will be selected.

On the other hand, SER uses a series of functions that allow the entire execution of the system, a core that for each module as a library of well-defined functions logically called Lienzo.js. This file is basically a repository that is accessed by the entire module and that manages through its functions everything that the actor requests.

4. Simulations and results
The scenarios used for each of the test cases were selected considering the degree of complexity of each of them starting from least to greatest.

4.1. Scenario 1: 4G robotic arm
The objective of this test is the creation of a robotic arm with 4 degrees of freedom including clamp. First, the parts that compose it in the modeling module will be designed, then assembled by joining them in the assembly module and as a last step it will be simulated in 3D. Figure 3 shows two pieces created in the modeling module, these being simple structures that can be linked to others in the assembly module. The created parts can then be assembled and joined with others in the assembly module through the green dots or joints, see Figure 4. After the assembly process, the 3D simulation is carried out which allows to see the movements that are requested by the end user, Figure 5.
4.2. Scenario 2: Human leg
The objective of this test is the creation of a human leg with its corresponding degrees of freedom and foot. First, the parts that compose it in the modeling module will be designed, then assembled by joining them in the assembly module and as a last step it will be simulated in 3D.

Figure 6 shows two pieces created in the modeling module, these being simple structures that can be linked to others in the assembly module. The created parts can then be assembled and joined with others in the assembly module through the green dots or joints, Figure 7. After the assembly process, the 3D simulation is carried out which allows to see the movements that are requested by the end user (see Figure 8).

![Figure 6. Parts design. Creation of a robotic arm with 4 degrees of freedom. Initial parts modeling.](image6)

![Figure 7. Assembly of the parts. Creation of a robotic arm with 4 degrees of freedom. Parts assembly.](image7)

![Figure 8. 3D simulation. Creation of a robotic arm with 4 degrees of freedom. 3D simulation.](image8)

Figure 9 shows the motion control panel for the 3D model, specifically for a robotic arm. Figure 10 shows the control panel for a robotic leg; in this panel you can manipulate the physical characteristics associated with each of the components of the virtual model, so that this movement can be transmitted to the model. Other experiments were carried out in the construction of a robotic entity of a human hand. The creation of a planar robotic arm with 3 degrees of freedom with physical model, Figure 11, for representation and communication with SER.

![Figure 9. Control panel for robotic arm.](image9)

![Figure 10. Control panel for robotic leg.](image10)

4.3. Communications model and physical prototype
To carry out the process of communicating the simulator with the physical model, a character string was created that implements a structure [variable: value], with which a very simple data stream was formed that could be interpreted by all internet of things (IoT) devices or plates of development as Arduino. In this way, the only thing the model will do will be to receive the orders that the simulator delivers leaving the processing and other functions on the side of SER, creating a great advantage for the development of new modules that allow more sophisticated movements or more complex tasks without having to modify the model. The motherboard will be coded according to the type of model that is developed and
the language that supports the motherboard, for example, in Arduino it would be programmed in C language as in WEMOS. In this way the development of the simulator of the physical model becomes independent creating an additional added value of flexibility in the whole system, see Figure 12.

![Figure 11. Virtual model of physical model of Figure 4.](image1.png)

![Figure 12. Robot planar model.](image2.png)

It was designed on a 50cm x 50cm base using 3 SG90 servomotors, an Arduino mega module and an ESP8266 WIFI communications module which functions as a link between the model and the simulator. The advantage of simulator communication is that it delivers three fundamental data for each of the joints, these data are position, initial position and speed. Due to this structure it is possible to obtain great versatility and in any of the models that could be created and where much of the movement processing is left to the base plate on which you want to develop. Any prototype can be created from articulated arms, cars, legs and some more complex ones such as hexapods and amphibians. Also, the materials that you want to use range from cardboard, acrylic and plastic, so a wide range of applicability is achieved with this project.

5. Conclusions
A simulator was created that meets all the expectations of what a simulator needs, covering the entire process of developing the parts through its assembly and finally observing through the use of a graphic engine such as its behavior virtually through the module of simulation. Useful for use in collaborative learning environments. Unlike the previous simulators, SER includes new functionalities in the simulation of robotic entities. Among which you have: Flexibility by allowing the creation of many types of robots such as arms, legs and hands. Multiplatform and multi-user by allowing multiple users to enter from multiple systems. Learning modules in locomotion can be easily developed with the reuse of functions and information that the simulator already has.

Due to its plug-and-play design of assembly, assembly and simulation it is possible to simulate different robots, from industrial robotic arms to anthropomorphic hands, allowing to cover many projects due to the high flexibility in the design of various entities. The scalability comes given by the technology used being avant-garde in the current development of applications and combining the languages and a database structured in the framework of the model entity relationship which allows future improvements without becoming a highly entropic system.

The simulator design applied the plug and play or plug and use concept, where what is expected is to quickly implement any project that you want to take to the physical development phase. An arm or hand inside the simulator would have minimal complexity and where with few hours of training, almost any user could manipulate the simulator. Users who will use the simulator range from external students to systems engineering or related students, it is also aimed at university professors who require that their classes have a development capable of visualizing before what is required to create. Work was done considering that this research development is aimed at research seedbeds or research groups related to robotics. This project is of a research type and responds to the need to have a tool for students that allows them to perform assembly practices, a quantitative method is also generated that in turn generates a model which allows to perform various works in the area of robotics and inside the simulator.
References

[1] Lopez-Ramirez P A and Andrade Sosa H 2013 Aprendizaje de y con robótica, algunas experiencias Revista Educación 37(1) 43

[2] Carnegie Mellon Robotics Academy 2020 Use the motivational effects of robotics to excite students about STEM (Pittsburgh: Carnegie Mellon University). Consulted on: https://www.cmu.edu/roboticsacademy/roboticscurriculum/smart-getting-started.html

[3] Villoslada Peciña A, González-Quijano Alvarez J and Abderrahim Fichouche M 2010 Diseño y aplicación de un actuador SMA en el control de manos robóticas (Madrid, España: Universidad Carlos III)

[4] Xie Z, Berseth G, Clary P, Hurst J and Van de Panne M 2018 Feedback control for cassie with deep reinforcement learning IEEE/RSJ International Conference on Intelligent Robots and Systems IROS (Madrid: IEEE) p 1241

[5] Apgar T, Clary P, Green K, Fern A and Hurst J 2018 Fast online trajectory optimization for the bipedal robot cassie Robotics: Science and Systems, Online Proceedings ed H Kress-Gazit, S Srinivasa, T Howard and N Atanasov (Pittsburgh: Robotics Science and Systems Foundation) Consulted on: http://www.roboticsproceedings.org/rss14/index.html

[6] Hugues L and Bredeche N 2006 Simbad: An autonomous robot simulation package for education and research International Conference on Simulation of Adaptive Behavior, from Animals to Agents 9. SAB 2006. Lecture Notes in Computer Science 4095 (Berlin, Alemania: Springer) p 831

[7] Bussola R, Legnani G, Callegari M, Palmieri G and Palpacelli M C 2019 Simulation assessment of the performance of a redundant SCARA Robotics 8(2) 45

[8] Jackson J 2007 Microsoft robotics studio: A technical introduction IEEE Rob. Autom. Mag. 14(4) 82

[9] James S and Johns E 2016 3D simulation for robot arm control with deep q-learning NIPS 2016 Workshop: Deep Learning for Action and Interaction (Barcelona: Conference on Neural Information Processing Systems) Consulted on: https://arxiv.org/abs/1609.03759

[10] Tian L, Thalmann N M, Thalmann D and Zheng J 2018 A methodology to model and simulate customized realistic anthropomorphic robotic hands CGI 2018: Proceedings of Computer Graphics International 2018 (Bintan Island Indonesia) (New York: Association for Computing Machinery) p 153

[11] Mendoza-Escobar E, Lopez-Martinez I, Herrera-Aguilar I, Lopez-Maldonado U J, Velázquez-Cruz A D and Ixmatlahu-Diaz S D 2014 Modelado de prototipos robóticos para robótica en aplicaciones web utilizando X3D/WebGL Revista Programación Matemática y Software 6(1) 39

[12] Toasa R M, Egas P F B, Saltos M AG, Perreño M A and Quevedo W X 2019 Performance evaluation of WebGL and WebVR Apps.VR environments Advances in Visual Computing. Lecture Notes in Computer Science vol 11845, ed G Bebis, et al. (Switzerland: Springer, Cham.) p 564

[13] Marca D A and McGowan C L 2006 IDEF0 and SADT: A Modeler’s Guide (Auburndale: OpenProcess, Inc.)