**CHAPTER 1**

**INTRODUCTION**

**1.1 Problem statements**

Employment of robots in manufacturing has been a value-added entity in a manufacturing industry. Robotic simulation is used to visualize entire robotic application system, to simulate the movement of robot arm incorporated with components consist in its environment and to detect collision between the robot and components. This paper presents result of a project in implementing a computer based model to simulate Okura A1600 palletizer robot. The application uses Okura A1600 robot for palletizing bags at the end of the production line and focuses on pick-and-place application. The project objective is to generate a computer simulated model to represent the actual robot model and its environment. The project simulates the robot’s first four joints, namely as the Waist, Shoulder, Elbow and Waist and focuses on the position of the robot’s end effector, regardless its orientation. Development of the model is using Workspace5 as a simulation tool. Two types of methodology are used, which are the methodology for developing the robotic work cell simulation model and the methodology for executing the robotic simulation. The output of the project will be a three-dimensional view of robot arm movement based on series of predefined Geometry Points, layout checking and robot’s reachability by generating working envelope, collision and near miss detection, and monitoring on the cycle time upon completing a task. The project is an offline programming and no robot language is generated.

**1.2 Computer graphics**

Graphics provides one of the most natural means of communicating with a computer, since our highly developed 2D and 3D pattern recognition abilities allow us to perceive and process pictorial data rapidly and efficiently. Interactive computer graphics is the most important means of producing pictures since the invention of photography and television. It has the added advantages that, with the computer, we can make pictures not only of concrete real world objects but also of abstract, synthetic objects, such as mathematical surfaces and of data that have no inherent geometry, such as survey results.

**1.3 OpenGL**

OpenGL (Open Graphics Library) is a standard specification defining a cross language cross platform API for writing applications that produce 2D and 3D computer graphics. The interface consists of over 250 different function calls which can be used to draw complex 3D scenes from simple primitives. OpenGL was developed by Silicon Graphics Inc. (SGI) in 1992 and is widely used in CAD, virtual reality, scientific visualization, information visualization and flight simulation. It is also used in video games, where it competes with direct 3D on Microsoft Windows Platforms. OpenGL is managed by the nonprofit technology consortium, the Khronos group Inc.

OpenGL serves two main purposes:

* To hide the complexities of interfacing with different 3D accelerators, by presenting programmer with a single, uniform API
* To hide the differing capabilities of hardware platforms, by requiring that all implementations support the full OpenGL feature set.

**Chapter 2**

**ANALYSIS AND DESIGN**

**2.1 System analysis**

The methodology consists of eight phases but the project only executes up to seven phases:

Phase 1: define the problem

Problem identification is defined during the preliminary analysis of problems background. Current system have no computer based model that represent the robotic application. Thus, it is impossible to monitor and evalute the performance of the robotic palletizing system.

Phase 2: Design the study

The study is limited to the project scopes. This phases acquire appropriate decisions for tools and methodology that are going to be used. Besides, a proper planning and milestone need to be done.

Phase 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 work cell development. These data include layout of the robotic applications, geometry configuration of the robot, robot motion parameters and the robot cycle time.

Phase 4: Formulate inputs, assumptions, and process definition

Modeling the robot application focuses on three activities, which are building the robot, motion path programming of the palletizing process and run the simulation. Building the robot’s model is heavily based on the geometrical data of Okura A1600 using the CAD features of Workspace5. The dimension is refers to the CAD drawing of Okura A1600.

Few spatial data needs to be considered in determining the motion path:

Point at the pickup station where the robot will pick the bag.

Five points that represent an arrangement of bags in an odd numbered layer.

Five points that represent an arrangement of bags in an even numbered layer.

Position of points is in x, y and z coordinate. Since each pallet consists of ten layers of bags with five bags on each layer, ten incremental value of z are needed.

Phase 5: Build verify, and validate the simulation model

During this phase, development of the robotic work cell is based on the methodology proposed by change [5]. This is an active phases which aim to improve the model’s precision and motion.

Validation towards the model is based on the visualization of system layout and robot’s cycle time in completing a task. The layout generated using workspace 5 is compared to the actual system layout .

During preliminary data gathering, a movie that shows the actual robot performing a task in one day operation is recorded. The model is assumed to represent the actual system once operated at the same movement of actual system and capable to perform at the similar cycle time as in the movie.

Phase 6: Experiments with the model and look for opportunities for design of experiments

A simulation is run in order to visualize the arm movement and analysis on collision detection. Execution of the simulation is using the features of workspace5 for simulation. Workspace5 allows layout checking in order to prepare other devices within the robot’s reachability. It also capable to generate working envelope for any 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 work cell. This project is an offline programming. Neither robot language is generated nor implementing at actual work cell.

Phase 7: Documentation and representation

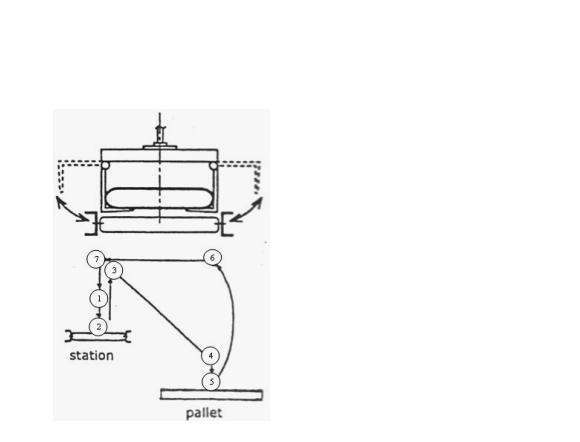
This phase gathers and documents all results generated from the simulation. A written report provide better understanding of the experiment’s executions and analysis.

**2.2 System Design**

This section presents the robot and its application system of the current system. The application uses Okura A1600 robot for palletizing bags at the end of the production line and focuses on pick-and-place application. The robot will pick bags at the pick up point and place the bags onto a pallet. The robot model is multi-articulated four-axis pneumatically powered Okura A1600 Palletizer. It has the fastest cycle times in robot class, which is up to 1600 cycles per hour, with high reliability and low maintenance requirements. The robot is capable to handle capacity up to 140kg. It also provided with high accuracy, which is 1mm of repeatable position precision.

The robot consists of a few components, namely as the Base, Rotor/Link1, Lower/Link2, Upper arm/Link3, Link lever/Link4, Flange/Link5, Motor D-axis1 and Motor D-axis2. Positioning of the robot is based on its Hand Coordinates System. Hand coordinates employ three-dimensional coordinate system based on the three axis (X, Y and Z).

The robotic palletizing system consists of Okura A1600 robot palletizer, conveyor system for transporting bags and pallets, pallet dispenser, guarding that provides boundary for robotic work cell and control panel for supervision and control with human-machine interface. Figure 1 shows the robotic work cell.

The palletizing process follows a standard movement patterns consists of seven steps that represent path of the robot arm moving from the pick-up station towards the pallet and back to the pick-up station. Figure 2 shows the movement pattern. 

**CHAPTER 3**

**SOFTWARE REQUIREMENTS SPECIFICATION**

3.1 Hardware requirements

* Pentium or higher processor.
* 512 MB or more RAM

3.2 Software requirements

This graphics package has been designed for WINDOWS 10 platform and use CODE BLOCKS integrated environment.

Development platform

WINDOWS 10

Development tool

CODE BLOCKS

Language Used In Coding

C++

**CHAPTER 4**

**IMPLEMENTATION**

This section present the Workspace5 as a simulation tool and the methodologies in developing the model and execution of the simulation. Workspace5 is PC-based robot simulation software that supports three-dimensional visualization. Some features of Workspace5 are collision and near miss detection, CAD functionalities that are similar to ACIS graphics kernel as AutoCAD, reach checking for optimization in robot placement, and offline programming.

There are two types of methodology being applied, which are the methodology for developing robotic work cell simulation model by Cheng and the methodology for robotic simulation by Grajo et al. Figure shows the methodology by Cheng and methodology by Grajo et al. is shown in Figure.

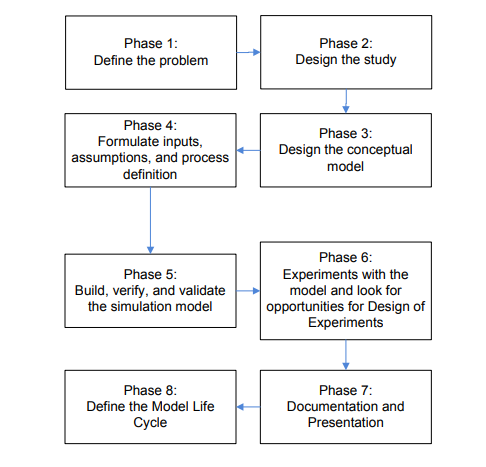
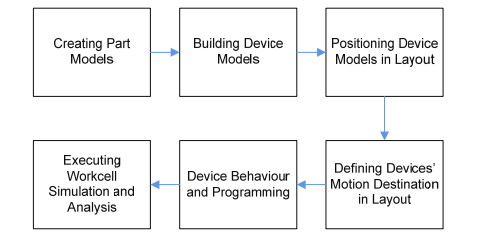


Figure . A methodology for robotic simulation

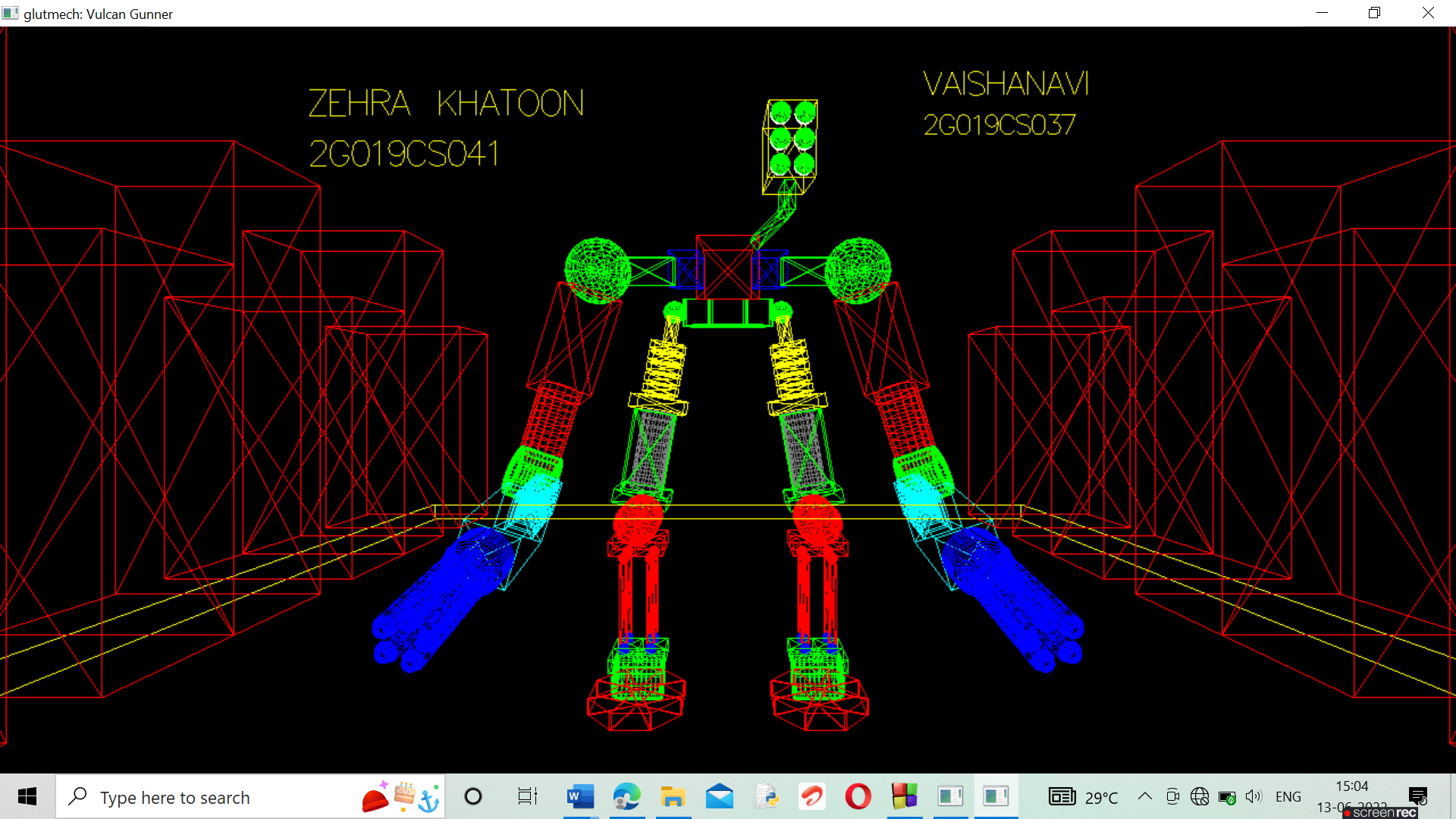


A methodology for developing robotic work cell .

**CHAPTER 5**

**RESULT**

**5.1 SNAP SHOTS**



**CHAPTER 6**

**CONCLUSION**

Robotic simulation is a modeling-based approach for analyzing performances of a robotic work cell. Unlike tradition discrete event simulation, robotic simulation emphasizes on detail study of robot behavior such as motion and interaction among other components in its environment. Robotic simulation provides highly detailed three-dimensional visual layout of a work cell. Some simulation packages are incorporated with collision detection capabilities that allow user to identify any interference between robot and objects. It also provided with cycle time monitoring to verify how long a robot will complete a given task. Programming a robot with minimal cycle time will ensure higher productivity of a production. Compared to DES, robotic simulation packages have a capability to translate a programming made for the simulated model into the real robot programming language.