Gutiérrez Estrada, Citlali; Díaz Zagal, Sergio; Cruz Reyes, Rafael; Rivera Alemán, Juan José; Moreno, Rosa Elvira; Sabel Hernández, Bany; Rodríguez Guido, Diana; Lemus González, Rafael

Systems Engineering as a critical tool in mobile robot simulation


International Journal of Combinatorial Optimization Problems and Informatics
Morelos, México

Available in: http://www.redalyc.org/articulo.oa?id=265225625004
Systems Engineering as a critical tool in mobile robot simulation

Citlalih Gutiérrez Estrada, Sergio Díaz Zagal, Rafael Cruz Reyes, Juan José Rivera Alemán, Rosa Elvira Moreno, Bany Sabel Hernández, Diana Rodríguez Guido, Rafael Lemus González

Instituto Tecnológico de Toluca, Ingeniería de Software, Av. Tecnológico s/n, Fraccionamiento la Virgen, Metepec Edo. de México. C.P. 52149
citlalihg@ittoluca.edu.mx, sdiaz@ittoluca.edu.mx, rcruzrey@gmail.com

Abstract. This paper explains a mobile robot simulation process in a static scenario with obstacles, which generated a location map supported by various prototypes created throughout the project, used to analyze the behavior of autonomous navigation algorithms and implemented mapping.

This project had the management and formalization support from Software Systems Engineering through the spiral lifecycle model, used to develop and test the intended processes and human reasoning. Each step of the spiral lifecycle model iteration considered the goals, artificial intelligence (AI) and simulation aspects by identifying the robot’s purpose, as well as solution alternatives to define the methods to achieve the intended goals.

As a project with a common objective for a multidisciplinary research group, the work and results obtained by means of systems engineering allowed tackling the interests of several end users and handling functional and non-functional requirements separately.

Keywords: Systems Engineering, Simulation, Spiral Model, Mobile Robot.

1 Introduction

In today’s world, Software Development has become increasingly important and, therefore, Software Systems Engineering is constantly looking to integrate activities, techniques, methodologies and models to its analysis, design and development processes in order to create state-of-the-art products [1, 2, 3].

The concept of “Software Engineering” was used for the first time after the NATO conference of 1968 [4], and the development of programs, still conceived at the time as a hardware complement, started to be taken seriously. In that sense, some of the issues of interest of Software Systems Engineering were the definition of requirements, the organization of a development team and the generation of product documentation, and it is necessary to apply a series of rules, use a common language and share a working discipline that results in the knowledge of each stage of a project, the real status of the conception process.

In another field, the construction of mobile robots follows a process that allows for the assessment of the robot’s future applications, the visualization of the prototype’s design, the programming task and the testing for implementation. According to this process, a robot is designed, built and implemented efficiently, considering the environment for which it was designed. Many times, however, the construction of mobile robots faces problems during the implementation stage, unpredictable during the design or programming stages, resulting in faulty prototypes that must be redesigned and corrected in their construction or programming, delaying delivery times or excessively increasing material and construction costs due to damage to electronic components, and where proper documentation was never generated.

A viable solution for this problem is to simulate the robot in an environment that closely resembles the one in which it will be normally operating, following a well-defined methodological process, so to test for design or programming failures before testing the actual robot, documenting and reusing the information generated. Another advantage of simulation is model testing
in different scenarios, with conditions subject to the normal rules of physics (such as gravity), resulting in more precise models and testing marks considerably higher than those obtained by actual prototypes.

The present paper details the design of a virtual model that simulates the behavior of a robot, specifically using the Spiral Lifecycle Model of Systems Engineering, and considering aspects of risk management, compliance of requirements, robot programming and versions control.

The main goal was to design a mobile robot prototype in a static scenario similar to another physical commercial or previously built prototype. Virtual scenarios and simulation of the robot’s motion routines were carried out using sensors in virtual environments for performance evaluation when implementing a navigation algorithm and a mapping algorithm in the programming stage.

Simulation allows observing the robot’s virtual performance in different operation maneuvers, a subject of multiple factors, making it possible to efficiently and effectively analyze the robots behavior in different circumstances; the newly generated requirements and demand of resources were tackled at that moment, so the best results were obtained before the design stage.

One of the main contributions of this work was the training required to use simulation software and the integration of techniques usually applied only in Software Engineering projects, formalizing work in stages as delicate as requirements gathering and interpretation, programming modeling and efficient control of code versions.

Another contribution of this project was the increased interest within the technological community to collaborate in the development of the mobile robot as an integral part of the actual Computer Systems Engineering and Electronics Ph.D. programs offered at the Instituto Tecnológico de Toluca [Technological Institute of Toluca], actively using simulation methods and techniques to build didactic robots.

This document is divided into six sections that summarize the research carried out with the methodology proposed for the project’s development. The structure is as follows:

- **Section 2.** Details the concepts that constitute the basis of this research.
- **Section 3.** Details existing work, similar to the one of this project.
- **Section 4.** Contains the procedure to obtain the results proposed and expected in the draft proposal. This section details the method used to integrate Software Engineering, Robotics and Simulation techniques to generate a document of requirements specifications, a formal simulation model and its implementation in a mobile robot prototype.
- **Section 5.** Details the results obtained after simulation, using easily understandable formats, as well as statistics that measure optimization levels of the prototype.
- **Section 6.** After results are interpreted, this section offers the project’s final conclusions and emphasizes the importance of simulation previous to the construction of mobile robots. It also includes possible improvements to the project’s development process and research lines to be considered in future works.

## 2 Theoretical Foundations

The following items detail several basic theoretical notions used for our research.

### 2.1 Software Systems Engineering and description

The term *Software Engineering* was coined in 1968 as a response to the bleak progress attained so far to develop quality software in time and at adequate costs. Software developers were not able to define concrete objectives, predict the required resources to accomplish such goals or satisfy clients’ expectations [5].

Software Engineering deals with the development of software systems using Systems Engineering mechanisms (problem definition, analysis, planning processes, control processes and product evaluation) that, following a sequence of activities and decisions, transform an operational need into a software system description and its corresponding documentation [6]. In general, it defines tasks like [7]:

- Transformation of predicates, programming calculations, types of abstract data and notations of axioms.
- Implementation of methodologies such as modular design, information withholding, data flow and abstraction layers.
- Methods to automatize program developments, like text editors, syntax editors, etc.
Finally, Software Engineering allows automatizing processes by applying a model within a specific organization, company, institution, etc. Each of these lifecycle models for systems development provides a framework in stages. The application of a specific model depends on the policies, guidelines and quality standards of the organization, company or institution, but the general goal is quality assurance in systems development [8].

### 2.2 Lifecycle models of software systems

Robot designing requires the precise knowledge of certain parameters to obtain a specific behavior from the mechanical structure, the transmission, the drive system (actuators), the sensory system (sensors), the terminal elements, the control system (controllers) and input. Although the calculation of these parameters may use a set of equations and algorithms, other issues must be considered before building the robot, such as its application, architecture, functionality, etc., which can be accomplished through a lifecycle model methodology useful for product development.

Table 1 contains a compilation of lifecycle models used in Systems Development [9,10]; however, this work used only the spiral model due to its characteristics and for being a risk-driven process model used in concurring software engineering intensive systems with multiple, simultaneous users [11].

<table>
<thead>
<tr>
<th>Model</th>
<th>Phases</th>
<th>Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterfall model</td>
<td>1. Analysis and definition of</td>
<td>Development with phase sequences and well-defined goals; the next phase doesn’t start until the previous phase is finished.</td>
</tr>
<tr>
<td></td>
<td>requirements.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. System design.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Units implementation and testing.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. System integration and testing.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. Operation and maintenance.</td>
<td></td>
</tr>
<tr>
<td>Evolutionary model</td>
<td>1. Analysis of requirements.</td>
<td>Develops initial implementation, subjecting it to user comments and refining it through different versions, until an adequate system is obtained.</td>
</tr>
<tr>
<td></td>
<td>2. Concurring activities (cyclic actions until an adequate system is defined).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Specification.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Development.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Validation.</td>
<td></td>
</tr>
<tr>
<td>Prototyping model</td>
<td>1. Analysis of objectives.</td>
<td>Develops initial implementation (based on objectives, not requirements), subjecting it to user comments and refining it through different versions, until an adequate system is obtained.</td>
</tr>
<tr>
<td></td>
<td>2. Quick plan. Quick modeling and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>design.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Prototype construction.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Development and delivery,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>feedback.</td>
<td></td>
</tr>
<tr>
<td>Spiral model</td>
<td>1. Evaluates alternatives, identifies and tackles risks.</td>
<td>Each turn of the spiral represents a phase of the process. Risk management-focused lifecycle model.</td>
</tr>
<tr>
<td></td>
<td>2. Develops and verifies the next product level.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Sets objectives, alternatives and restrictions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Plans the next phase.</td>
<td></td>
</tr>
</tbody>
</table>
2.3 Simulation

Lately, several simulation tools have become available thanks to new technological breakthroughs. Such software availability in the market has allowed for simulation processes to be widely used in systems analysis that utilize systems engineering, as pointed out by Thomas H. Taylor [11].

This project developed a virtual prototype from a real model that simulates the behavior of a mobile robot in a static scenario with obstacles, generating a map of the scenario for location purposes. The robot and scenario described in this paper were developed using the MariLou Robotics 2010 software [12] and the C/C++ programming language, which, combined, allow the user to carry out a set of simulations in an environment that represents the real environment where the mobile robot will travel; the autonomous navigation algorithm and mapping algorithm are assessed.

The methodology used to create and develop the simulation of the robot in this paper is based on that proposed by Azarang [13], which has the following phases:

1. **System definition.** This phase includes a description of the problem, goal identification, decision variables, definition of restrictions, uncontrollable variables and robot’s behavior.
2. **System analysis.** Description of logical interactions between decision variables, based on uncontrollable variables.
3. **Model formulation.** A logical-mathematical code was created to exactly define interactions between variables.
4. **Language selection.** This system used the C/C++ language, following the needs of the research team.
5. **Model coding.** The required commands were generated for the model to be run in the computer.
6. **Model validation.** The capacity of a model to represent reality was validated through a statistical comparison between the model results and real results.
7. **Experimentation.** The different alternatives to be evaluated were selected through input variables and different levels, so to optimize response variables of the real robot’s system.
8. **Implementation.** The best alternative was selected and implemented.
9. **Monitoring and control.** This phase was very useful, as it was necessary to modify the simulation model with the new changes to the real system, so that the robot model continued to be an exact representation of the system.

3 Similar works

There are several works related to robot design, modeling and construction. Considering the proposal of the work hereto described, we will highlight the following:

3.1 Design, modeling and construction of a mobile robot

The work [15] describes the design and construction of a mobile robot, manufactured for research purposes and to experiment control and AI algorithms; it also has applications with different sensors (ultrasound, infrared, magnetic, position, etc.). This project also integrates mechanical design engineering and modeling.

The project’s modeling phase uses CAD (AutoCAD software), which allows only for the study of the mobile robot’s mechanical design as a whole, but it does not allow the integration (in the design) of sensors or actuators, let alone observe its performance during simulation. However, contribution of this work is observed in the mathematical models used to program the robot’s performance.

3.2 Modeling of 3D environments to plan robot paths using stereoscopic vision and active control of optic parameters [16]

Modeling system of 3D environments for remote control tasks, such as the repair of live mid voltage distribution lines. It emphasizes 3D reconstruction of the surroundings by integrating stereo vision techniques and active control of the stereoscopic head parameters, with convergence and focus control, to obtain a solid system that generates visual maps that can be integrated in the robot-path planner, performing tasks in complex, unstructured surroundings.
The reconstruction result updates a database to implement crash control of robotic arms, plan paths for standard tasks and generate visual information for the operator, such to attain adequate remote presence for repair tasks.

This work designed the 3D surroundings where the robot’s navigation is partially simulated; however, realistic features are not applied to scenarios, but objects serving as obstacles are included, so the robot is able to gather data and send them to the control interface.

The main contribution of this work is the interface design, which can be used to modify simulation parameters. This was taken into consideration for our project’s proposal.

3.3 Tracking 3D paths using image-based visual control [17]

Description of a visual control system designed to track 3D paths from an object in motion, emphasizing the need to “predict” the motion of the object that is being tracked. Different components of the visual control system implementation are analyzed, using a camera placed at the robot’s end for visual feedback purposes. Using a previous formulation, this work shows the different results of the robot according to specific 3D paths.

This work’s simulation neither indicates if it is subject to real conditions, nor mentions the mounting of sensors or actuators in the proposed mobile robot model. Its contribution, however, focuses on the use of cameras within a 3D environment, mounted on the mobile robot for better visual control at the moment of simulation.

3.4 Design and simulation of robotic orthosis for walking rehabilitation program [18]

This work details the design of a robotic orthosis for walking rehabilitation, with the goal to optimize both the patient’s recovery and the therapist’s professional practice. The proposed robotic orthosis has six degrees of motion freedom and is used on the hips, knee, ankle and foot. It is paired with a system to support the patient’s weight and a rolling tape. The design considers only the sagittal plane components of biomechanical variables.

The work proposes a brace if height-adaptable carbon fiber segments, attached to the patient, made out of polypropylene, covered with low-density foam and including a Velcro strap. For actuators, the work proposes the use of German servomotors with Harmonic Drive reducers for each joint, considering it an alternative for a country-level industry. It also proposes a system to control position of joints. For design verification purposes, a complete model of the orthosis for simulation is detailed, which was used to evaluate the position of ankle and knee joints at the sagittal plane. Simulation demonstrates that the designed orthosis is capable of following a normal walking pattern, with errors tolerable for the application.

The work uses the same software of the mobile robot simulation project described in this article. It is, therefore, a valuable document for the references made to the application of the Marilou 2010 software in robotics simulation projects, and because it details how to integrate simulations with Matlab during testing of Matlab algorithms for robotics directly in Marilou 2010 scenarios.

4 Experimental procedures

This project was based on a formal simulation study and follow-up of its stages, integrating software engineering techniques such as the Spiral Model [6], with its first process focused on the definition and specification of objectives; for this, the first step is the analysis and documentation of prototype requirements, followed by the risk management phase, which includes risk analysis and model formulation, resulting in the construction of the robot’s first prototype; for this phase, the Marilou Robotics Studio 2010 software [12] was used. The next phase (development) used the support of a C/C++ language compiler (Visual C++.NET, specifically), which included programming and experimentation to assess the model’s performance, using Marilou 2010’s MODA (Marilou Open Devices Access) component. This module allows accessing robot-mounted devices like sensors, which are then used as variables in the C/C++ code.
The final part of this project was the evaluation of the mobile robot; this process included the interpretation and control of versions of decision-making processes to better manage the following spiral. Figure 1 offers an overview of the process described [9], [10].

![Diagram of a mobile robot simulation project](image)

**Fig. 1.** Diagram of a mobile robot simulation project.

The mobile robot’s simulation was planned as a series of experiments. The following paragraphs offer an overview of the four quadrants of the five evolution cycles of the robot prototype, using the spiral lifecycle model.

### 4.1 First iteration around the spiral

**Objectives:** To know the Marilou software [12] to simulate, model a first testing prototype, develop a preliminary algorithm of autonomous navigation, implement the mentioned algorithm and initiate development of a communication interface with Matlab.

**Quadrant 1. Definition and specification of requirements.** The following requirements were considered according to the objectives planned for this first spiral, for validation purposes:

- **Prototype of mobile robot.** Using the Marilou Robotics 2010 simulation software, a preliminary 3D geometrical model was developed, featuring an assembly of basic figures like rectangular prisms and spheres.
- **Modular programming.** Object-oriented programming was used. A navigation algorithm was developed and communication of commands was carried out using Classes and Methods. The C++ simulator-native programming language was used.
- **Connection with Matlab.** The first communication approach between the simulation software and Matlab was considered a sort of solution for future processing of data obtained by the robot’s sensors.
- **User interface.** For validation of results, work was done to obtain an interface capable of manipulating real-time simulation parameters.
- **2 ultrasonic sensors.** These sensors were used for the autonomous navigation algorithm to interact with data received, and for the mobile robot to follow a defined path and avoid obstacles.

**Quadrant 2. Risk analysis**

- **Changes in functional and non-functional requirements.** Most projects consider this type of risk due to the changing nature of
each prototype generated at each spiral, and this project was not the exception.

- **Lack of experience using Matlab.** At the beginning of the first spiral, knowledge of the mathematical software Matlab was considerably limited, delaying some of the activities and the timely accomplishment of proposed objectives.

- **Lack of experience using the simulator.** The first approach to the Marilou Robotics 2010 software occurred during the first spiral, after comparing and evaluating it against other robotics simulation software; therefore, the project’s progress at this phase was limited to moderate.

**Quadrant 3. Design, testing and implementation**

- **First approach to physical model.** Figure 1 depicts the first prototype developed with the Marilou Robotics 2010 simulation software, using basic geometric figures.

- **Simulation of a basic algorithm of autonomous navigation.** The simulation of this preliminary prototype required a scenario that allowed evaluating the navigation algorithm’s performance. This scenario is depicted in Figures 2 and 3.

![Fig. 2. Preliminary prototype.](image)

![Fig. 3. First testing scenario.](image)

**Quadrant 4. Evaluation**

- **The virtual model is not completely similar to the physical model.** Being the first prototype evaluated, resemblance with a real mobile robot model was discarded, as the purpose of this first stage was to evaluate the navigation algorithm.

- **Partial connection with Matlab.** Due to lack of knowledge of the Matlab mathematical software, connection with the simulator was only carried out to receive a few robot manipulation commands.

- **Knowledge of basic programming with simulator.** Preliminary knowledge of every function available in the simulator was attained and, as the navigation algorithm performance was evaluated without problems, the first simulation approach was considered successful.

- **Integration of ultrasonic sensors and simple motors** was finally generated, with a differential configuration.

**4.2 Second iteration around the spiral**

**Objectives:** Obtain detailed knowledge of the simulation software model a second testing prototype, improve the autonomous navigation algorithm, implement the navigation algorithm, and continue developing the communication interface.

**Quadrant 1. Definition and specification of requirements:**

- **Prototype of mobile robot.** A second mobile robot prototype was required, which considered the similarities with a real, existing model.

- **Modular programming.** Same considerations as in the first spiral.

- **Connection with Matlab.** Same considerations as in the first spiral.

- **User interface.** Work was continued to obtain an interface capable of manipulating real-time simulation parameters.

- **2 ultrasonic sensors.** Same considerations as in the first spiral.

**Quadrant 2. Risk analysis**

- **Changes in functional and non-functional requirements.** Same considerations as in the first spiral.

- **Lack of experience using Matlab.** Same considerations as in the first spiral.

- **Additional 3D design software.** This type of software was used to build virtual models more similar to the real world. However, the lack of knowledge of this software delayed the project’s progress.
Quadrant 3. Design, testing and implementation

- **Closer similarity with the real model.** Figure 3 depicts the second prototype, based on a homemade physical model used at the Robotics laboratory of the doctoral degree programs of the **Instituto Tecnológico de Toluca**.

- **Scenario with moderate obstacles.** A new testing scenario featuring more obstacles was designed to test and improve the autonomous navigation algorithm; refer to Figures 4 and 5.

- **Navigation algorithm with double differential configuration.** The autonomous navigation algorithm was modified and adapted to a double differential configuration due to the morphology of the real prototype.

Quadrant 4. Evaluation

- **Virtual model 80% similar to the real model.** This second spiral used a reference physical model for comparison purposes. The achieved similarity resulted in better use of the simulator and the 3D modeling software.

![Fig. 4. Second prototype developed.](image1)

![Fig. 5. Second testing scenario.](image2)

- **Connection with Matlab.** Constant use of Matlab resulted in good communication with the simulator, achieving good control through commands placed within the interface.

- **Ultrasonic sensors and simple motors in a double differential configuration.** Modification of the preliminary prototype allowed for increased knowledge of the simulator, less implementation time and redesign of the autonomous navigation algorithm.

As part of a research project, some of the requirements evolved according to the changes made to the robot’s navigation methods; therefore, its requirements had to change, with the consequent evolution of the prototype.

4.3 Third iteration around the spiral

The objectives for the third spiral were:

- Use the advanced functions of simulation software.
- Model a third testing prototype.
- Improve the autonomous navigation algorithm.
- Implement navigation algorithm.
- Continue development of a communication interface with Matlab.

Quadrant 1. Definition and specification of requirements

- **Prototype of mobile robot.** This third spiral required a more accurate model, almost completely comparable to a real model; therefore, this phase considered a commercial model provided in the Marilou Robotics 2010 simulation software catalog.

- **Modular programming.** Object-oriented programming and module development were still used to control both the communication with sensor devices and also the module that would be used to redesign the navigation algorithm. Visual C++ was defined as programming IDE.

- **Connection with Matlab.** Connection with Matlab had to be validated in stability and scalability.

- **User interface.** Development of this interface continued to modify simulation parameters.

- **2 ultrasonic sensors.** The requirement of these two sensors is kept to continue testing and performance of the navigation algorithm.
Quadrant 2. Risk analysis
- **Navigation algorithm was inadequate in scenarios with many obstacles.** An unforeseen risk was related to performance of the autonomous navigation algorithm, because previous testing was carried out in scenarios with a limited number of obstacles; if the navigation algorithm performed poorly in other scenarios, its constant modification resulted in a delay in the remaining activities of the project.
- **Changes in functional and non-functional requirements.** This risk was considered again.
- **Additional 3D design software.** Development of future prototypes requires knowledge and skills to use the 3D modeling software in order to build virtual models that closely resemble real models. Lack of knowledge to use this software may delay the project’s progress.

Quadrant 3. Design, testing and implementation
- **Model alternative to real model.** Figure 6 depicts a model alternative to the real model used for testing at the second spiral; however, this model coincides with a real commercial model, so the objective of carrying out tests with virtual models similar to real models is still accomplished.

![Fig. 6. Third prototype developed.](image)

- **Scenario with increased number of obstacles.** The scenario used for this spiral is depicted in Figure 7, which included more obstacles to keep testing the performance of the autonomous navigation algorithm.
- **Interface shows distances of ultrasonic sensor during navigation.** A second interface was designed to consider distances of the ultrasonic sensor and status of components coupled to the mobile robot.

![Fig. 7. Third testing scenario.](image)

Quadrant 4. Evaluation
- **Interface showing distances of ultrasonic sensor.** After improving knowledge of the simulator functions and programming language used, it was possible to create a second interface, which shows real-time distances taken by ultrasonic sensors; this
implied a considerable progress for the future implementation of the mapping algorithm, along with the navigation algorithm.

- **Correct avoidance of obstacles in a scenario similar to the real testing scenario.** Due to the constant navigation algorithm redesigning, it was better adapted and capable of avoiding obstacles consistently and correctly.

### 4.4 Fourth iteration around the spiral

Objectives suggested for the fourth spiral were:

- Use advanced functions of simulation software.
- Model a fourth testing prototype.
- Improve the autonomous navigation algorithm.
- Implement navigation algorithm.
- Carry out the first round of integration tests with the mapping algorithm.

#### Quadrant 1. Definition and specification of requirements

- **Prototype of mobile robot.** At this level, the design of a fourth mobile robot prototype was required; for this, the existing real prototype was used as reference. The similarity between the virtual and real prototypes was determined to be above 80% with respect to physical features.
- **Modular programming.** Object-oriented programming was still being used, and a new module responsible for managing data received for the mapping algorithm was integrated.
- **User interface.** A new C++ programmed user interface was required in order to show the status of all the robot devices when simulation started and the measurements received by ultrasonic sensors, as well as laser sensor measurements.
- **2 ultrasonic sensors.** These two sensors were still used for the proper autonomous navigation of the robot through the testing scenario.
- **1 laser sensor.** Usage of this sensor was considered for the fourth spiral to initiate integration of the mapping algorithm with the project’s original code.

#### Quadrant 2. Risk analysis

- **Complex integration of Matlab with the mapping algorithm.** Due to the mapping algorithm’s nature, we considered the possibility that data was not going to be processed completely by the Matlab software, so it was necessary to consider discarding this software’s usefulness for the project.
- **Changes in functional and non-functional requirements.** This risk was considered again in the fourth spiral.
- **Additional 3D design software.** This risk was considered too in the fourth spiral.

#### Quadrant 3. Design, testing and implementation

- **Model alternative to real model.** Figure 8 depicts the fourth prototype developed for simulation tests; as in previous models, this model corresponds to an existing commercial prototype.

![Fig. 8. Fourth prototype developed.](image)

- **Scenario with increased number of obstacles.** Figure 9 shows the fourth testing scenario, partially corresponding to the layout of the office of the Department of Postgraduate Studies and Research of the Instituto Tecnológico de Toluca, in which physical tests with the real model were planned.
Quadrant 4. Evaluation

- Connection with Matlab is discarded. Connection with Matlab is no longer considered part of the project due to the complexity of integration with the mapping algorithm being implemented, and that will be considered in later stages.
- Virtual model is 90% similar to the real model. This measurement is based on visual and performance features compared against the proposed real model. This reflects increased knowledge and experience using the simulation software, the 3D modeling software and the improvements in the autonomous navigation algorithm.

4.5 Fifth iteration around the spiral

Objectives: Use advanced functions of simulation software, model the fifth testing prototype, improve the autonomous navigation algorithm, implement navigation algorithm, improve SLAM mapping algorithm, implement SLAM mapping algorithm.

Definition and specification of requirements:

- Prototype of mobile robot. A model 95% similar in physical and performance features to a physical model was required.
- Modular programming. The same consideration of the first spiral was made; furthermore, the autonomous navigation algorithm was integrated with the mapping algorithm, using the SDL library, for the graphic sketching of the real-time map generated by simulation.
- SLAM algorithm. Programming of the proposed mapping algorithm was required, based on the FastSLAM algorithm.
- Autonomous navigation algorithm. The original autonomous navigation algorithm was modified for increased efficiency and flexibility in a new virtual environment, which corresponded to a real scenario.
- User interface. A user interface showing the status of the mobile robot, the devices mounted on it, the measurements gathered by ultrasonic sensors and laser sensor measurements was required.
- 3 ultrasonic sensors. A new sensor was required to improve testing efficiency.
- 1 laser sensor. This type of sensor was required to experiment with the SLAM mapping algorithm.

Risk analysis

- Changes in functional and non-functional requirements. Same consideration of the first spiral.
- Implementation complexity of mapping algorithm (SLAM). Due to the lack of experience of previous implementation of mapping algorithm.
- Inadequacies of autonomous navigation algorithm during integration with mapping algorithm. When combining two algorithms in a same programming code, inadequacies inherent to their complexities may arise, resulting in malfunctioning.

Design, testing and implementation

- Model alternative to physical model. Figure 10 depicts the fifth prototype developed in simulation tests.
- Redesign of navigation algorithm. The autonomous navigation algorithm was redesigned to adapt it to scenarios with a great number of obstacles.
- Integration of laser sensor. The laser sensor was integrated during design, and axial scanning wide enough to implement the mapping algorithm (SLAM) was configured.
End of definitive testing scenario. Scenario modeling was finalized during this fifth spiral, which was based on the layout of the office of the Department of Postgraduate Studies and Research of the Instituto Tecnológico de Toluca (see Figure 11).

Evaluation

- Similarities between the virtual and physical models of the mobile robot. The virtual model was 95% similar to the proposed real model in visual and performance features.
- Implementation of the SLAM mapping algorithm. The mapping algorithm was implemented with optimal results in scenarios with a small amount of obstacles. The algorithm obtained good results in the Postgraduate Studies office layout (fifth scenario, refer to Figure 11) scenario.
- Improvement to the autonomous navigation algorithm. This algorithm adapted better to scenarios with a great number of obstacles, thanks to the three ultrasonic sensors that allowed sketching new paths.
- Finalization of testing scenario. The testing scenario proposed for this spiral was considered definitive and will be used in later cycles of the spiral, as it is very similar to the physical scenario.

5 Results

In order to validate the simulation of the fifth mobile robot prototype, we had to analyze considerations such as usability, efficiency and functionality when creating the robot. This was possible after applying the criteria proposed by Marcelo Pérez Ibarra [19], who uses measurements to validate the quality of credibility and interaction of distance learning courses. These measurements were applied to evaluate the fifth prototype, considering the following categories:

- **Functionality**. To provide functions that correspond to the user needs when the prototype is used under specific conditions.
- **Usability**. To be understood and used by the user when used under specific conditions.
- **Efficiency**. To provide appropriate performance of the prototype with respect to the resources used under established conditions.

The four categories were used as a basis to develop an assessment questionnaire of 26 points for our virtual prototype, which was distributed in a sample of 30 users that included professors, researchers and students of undergraduate, master and doctoral studies. From the four categories to evaluate, the “usability” one integrated four evaluation parameters: environment comprehension, obstacle repertoire, usage and adaptability, with resulting values displayed in Table 2.

<table>
<thead>
<tr>
<th><strong>Usability</strong></th>
<th>Does the virtual robot travel correctly in the scenario?</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Are there enough obstacles shown in the scenario where the virtual robot is traveling?</td>
<td>24.4</td>
</tr>
<tr>
<td></td>
<td>Is the user maintaining interest in the motions and behavior of the virtual robot?</td>
<td>24.3</td>
</tr>
<tr>
<td></td>
<td>Does the virtual robot easily adapt to an unknown scenario?</td>
<td>25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Usability</strong></th>
<th>Does the virtual robot travel correctly in the scenario?</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Are there enough obstacles shown in the scenario where the virtual robot is traveling?</td>
<td>24.4</td>
</tr>
<tr>
<td></td>
<td>Is the user maintaining interest in the motions and behavior of the virtual robot?</td>
<td>24.3</td>
</tr>
<tr>
<td></td>
<td>Does the virtual robot easily adapt to an unknown scenario?</td>
<td>25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Usability</strong></th>
<th>Does the virtual robot travel correctly in the scenario?</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Are there enough obstacles shown in the scenario where the virtual robot is traveling?</td>
<td>24.4</td>
</tr>
<tr>
<td></td>
<td>Is the user maintaining interest in the motions and behavior of the virtual robot?</td>
<td>24.3</td>
</tr>
<tr>
<td></td>
<td>Does the virtual robot easily adapt to an unknown scenario?</td>
<td>25</td>
</tr>
</tbody>
</table>

After graphing the results from the four categories, the information shown in Figure 12 is obtained.
Finally, the results from the four categories included one high mark (90.0 and 92.0 respectively) for usability and functionality of the virtual prototype, and one low mark (75.0) for efficiency, as the questionnaire answers reflected the need to integrate more aspects of real environments and a few adjustments before manufacturing the robot.

6 Conclusions and future work

Using the simulation process and formality recommendations set by Systems Engineering, it was possible to observe the robot’s behavior and study its operation in different environments and conditions. The robot’s simulation discarded the physical construction of all different prototypes developed in five lifecycles of the spiral model, therefore avoiding great losses in time and cost, optimizing the process.

Documentation resulted from the four quadrants of the five lifecycles of the spiral model made it easier to access information, offering an increased number of cases of studied applications. Future projects will benefit from reusing this information, early detection of risks and the creation of quality products.

The simulation work managed and formalized with the support of Software Systems Engineering through the spiral lifecycle model allowed testing operation theories of processes to carry out and human reasoning.

Each iteration of the spiral lifecycle model considered the simulation objectives and methodology to identify the needs to be satisfied by the robot and solution alternatives, by defining ways to accomplish the proposed objectives.

The use of software engineering in systems simulations will focus on the new paradigms of engineering, in order to satisfy the expectations of current technologies and simulation.

As integral part of our research work, software engineering allowed us to analyze, model and simulate a mobile robot in a static scenario when interacting with the simulation process, applying the proper methodologies and techniques to solve problems, and defining the necessary steps to reach knowledge representation.

7 Acknowledgements

We would like to thank DGEST for their support and approved resources to carry out this work as part of the project “Systems engineering as an integral model to develop adaptive applications”, coded 4389.11-P. Likewise, we thank them for approving the project “Prototype of a virtual system of a pair of opposing pneumatic artificial muscle miniactuators based on a McKibben model”, coded TOL-DCIET-2010-102, which has been used as a basis to continue the research related to this project.
References