

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# ROBOTIC MODELLING AND SIMULATION: THEORY AND APPLICATION

<sup>1</sup>Muhammad Ikhwan Jambak, <sup>1</sup>Habibollah Haron,  
<sup>2</sup>Helmee Ibrahim and <sup>2</sup>Norhazlan Abd Hamid

<sup>1</sup>*Soft Computing Research Group, Universiti Teknologi Malaysia,  
Malaysia*

<sup>2</sup>*Department of Modeling & Industrial Computing, Faculty of Computer Science &  
Information System, Universiti Teknologi Malaysia,  
Malaysia*

## 1. Introduction

The employment of robots in manufacturing has been a value-adding entity for companies in gaining a competitive advantage. Zomaya (1992) describes some features of robots in industries, which are decreased cost of labour, increased flexibility and versatility, higher precision and productivity, better human working conditions and displaced human working in hazardous and impractical environments.

Farrington *et al.* (1999) states that robotic simulation differs from traditional discrete event simulation (DES) in five ways in terms of its features and capabilities. Robotic simulation covers the visualization of how the robot moves through its environment. Basically, the simulation is based heavily on CAD and graphical visualization tools. Another type of simulation is numerical simulation, which deals with the dynamics, sensing and control of robots. It has been accepted that the major benefit of simulation is reduction in cost and time when designing and proving the system (Robinson, 1996).

Robotic simulation is a kinematics simulation tool, whose primary use is as a highly detailed, cell-level validation tool (Farrington *et al.*, 1999), and also for simulating a system whose state changes continuously based on the motion(s) of one or more kinematic devices (Roth, 1999). It is also used as a tool to verify robotic workcell process operations by providing a “mock-up” station of a robots application system, in order to check and evaluate different parameters such as cycle times, object collisions, optimal path, workcell layout and placement of entities in the cell in respect of each other.

This paper presents the methodology in modelling and simulating a robot and its environment using Workspace and X3D software. This paper will discuss the development of robotic e-learning to improve the efficiency of the learning process inside and outside the class.

This paper is divided into five sections. Section 2 discusses the robotic modelling method. Section 3 discusses robotic simulation. Its application using Workspace and X3D is presented in Section 4, and a conclusion is drawn in Section 5.

## 2. Robotic Modelling Method

This section presents the methodology in modelling and simulating the robot and its environment. There are two types of methodology being applied, which are the methodology for modelling the robot and its environment proposed by Cheng (2000), and the methodology for robotic simulation proposed by Grajo *et al.*(1994). The methodologies have been customised to tailor the constraints of the Workspace software. Section 4 presents experimental results of the project based on the methodologies discussed in this section.

### 2.1 Robotic Modelling

Robotic workcell simulation is a modelling-based problem solving approach that aims to sufficiently produce credible solutions for a robotic system design (Cheng, 2000). The methodology consists of six steps, as shown in Figure 1.

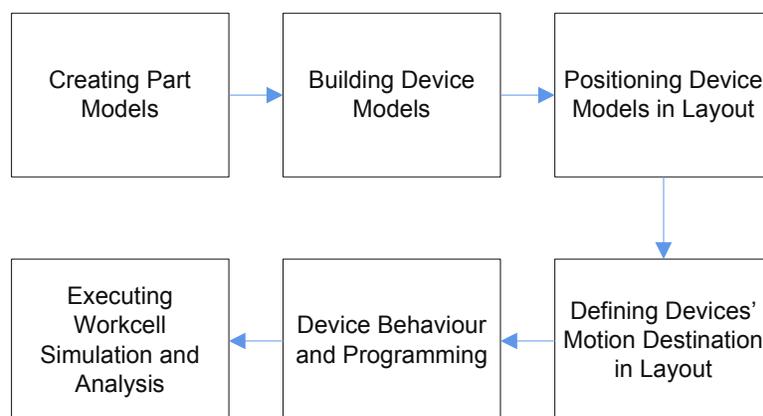


Fig. 1. A methodology for robotic modelling

#### 2.1.1 Creating part models

Part model is a low-level or geometric entity. The parts are created using basic elements of solid modelling features of Workspace5. These parts consist of the components of the robot and the devices in its workcell, such as the conveyor, pallet and pick up station.

#### 2.1.2 Building device models

Device models represent actual workcell components and are categorized as follows: robotic device model and non-robotic device model. The building device model starts by positioning the base of the part model as the base coordinate system. This step defines links with the robot. These links are attached accordingly to its number by using the attachment feature of Workspace5. Each attached link is subjected to a Parent/Child relationship.

#### 2.1.3 Positioning device models in layout

The layout of the workcell model refers to the environment that represents the actual workcell. As in this case, the coordinates system being applied is the Hand Coordinates System of the robot. Placement of the model and devices in the environment is based on the actual layout of the workcell.

### 2.1.4 Defining devices motion destination in layout

The motion attributes of the device model define the motion limits of the joints of the device model in terms of home, position, speed, accelerations and travel. In Workspace5, each joint is considered part of the proceeding link. A joint is defined by linking the Link or Rotor of the robot, for example Joint 1 is a waist joint which links the Base and the Rotor/Link. Each joint has its own motion limits. Once the joints have been defined, Workspace5 will automatically define the kinematics for the robot.

### 2.1.5 Device behaviour and programming

Device motion refers to the movement of the robot's arm during the palletizing process. The movement is determined by a series of Geometry Points (GPs) that create a path of motion for the robot to follow. Positioning the GP and the series is based on the movement pattern and the arrangement of bags. The GP coordinates are entered by using the Pendant features of Workspace5. There are three ways to create the GP: by entering the value for each joint, by entering the absolute value of X, Y and Z, and by mouse-clicking.

### 2.1.6 Executing workcell simulation and analysis

The simulation focuses only on the position of the robot's arm, not its orientation. After being programmed, the device model layout can be simulated over time. Execution of the simulation and analysis is done using the features of Workspace5. The simulated model is capable of viewing the movement of the robot's arm, layout checking, the robot's reachabilities, cycle time monitoring, and collision and near miss detection.

## 3. Robotic Simulation

### 3.1 Definition of Simulation

Shannon (1998) offered a good definition of simulation: "We will define simulation as the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies for the operation of the system". Robot simulation software or simulator is a computer program which mimics the elements of both the internal behaviour of a real-world system and the input processes which drive or control the simulated system. There are a few reasons why the simulation approach became the main option in real-world robotic related activity. Typically, most users make simulations because the experiment with the real world still does not yet exist, and experimentation with the robot's hardware is expensive, too time-consuming and too dangerous.

### 3.2 Type of Simulation

There are two types of simulation (F.E Cellier, 2006): *discrete event* simulation and *continuous system* simulation. *Discrete event* simulation divides a system into individual events that have their own specific start time and duration. The overall behaviour of a complex system of a real-world object will be determined from the sequencing and interactions of each event. This technique usually focuses on modelling the control logic for the routing of material and interaction of equipment. It also typically applies statistics to the system to simulate things like equipment breakdown or mixtures of different product models.

*Continuous system* simulation (F.E Cellier, 2006) describes systematically the mathematical models used in dynamic systems, and is usually done so using sets of either ordinary or partially different equations, possibly coupled with numerical integration, differential equation solvers or other mathematical approaches that can be simulated on a digital computer. Often, electrical circuits, control systems or similar mechanical systems are simulated in this way. Specific applications like thermal dynamics, aerodynamics, aircraft control systems or automobile crashes are commonly simulated with continuous system simulation.

### 3.3 Robot Programming

The main importance of robot functionality is its flexibility and ability to rearrange the new production, and its movement range. The flexibility of the robot depends on presupposed effective programming. Principally, the robot programming can take place in two different ways: on-line or off-line. In on-line programming, the use of the robot and equipment is required, whereas off-line programming is based on computer models of the production equipment. Both these methods have advantages and disadvantages. In this section we will look at how the two methods can be combined.

#### 3.3.1 On-Line Programming

Currently, the operation of robot programming is through either on-line programming or off-line programming. The definition of on-line programming (Kin-Hua Low [10]) is a technique to generate a robot program using real robot systems. An on-line programming robot may be suitable to implement robot use by repeating a monotonous motion. The advantage of on-line programming is that it is easy to access. Its most significant advantage is that the robot is programmed in concordance with the actual position of the equipment and pieces.

However, the most significant disadvantage of on-line programming is the slow movement of the robot, the program implementation, program logic and calculations being hard to programme, the suspension of production whilst programming, poor documentation and costs equivalent to production value. The differentiation between on-line and off-line programming is shown clearly in the picture below.



Fig. 2. On-line robot programming

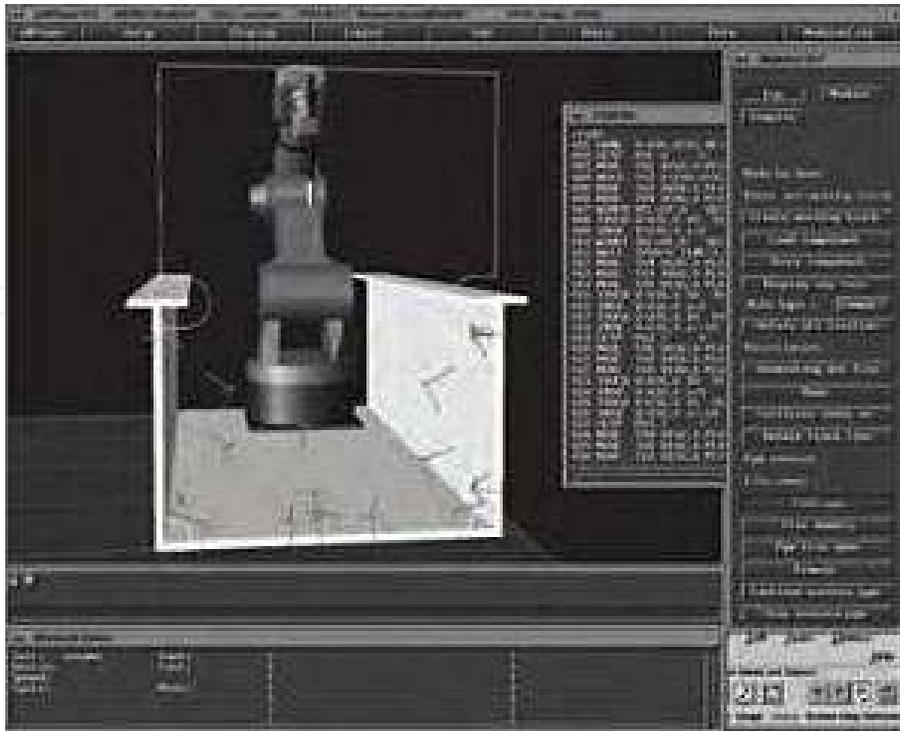


Fig. 3. Off-line robot programming

### 3.3.2 Off-Line Programming

The definition of off-line programming is that it is the technique used to generate a robot program without using a real machine. There are several advantages to this that have been cited by Bien (1998) resulting in a reduction in robot downtime, programs being prepared without interruptions to robot operation, and removal of the human from the potentially dangerous environment. There are several types of programming language used in off-line robotic programming, but the most popular off-line programming software was built using the JAVA language. Off-line programming using JAVA language become famous because it was easy to use, could be integrated with UML, supported C/C++ languages, had architecture independence and had an advanced network support.

The main advantage of off-line programming is that it does not occupy production equipment, and in this way, production can continue during the programming process. But on-line programming is the largest proportion of robot programming today due to the fact that off-line programming has had a very high burden rate and has demanded the need of expert users.

Advanced off-line programming tools contain facilities for debugging and these assist in effective programming. The programming tools support the utilization of supporting tools for the programming process, for instance optimization of the welding process.

However, off-line programming also has its disadvantages such as the fact that it demands investing in an off-line programming system. Most simulation tools/simulators are implemented with four characteristic: workspace visualization, trajectory planning, communication with robot control and system navigation.

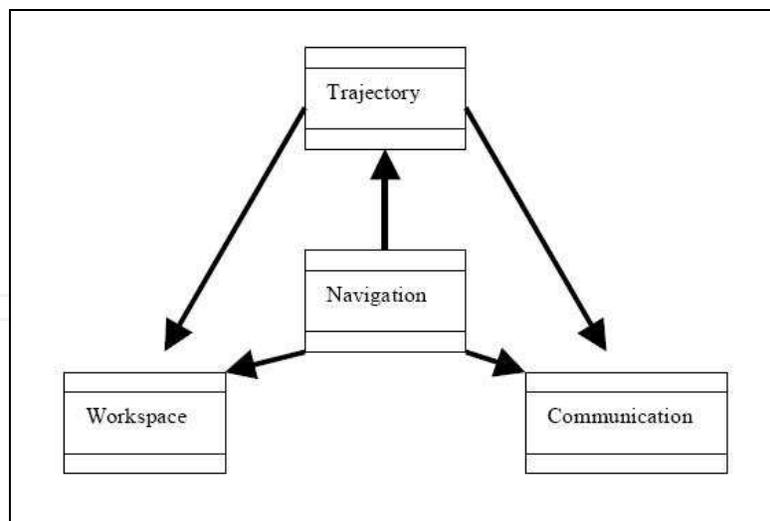


Fig. 4. Simulation tool modules

### 3.3.3 Off-Line Programming

Hybrid programming is a combination of both of the robotic programming methodology advantages shown above. By using both advantages, the programming technique can be optimized. A robot program consists mainly of two parts: locations (position and alignment) and program logics (controller structures, communication and calculations).

The program logics, debugging and simulation facilities are effectively developed on off-line programming. The main part of the movement can be created off-line by reusing the availability of CAD data and by programmer interaction.

Commands for movement to locating the piece placement in the robot's workcell can be more properly programmed on-line. In this situation, the advantages of both programming method can be utilized, indirectly increasing the flexibility in production.

The usage of hybrid programming is a very practical way of increasing flexibility in production and thereby increasing the effect of robot manufacturing. In the same way, rearrangement time can be substantially reduced, allowing for cost effectiveness even in the production of small batches.

### 3.4 Simulation Packages

The robotic simulation package is a tool which is used to create embedded applications for a specific (or not) robot without depending "physically" on the actual robot, thus saving cost and time. In some cases, the applications that were developed with the simulation package can be transferred to the real robot without modifications. This application allows the user to create a simple world and to programme this robot to interact with these worlds.

Most robotic simulation packages have their own unique features, but the main features for 3D modelling are robot rendering and environment. This type of robotics software has a simulator that is a "virtual" robot, which is capable of emulating the motion of an actual robot in a real work envelope. Some robotic simulation tools such as Matlab-Simulink can be used significantly in robot simulation, providing an interesting environment. Matlab-Simulink is an interactive robot simulation software that can be used as an interface of the system so that users can communicate with the system. This robotic simulation tool gives alternatives to minimize the limitation of Web Programming Language (WPL) and

Structured Programming Language (SPL). M. I. Jambak *et al.* (2008) state that Matlab-Simulink has been used in their previous research to model the graphical design of the Mitsubishi RV-2AJ robots and is dynamic in a 3D virtual reality (VR) environment, and uses the V-Realm Builder virtual programming language to apply the virtual reality modelling language (VRML).

Nathan *et al.* (2006) describe Virtual Reality Modelling Language (VRML) currently, as the de facto standard for web based 3D visualizations, which allows for easy definition of geometric shapes and provides many advanced 3D graphical functions such as lighting models and surface materials. VRML allows for simple interactions between a user of a virtual world and various objects within the world. Currently, VRML has been supported with various user browser and modelling programs.

Java3D (Nathan *et al.* 2006) is a simulation package which provides an object-oriented language-based approach for designing a 3D system. Java3D offers a high-level Application Programming Interface (API) for 3D scene description and graphical control. Besides that, it also allows for a fully object-oriented approach to define and control the virtual agent and its environment. Java3D is also designed to take advantage of multi-threaded programming techniques, allowing for better performance from the implementation.

Webots (Michel, 2004) is one of most popular mobile robot simulations and is widely used for educational purposes. Webots uses the ODE (Open Dynamics Engine) for collision detection and simulating rigid body dynamics. It contains a rapid prototyping tool, allowing the user to create a 3D virtual world. Webots runs on Windows, Linux and Mac OS X.

Microsoft Robotics Studio (Eric Colon and Kristel Verbiest, 2008) is a 3D modelling and simulation environment for mobile robots operating in real-world conditions, which respects the law of physics and runs on top of DirectX.

### 3.5 Robotic Simulation

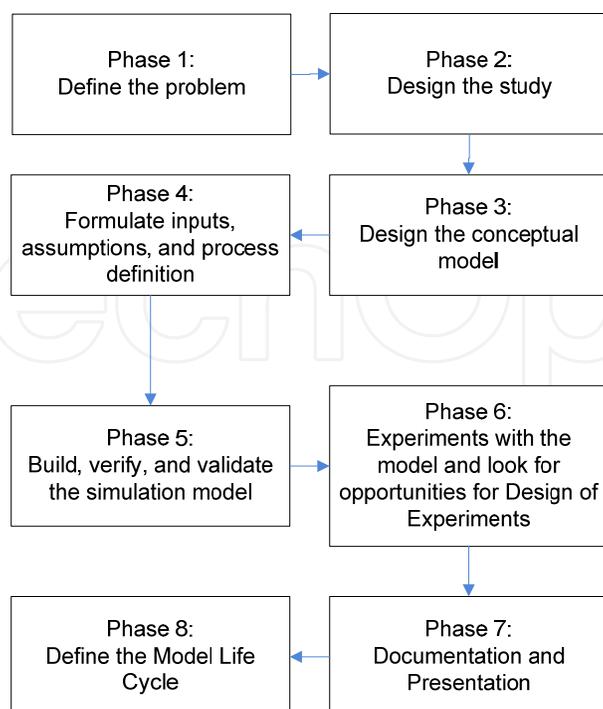


Fig. 5. A methodology for robotic simulation

The methodology consists of eight phases but the discussion only executes up to eight phases, as shown in Figure 5.

### **3.5.1 Define the problem**

Problem identification is defined during the preliminary analysis of the problem's background. If the current system has no computer-based model that represents the robotic application, it is impossible to monitor and evaluate the performance of the robotic palletizing system. In contrast, the definition and analysis of the current system are easier to implement.

### **3.5.2 Design the study**

The study is limited to the scope of the project. This phase acquires appropriate decisions for the tools and methodology to be used. Besides, proper planning and milestones need to be developed.

### **3.5.3 Design the conceptual model**

The conceptual model is using the current application of the robotic system. This phase acquires collection of data of the parameters for the robotic workcell development. These data include layout of the robotic application, geometry configuration of the robot, robot motion parameters and the robot cycle time.

### **3.5.4 Formulate inputs, assumptions, and process definition**

Modelling the robot application focuses on three activities: building the robot, motion path programming of the palletizing process, and running the simulation. Building the robot model is based heavily on the geometrical data of the robot using the CAD features of Workspace5. The dimension refers to the CAD drawing of the robot. Spatial data need to be considered in determining the motion path, such as the point of the pick up station where the robot will do the pick and place operation, the points that represent an arrangement and layer of the item to be picked, and the position of points in  $x$ ,  $y$  and  $z$  coordinates.

### **3.5.5 Build, verify and validate the simulation model**

During this phase, development of the robotic workcell is based on the methodology proposed by Cheng (2000). This is an interactive phase which aims to improve the model's precision and motion. Validation towards the model is based on the visualization of the system layout and robot cycle time in completing a task. The layout is generated using Workspace5 and compared to the actual system layout. During the gathering of preliminary data, a movie that shows the actual robot performing a task in a one-day operation is recorded. The model is assumed to represent the actual system once operated at the same movement of the actual system and is capable of performing at a similar cycle time as in the movie.

### 3.5.6 Experiments with the model and look for opportunities for Design of Experiments

This phase is similar to step six in the methodology by Cheng (2000). A simulation is run in order to visualize the arm movement and an analysis on collision detection is provided. Execution of the simulation is done using the features of Workspace5 for simulation. Workspace5 allows layout checking in order to prepare other devices within the robot's reachability. It is also capable of generating a working envelope for namely, two joints. During simulation, a cycle time is displayed. The simulation allows collision and near-miss detection among robot joints, and between the joints and any object within the workcell. The result is displayed in a report. This project is off-line programming. Neither robot language is generated or implemented at the actual workcell.

### 3.5.7 Documentation and presentation

This phase gathers and documents all the results generated from the simulation. A written report provides a better understanding of the experiment's executions and analysis.

There are advantages and disadvantages for this methodology (Mohd Johari *et al.*, 2008). The advantage of using this methodology is that it saves costs, avoiding designing, building, testing, redesigning, rebuilding and retesting which would be an expensive project. Simulations take the building or rebuilding phase out of the loop by using the model that has already been created in the design phase. Usually, the simulation test is cheaper and faster than performing multiple tests of the design each time.

The second advantage of using this methodology is the level of detail that we can get from the simulation. A simulation can give results that are not experimentally measurable with our current level of technology. Results such as time taken to complete the simulation and the details of collision detection of the simulation are not measurable by any current device.

There are also disadvantages to performing this methodology for robotic simulation. The first is simulation errors. Any incorrect key store for the value of the robot's details has the potential to alter the result of the simulation or give the wrong result. To get an accurate result, we must first run a baseline to prove that it works. In order for the simulation to be accepted in the general community, the experimental result is taken and simulates them. If the two data sets are compared, then any simulation of the design will have some credibility.

## 4. Application

This section describes two of the several projects that are related to modelling and simulation. The first is building robot simulation using Workspace5 and the second is robot simulation using X3D for e-learning. Below is an explanation of both of these:

### 4.1 Building Robot Simulation Using Workspace5

The experimental results presented in this section are based on authors' experience in supervising undergraduate and postgraduate final project works reported (Mohd Johari, 2008; Ariffin, 2007; Mohd Salih, 2008; Abdul Rahim, 2008; Muhammad Noor, 2005; Arifin, 2007; Zainal, 2008; Shafei, 2008, and Sukimin, 2007). Different types of robots were involved

in the experiments, which are situated in Universiti Teknologi Malaysia and other institutions.

Basic elements of solid modelling features in Workspace5 have been used to develop the robot and device models. Figures 3(a) and (b) show the development of the robot gripper and screwdriver device (Ariffin, 2008). Some solid modelling methods, such as union, subtract, or both, were applied in the models' development. Eventually, these models were compared with the actual robot for visual validation, as depicted in Figure 4.

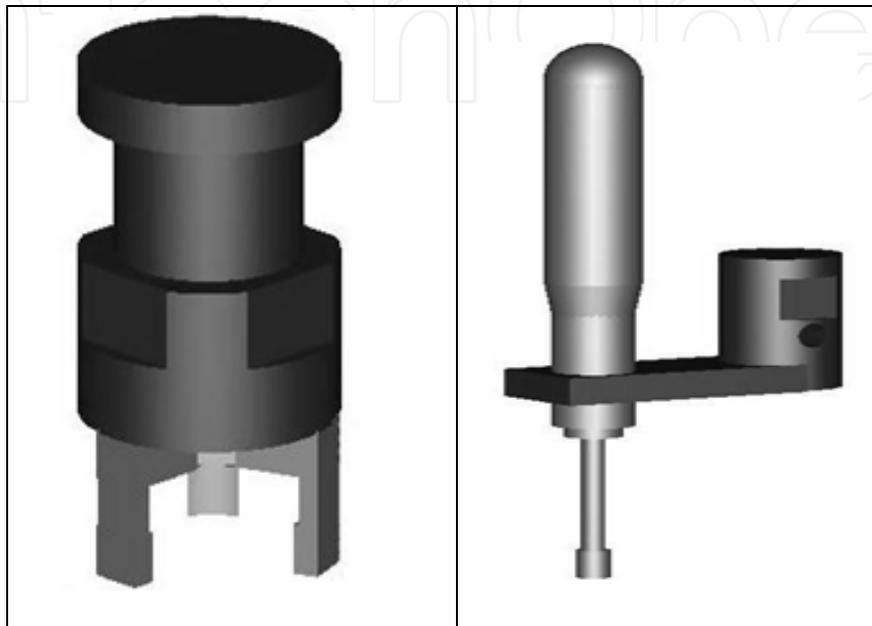


Fig. 6(a). Robot gripper and screwdriver model

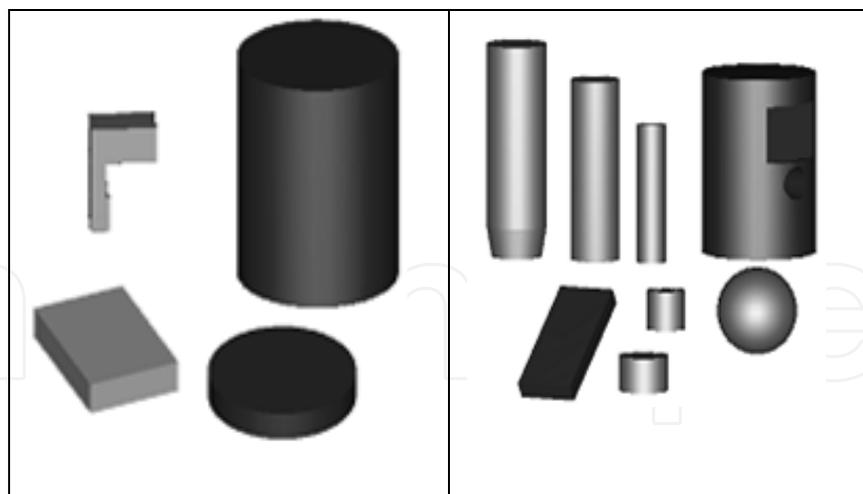


Fig. 6(b). Elements of robot gripper and screw driver model

Prior to simulating the robot movement and validating the simulation created in Workspace5, the actual robot movements first have to be specified and recorded. The cycle time of the actual robot completing time of certain tasks then has to be defined and compared with the cycle time of model simulation.

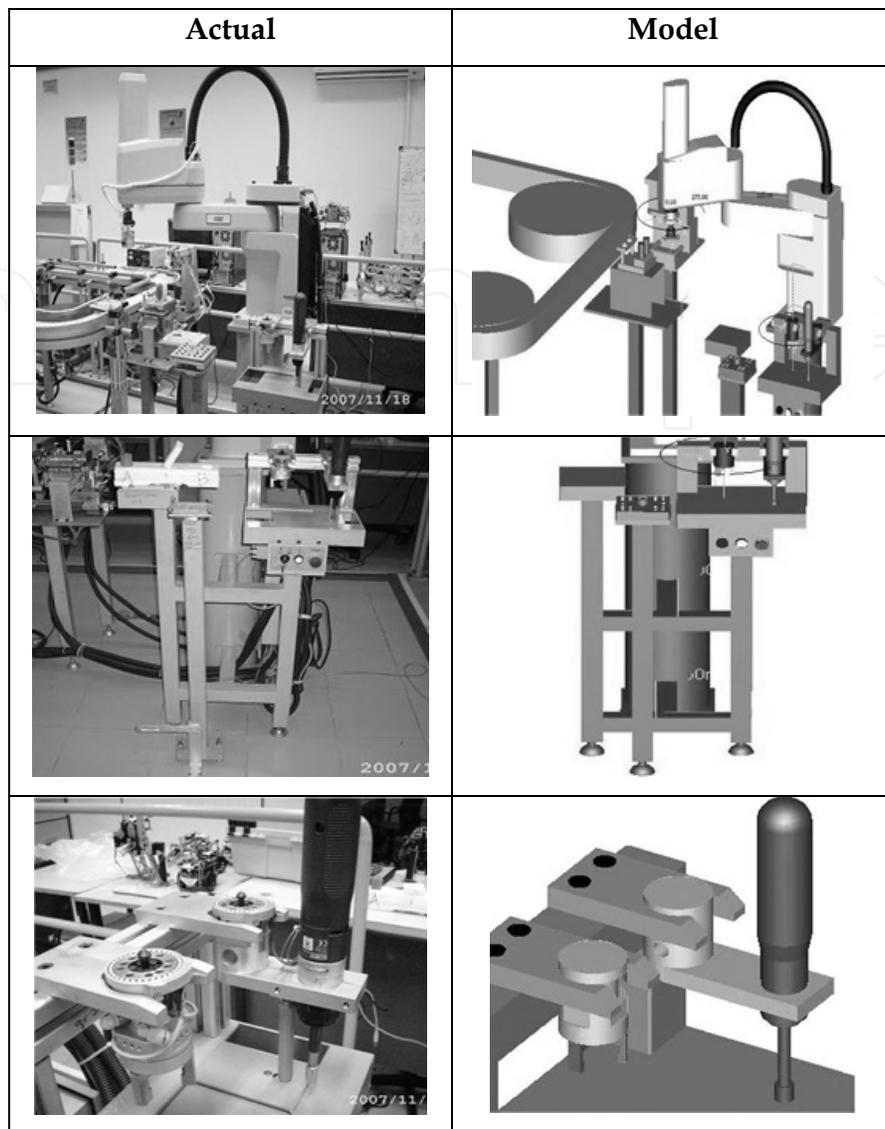


Fig. 7. Visual validation

Another project-work is reported in Nepal, R., and Baral, M. (2004), which is located in St. Cloud State University. Figures 5(a) and (b) show the development of the vacuum gripper attached to the Kawasaki 06L robot.

At the end of this project, the simulation is ready to grasp the object as depicted in Figures 6(a) and (b). When the cell reaches the bottom of the sooth, the robot grasps the object by its vacuum gripper and un-grasps the cell on the table, and moves back to its home position. Similarly, the remaining seven cells slide down the sooth in sequence and the robot picks and arranges the cells into a block on the table. The time taken for the complete simulation is 123.40 sec. There is no collision detected during the simulation.

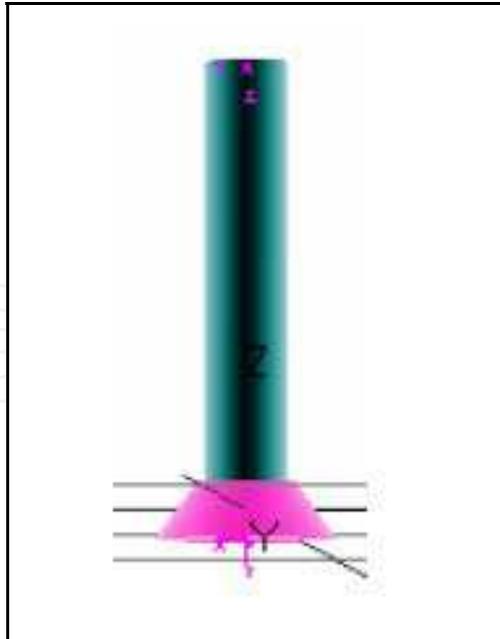


Fig. 8(a). Model of vacuum gripper

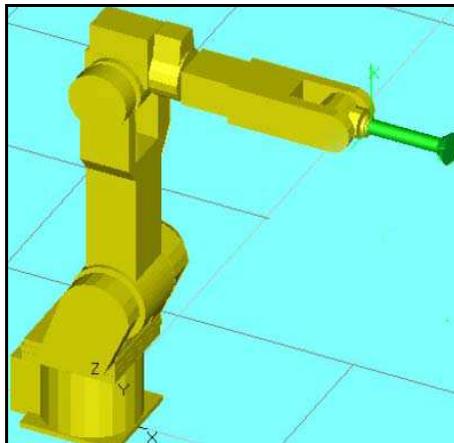


Fig. 8(b). Vacuum gripper is attached to Kawasaki 06L

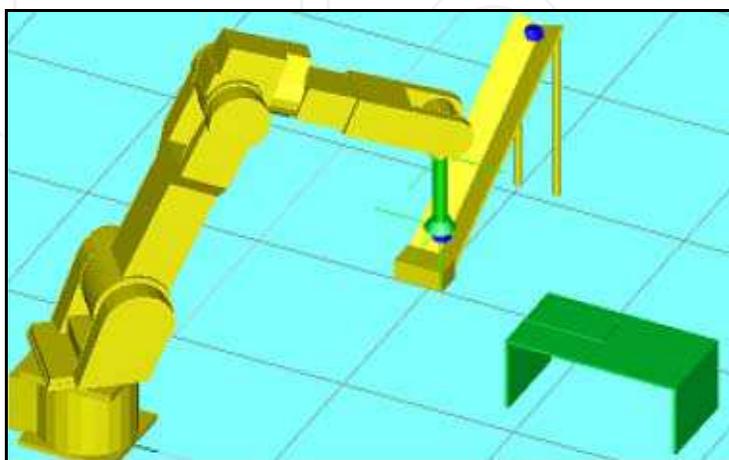


Fig. 9(a). Robot-picking cell

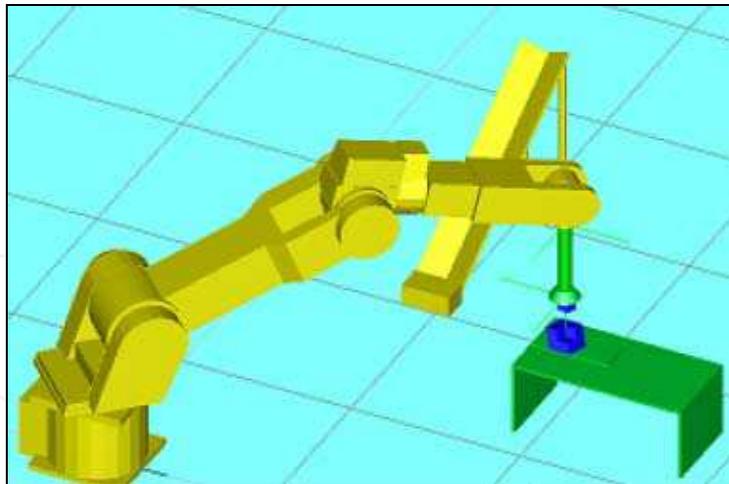


Fig. 9(b). Robot placing cells on the table

**4.2 Robot Simulation Using X3D for E-Learning**

This section will show the initial results based on the authors’ experience in developing the X3D model. Figure 4 shows the development of a virtual robot arm using the X3D programming written in X3D Edit 3.2 software. The X3D programming is similar to XML programming. Figure 5 shows some of the development programming.

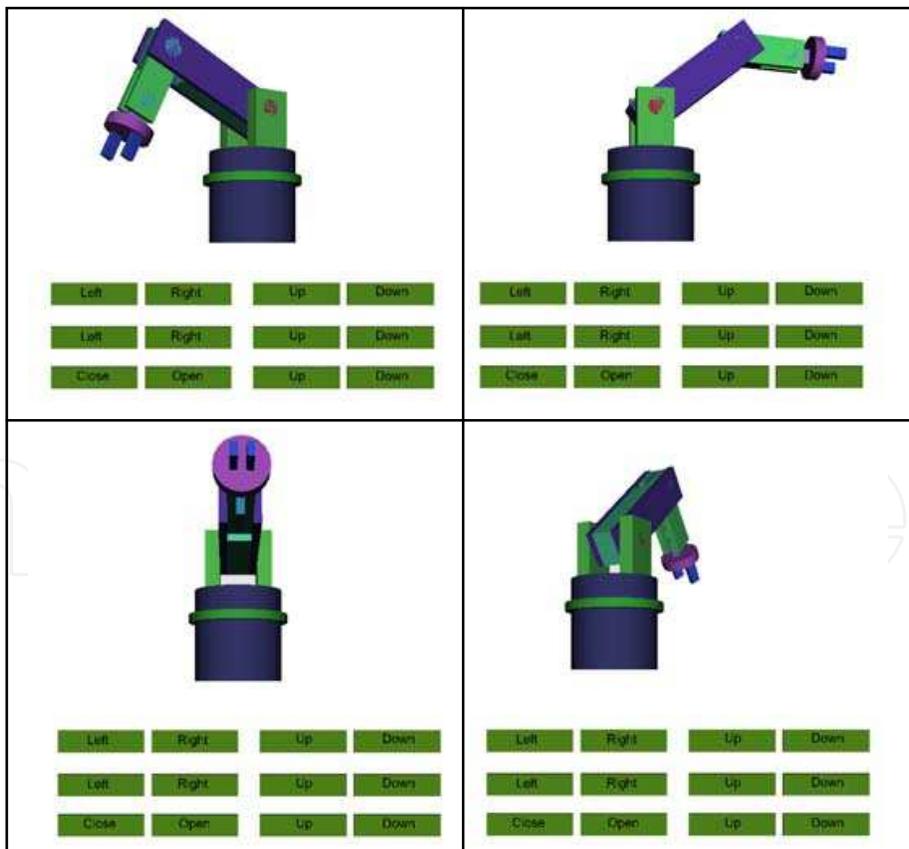


Fig. 10. Example movement of robotic simulation

At the end of this project, the virtual robot arm simulation is ready to capture the point of each movement and is also expected to generate the code based on Melfa Basic. To validate the robot simulation, the generated Melfa Basic code from the virtual robot simulation will be tested in the real environment of Melfa Basic Software and executed to the real robot. The virtual environment can also perhaps be simulated based on the input of the Melfa Basic code.

The virtual robot simulation will be embedded into an Internet web-server on a high-end server, and will be managed by content management tools. This phase also includes reliability and security testing. A simulation is run using the virtual tech pendant in order to visualize the arm movement using the client computer through web based. To simulate the virtual robot arm, we have to install a plug-in for the web browser such as Octaga Player or Cortona 3D. Different browsers will be used to make sure that the system is compatible with the browser to simulate the virtual robotic simulation.

The system can hopefully give students the realistic experience of simulation and modelling using this virtual robot arm through the e-learning portal. The information can be accessed simultaneously by users and they would not have to wait to seek the virtual robot arm simulation as this can be achieved by many users at the same time.

```

<TimeSensor DEF="TimerKanan" cycleInterval="5" loop="false"/>
  <OrientationInterpolator DEF="MuterKanan"
    key =      "0, 0.05, 0.1, 0.15,
                0.2, 0.25, 0.3, 0.35,
                0.4, 0.45, 0.5, 0.55,
                0.6, 0.65, 0.7, 0.75,
                0.8, 0.85, 0.9, 0.95,
                1"
    keyValue = "0 0 1 0,
                0 1 0 0.314159,
                0 1 0 0.628318,
                0 1 0 0.942478,
                0 1 0 1.25664,
                0 1 0 1.5708,
                0 1 0 1.88496,
                0 1 0 2.19911,
                0 1 0 2.51327,
                0 1 0 2.82743,
                0 1 0 3.14159,
                0 1 0 3.45575,
                0 1 0 3.76991,
                0 1 0 4.08407,
                0 1 0 4.39823,
                0 1 0 4.71239,
                0 1 0 5.02655,
                0 1 0 5.34071,
                0 1 0 5.65487,
                0 1 0 5.96903,
                0 0 1 0 "/>
<TimeSensor DEF="TimerKiri" cycleInterval="5" loop="false"/>

```

Fig. 11. Example movement robotic simulation programming

## 5. Discussion

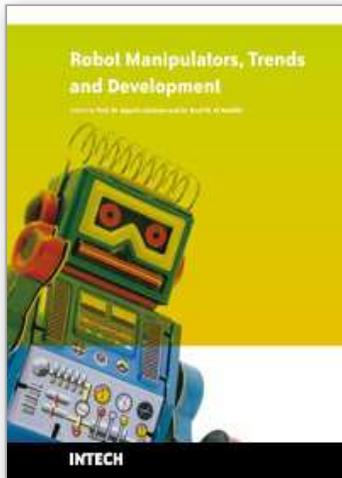
The system development of virtual robotics simulation will be the alternative for the robotics learning process. This project will effort the virtual robotics simulation that will combine all the information about the robot arm. This system can also hopefully be the training place for students to simulate the robot's movement virtually and remotely for learning purposes.

## 6. References

- Abdul Rahim, N. (2008), Modelling and Simulation of FARA Robot RSM7 Movement and Its Environment (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Ariffin, N. S. (2008), Modelling and Simulation of SCARA Adept Cobra i600 Robot Arm Movement and Its Environment (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Arifin, S. M. (2007), Modelling and Simulation of Mitsubishi RV-2AJ Robot Arm Movement (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Bien, C. (1998) Simulation A Necessity In Safety Engineering. *Robot World*, Vol.10, No.4, pp.22-27.
- Cheng, F. S. (2000). A Methodology for Developing Robotic Workcell Simulation Models. *Proceedings of the 2000 Winter Simulation Conference*.
- Eric Colon and Kristel Verbiest (2008). 3D Mission Oriented Simulation. Royal Military School.
- Farrington, P.A., Nembhard, H.B., Sturrock, D. T. and Evans, G. W. eds. (1999). Increasing the Power and Value of Manufacturing Simulation Via Collaboration with Other Analytical Tools: A Panel Discussion. *Proceedings of the 1999 Winter Simulation Conference*.
- F.E Cellier. (2006). *Continuous System Simulation*. Argentina: Springer Science Business Media.
- Grajo, E. S., Gunal, A., SathyaDev, D. And Ulgen, O.M. (1994). A Uniform Methodology for Discrete-event and Robotic Simulation. *Proceeding of the Deneb Users Group Meeting*. Deneb Robotic, Inc. 17-24.
- Kin-Hua Low. (2008) *Industrial Robotics: Programming, Simulation And Applications*: Germany, Advanced Robotics Systems International.
- Michel, O/ Cyberbotics Ltd (2004). *Webot: Professional Mobile Simulation Robot*. *International Journal of Advance Robotic System*. Volume 1, Number 1.
- Mohd Johari, N. A. and Haron, H. ( ), *Robotic Modeling and Simulation of Palletizer Robot Using Workspace5*, Master Thesis, Universiti Teknologi Malaysia
- Mohd Salih, N. H. (2008), Modelling and Simulation of Adept Viper S650 (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Muhammad Ikhwan Jambak, Habibollah Haron, Dewi Nasien. (2008) Development of Robot Simulation Software For Five Joints Mitsubishi RV-2AJ Robot Using MATLAB/Simulink And V-Realm Builder. *Fifth International Conference on Computer Graphics, Imaging And Visualization*.

- Muhammad Noor, N. F. (2005), Mitsubishi RV-2AJ Robot Arm Basic Movement Simulation Using Workspace 5 Software (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Nathan Smith, Cristopher Egert, Elisabeth Cuddihy, Deborah Walters (2006). Implementing Virtual Robots in Java3D Using a Subsumption Architecture. Proceedings from the Association for the Advancement of Computing in Education.
- Nepal, R., and Baral, M. (2004). Simulation of Kawasaki 06L Robot in Workspace 5.0. Bachelor Thesis. St. Cloud State University.
- R.D. Kriz, D. Farkas, A.A. Ray, J.T. Kelso, and R.E. Flanery, Jr. ( ), Visual Interpretation and Analysis of HPC Nanostructure Models using Shared Virtual Environments, Conference Proceedings, High Performance Computing: Grand Challenges in Computer Simulations 2003, The Society for Modeling and Simulation International (SCS), San Diego, California.
- Robinson, P. (1996). Robotics Education and Training: A Strategy for Development. *Industrial Robot*. 23(2): 4-6.
- Robotic Simulation (2006), KUKA Robotic Corporation.
- Roth. N. (1999). *The International Journal of Robotics Research*. On the Kinematic Analysis of Robotic Mechanisms. 18(12): 1147-1160.
- Shafei, S. A. (2008), Modelling and Simulation SCORBOT-ER 4u Robot Arm Movement Using Workspace5 (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Shannon, Robert E. (1998). Introduction to the art and science of simulation, Proceedings of the 1998 Winter Simulation Conference.
- Sukimin, Z. (2007) Design, Visualization and Simulation of Automatic Chopping Process (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Zainal Abidin, M. A. (2008), Modelling and Simulation Kawasaki FS03N Robot Arm Movement Using Workspace5 (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Zomaya, A. Y. (1992). Modeling and Simulation of Robot Manipulators: A Parallel Processing Approach. Singapore: World Scientific Publishing Co. Pte. Ltd.

IntechOpen



## **Robot Manipulators Trends and Development**

Edited by Agustin Jimenez and Basil M Al Hadithi

ISBN 978-953-307-073-5

Hard cover, 666 pages

**Publisher** InTech

**Published online** 01, March, 2010

**Published in print edition** March, 2010

This book presents the most recent research advances in robot manipulators. It offers a complete survey to the kinematic and dynamic modelling, simulation, computer vision, software engineering, optimization and design of control algorithms applied for robotic systems. It is devoted for a large scale of applications, such as manufacturing, manipulation, medicine and automation. Several control methods are included such as optimal, adaptive, robust, force, fuzzy and neural network control strategies. The trajectory planning is discussed in details for point-to-point and path motions control. The results in obtained in this book are expected to be of great interest for researchers, engineers, scientists and students, in engineering studies and industrial sectors related to robot modelling, design, control, and application. The book also details theoretical, mathematical and practical requirements for mathematicians and control engineers. It surveys recent techniques in modelling, computer simulation and implementation of advanced and intelligent controllers.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Muhammad Ikhwan Jambak, Habibollah Haron, Helmee Ibrahim and Norhazlan Abd Hamid (2010). Robotil Modelling and Simulation: Theory and Application, Robot Manipulators Trends and Development, Agustin Jimenez and Basil M Al Hadithi (Ed.), ISBN: 978-953-307-073-5, InTech, Available from: <http://www.intechopen.com/books/robot-manipulators-trends-and-development/robotic-modelling-and-simulation-theory-and-application>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen