DEVELOPMENT OF A COMPUTER PROGRAM FOR OFF-LINE PROGRAMMING OF INDUSTRIAL ROBOTS FOR SPRAY PAINTING

A THESIS SUBMITTED TO
THE GRADUAL SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

KORAY ŞEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF MECHANICAL ENGINEERING

FEBRUARY 1997

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Tayfur ÖZTÜRK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ediz PAYKOÇ
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Tuna BALKAN
Co-Supervisor

Prof. Dr. M.A. Sahir ARIKAN
Supervisor

Examining Committee Members

Prof. Dr. Kemal ÖZGÖREN (Chairman)

Prof. Dr. M.A. Sahir ARIKAN (Supervisor)

Assoc. Prof. Dr. Tuna BALKAN (Co-Supervisor)

Prof. Dr. Reşit SOYLU

Prof. Dr. Can ÇOĞUN

ABSTRACT

DEVELOPMENT OF A COMPUTER PROGRAM FOR OFF-LINE PROGRAMMING OF INDUSTRIAL ROBOTS FOR SPRAY PAINTING

Şen, Koray

M.S., Department of Mechanical Engineering

Supervisor : Prof. Dr. M.A. Sahir Arıkan

Co-Supervisor : Assoc. Prof. Dr. Tuna Balkan

February 1997, 111 pages

A computer program is developed for off-line programming of industrial robots for spray painting. The program runs in AutoCAD environment. AutoSurf is used to model the surface which is going to be painted. The surface data and the painting parameters are used to prepare the robot program in programming language of Fanuc Arc Mate Sr. Robot with R-G2 Controller. An interactive program module is developed to input and/or to determine the painting parameters. Robot program is downloaded to the robot for spray painting. A set of experiments are conducted to determine the relationships among the paint flowrate, paint tank pressure and nozzle opening valve position of the spray gun.

Keyword: Spray painting, robot painting, robot programming, surface model for spray painting.

ÖZ

ENDÜSTRİYEL ROBOTLARIN SPREY BOYAMA İÇİN PROGRAMLANMASI AMACIYLA BİR BİLGİSAYAR PROGRAMININ GELİŞTİRİLMESİ

Şen, Koray

Yüksek Lisans Tezi, Makina Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. M. A. Sahir Arıkan

Ortak Tez Yöneticisi : Doç. Dr. Tuna Balkan

Şubat 1997, 111 sayfa

Endüstriyel robotların sprey boyama için programlanması amacıyla bir bilgisayar programı geliştirilmiştir. Geliştirilen yazılım AutoCAD paket programının içerisinde çalışmaktadır. Boyanacak yüzeyin modellenmesi için AutoSurf kullanılmaktadır. Yüzey bilgisi ve boyama parametreleri Fanuc Robot Arc Mate Sr.'ın programlanması için R-G2 Kontrolcüsünün anlayacağı robot programını hazırlamada kullanılır. Boyama parametrelerinin kararlaştırılmasında kullanılmak üzere kullanıcı ile diyalog kuran program modülü geliştirilmiştir. Yaratılan dosya robota yüklenerek çizilmiş olan yüzeyin boyanması sağlanır. Boya akışı, boya tankı basıncı ve boya tabancası nozul açıklığını sağlayan valf arasındaki ilişkileri görmek için bir dizi deney yapılmıştır.

Anahtar Kelime : Sprey boyama, robot boyama, robot programlama, sprey boyamada yüzey modelleme.

ACKNOWLEDGMENTS

I would like to express my thanks and appreciation to Prof. Dr. M. A. Sahir Arıkan for his guidance and constructive suggestions during the preparation of this thesis.

I also would like to express my sincere appreciation to Assoc. Prof. Dr. Tuna Balkan for his guidance and insight throughout the preparation of this thesis.

I would like to thank to the technician Muhlis Baharlı in the Materials Testing Laboratory and to Bülent Önalır for developing robot-PC interface routines.

I thank to my family for their understanding and to my fiancee, Mehtap Tunca, for her great support in completing this thesis.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In this thesis, it is aimed to develop a computer program for off-line programming of industrial robots for spray painting process. The software is developed for low-level programmers. Geometric modeling of surfaces is performed in a professional drawing program.

1.1 Robots in Spray Painting Process

A painting robot is not a general-purpose manipulator called in to spray finish a part. Since it works in a hazardous environment, it must be intrinsically safe. It works while it moves, not while it is stopped, so it needs good turning ability, high accuracy, speed, and fluid motion. It is generally compact so it can fit into a small paint booth that may have been designed for a human worker.

Spray painting process is one of the first processes to be robotized using continuous path control. It is ideal for robotization due to a number of factors :

1. The paint spraying environment and working conditions are unpleasant and dangerous for humans, this is due to: the materials used which are often toxic and necessitate the wearing of protective clothing; the noise produced

from the nozzles as compressed air forces the paint out through them in a fine spray, which requires the operator to wear ear protection; and the working area is often restricted and the amount of manipulation required is often complex and requires relatively great effort.

2. Conventional automated paint sprayers are expensive, relatively inflexible, can be inefficient and are not usually able to fully complete the job especially in complex tasks such as spraying car bodies which requires touch-up by a human operator at a later stage.

3. Due to health and safety regulations energy costs can be high. This is due to the necessity to provide adequate ventilation and extraction of fumes to allow a constant flow of sufficient fresh air to the operator. If humans were removed from the work area then the necessity of energy intensive ventilation systems would be reduced, thus lowering costs.

4. Human operators do not produce work of consistent quality. Besides overpainting often occurs increasing the overall cost of paint.

1.1.1 Type of Spraying

Most robots can use a variety of atomizers that spray paint in the form of a fan. With conventional *air* spraying, air strikes a low-pressure fluid stream at high pressure - about 22 MPa. Air spraying gives a fine finish without an unpleasant orange peel effect. It's the method of choice in a majority of automotive applications.

Airless spraying is a high-pressure hydrostatic method with less atomization, good for heavy-viscosity materials. It's generally associated with a functional finish.

Air-assisted airless falls somewhere in between. It involves spraying the liquid under high pressure (8.3-10.3 MPa) through a small orifice. It has higher transfer efficiency than air, but finishes are less glossy.

Finally, there is high-volume, low-pressure (HVLP) atomization. HVLP offers higher transfer efficiency with nonelectrostatic painting. Because there is not much overspray, very detailed work can be done.

All of these are available in electrostatic and nonelectrostatic forms. Electrostatic application, where the paint is charged to a given polarity, positive or negative, while the part is grounded to the opposite polarity, offers the highest transfer efficiency.

Electrostatic application is a must with powder. Powder must be sprayed slowly, so it is inappropriate for high-velocity applications. Finishes are not as good as with fluids, and the initial investment is high, but the powder can be recirculated.

1.2 On-line and Off-line programming

Today most robots are programmed by a method called on-line programming by which the robot in a manual control mode is moved through a set of geometrical frames defining the given task. The frames are recorded in a memory which a sequencer in the robot controller will read during play back. The method fully relies on the operator's performance and the accuracy of the programmed robot task cannot be better than that of the operator. This can be a serious limitation in robot applications requiring a high accuracy and performance. Besides, during the programming phase, the robot itself and the equipment it is serving are idling in on-line programming.

Modern approaches in robot programming, however, focus on using computers for programming without access to the physical robot when the programming is performed. This is called off-line programming and is more efficient since the robot can be in operational mode while the programming of new tasks is prepared outside the robot and its controller. Also off-line programming allows for the use of advanced computer software for modeling of the robot world, robot task planning, simulation and program verification.

Programming is usually done *on-line*. There are several on-line programming methods. A typical method being used for motion planning of the painting robot is *lead-through real-time*, in which an operator attaches handle to the end of the spray gun and robot arm and literally leads the robot through a path. The robot memorizes the operator's motions and duplicates them every time a similar part comes down the line.

However, on-line programming methods has several drawbacks :

1. The robot cannot be used during the teaching period,
2. A human is exposed to a hostile environment,
3. The motions taught are, at best, at the human's skill level.

An alternative to deal with above problems is off-line robot programming method. Off-line is a programming method in which robot program is created and developed while isolated from the robot so that the robot can continue to perform its current program.

Off-line programming and simulation is a very efficient technique in the process of design and optimization of robot cells and fast evaluation of robot programs. This technique is specially applicable to small batch manufacturing, where programming of robots for different tasks is frequently required.

The benefits of off-line programming are generally known, and include ;

- Reduced downtime, giving greater capacity for robots.
- Better understanding of process through simulation.
- Reduced risk to damage an expensive equipment or injury to human operator.
- Reduced stress and increased productivity of programmers.
- Increased likelihood of process optimization.
- Reduced variety of program languages, giving reduced training costs.

Off-line programming can be broadly divided into two according to the programming method used:

I. Text-based off-line programmers : This group usually employs a high-level robot-programming language in order to create robot programs. Such systems are good for control over the robot, since off-line language is identical to the native language of the programmed robot. The main advantage is that they are cheap and provide good programming facilities. However, they offer little or no visualisation features and they require high-level programmers. Moreover, text based off-line programming systems provide no compatibility with different types of robots. This disadvantage is giving rise to the adoption of second group of off-line programmers; the graphic-based simulation systems.

II. Graphic-based simulation systems : Graphic-based simulators provide three dimensional models of robot cells which can be simulated according to the program. Therefore, they eliminate the problems with visualization. The robot programs are created interactively with the computer model of a robot cell by simple point-and-click operations on different instructions in a menu system. Each of these macro instructions generates a few lines of a built-in programming language, often referred as 'neutral language'. Graphic based systems still provide the possibility of editing the robot programs manually in a text editor, thus giving a facility of more advanced programming. However, such typed input of program requires a detailed knowledge of the neutral language itself. The programs created are then post processed into target native language of the given robot and downloaded to its controller.

The main benefit of these systems is the increased possibility of getting closer to an optimum solution in workspace layout and program design since the user can visualize the results of simulation before committing the program to be post processed. In addition, they provide an advanced user interface. Examples of such systems are listed in Table 1.1.

The price to pay for these advanced features is the total cost of the system which is in some cases exceeds the actual cost of the robot and the its workcell. Therefore, these systems are profitable only for large industries such as automotive, aerospace or electronics, where the total number of robot installations is high and product variety is great.

Table 1.1 Examples of graphic-based simulation systems.

Products	Company	Operating System	Hardware
Grasp	BYG Systems Ltd.	Unix	Vax, Apollo
Igrip	Deneb Robotics Inc.	Unix	Silicon Graphix IRIS
RobCad	RobCad Ltd.	Unix	Workstation
Robographix	Computervision	N/A	Mainframe
Workspace	Robot Simulations Ltd.	MS-DOS	IBM/PC compatible

1.3 Survey of Previous Literature

S. Suh, I. Woo, and S. Noh [1] developed an Automatic Trajectory Planning System (ATPS) for painting robots by a new scheme. This scheme considers geometric modeling, painting mechanics, and robot dynamics to output an optimal trajectory based on the CAD data describing the shape of the objects. ATPS allows a user to determine the optimal painting parameters iteratively, by generating spray gun path and robot trajectories, and displaying the distribution of the coating thickness and painting time.

G. Duelen, H.D. Stahlmann, and X. Liu [2] developed a computer aided planning system for programming NC coating machines. This system is based on a mathematical model computing the repartition of the coats on the sculptured surfaces. The mathematical model describes the relation between the lacquer distribution in the spraying cones and the thickness of coat on the surface. Computing the development of the thickness of layer, the system defines the optimal trajectories of the spray gun using the geometrical description of the surfaces to be coated from a CAD module. Furthermore, the mathematical model is able to fix the speed of the gun motion and the lacquer flowrate.

Phuong-Anh P. Ngo, G. Dale Cheever, Robert A. Ottoviani and Tadeusz Malinski [3] developed a model involving surface and coating parameters to relate their effects on the formation of the paint/air interface. The results showed that initial roughness, film thickness, and viscosity of the resin have the most influence on the roughness of the topcoat.

De Vilbiss [4] which is a surface-coating company introduced an electrically-driven robot the GMF P-100 for paint spraying in 1989. The robot can be programmed by the continuous-path lead-through method, not available with other electric-drive robots, as well as the lead-through point-to-point and power-teach point-to-point methods. One of the P-100's features that industry will find it useful is its reachability. It is claimed to be the largest for spray painting robots. Its envelope extends to 2 m in front of the arm base and it can reach over 2.5 m in height.

Deneb and Technomatix [5] offer robotic painting modules based on their simulation software. These softwares can generate paint paths, maintain orientation and distance of the paint gun in relation to the part surface, and set runtime simulation parameters. They also display overlap, paint gun velocity, ongoing transfer efficiency, and thickness information in different colors and granularities.

Exterior wall painting robot (SB-Multi Coater), developed by T. Ueno,Y. Kajioka, H. Sato and T. Yoshida and R. Tsutsui [6] provides a rapid and high quality painting. The robot's work speed is five times faster than that of human workers and spray quality is more uniform than that achieved using manual applications. The robot has a traveling speed of 3 m/min and uses automatic gun for spraying material. The operation process of this robot consists of loading material, selecting traveling pattern, determining coating width, lifting robot to top of a wall, and starting automatic coating.

1.4 Objective and Scope of the Study

In theory, the most interesting characteristic of a robot is its ability to execute an infinite number of different tasks. This versality comes from the generality in its mechanical and kinematic structure, its modes of control and its programmability. The full development of a robot system however requires cost effectiveness in the investment phase as well as in the operation stage.

One of the bottlenecks for the full exploitation of a robot system is the programming of the robot. To program a robot means to derive a set of robot actions necessary for carrying out a desired task and to express these in a language format and code so that they can be read by the given controller and finally be executed by the given robot driven end-effector.

For an industrial robot of the present-day technology, the cost of the robot programming for a given set of applications may be comparable with the cost of the robot itself. This is due to the fact that robot programming is still difficult since robot task has to be executed in the physical world by real-time control. Hence there occurs a need for development of softwares which are more efficient and less costly robot programming systems.

In industrial robots, necessary end-effectors, peripheral devices and softwares provided by the supplier change according to the application to be performed. However, not only these equipments, but also the softwares are very expensive. Even in some cases, the price of the software can be higher than the price of the robot itself. Hence, any volunteer who decides to buy a robot should consider much, to make the correct selection of the software to be used by the robot controller.

In some cases, not one but a few softwares may be required due to the change of production line in factory or, if robot placed in a laboratory, the different researches to be performed, one way to avoid from so much expense is to make your own software.

In this thesis, the study is carried on the Fanuc Arc-Mate Sr. Arc Welding Robot within a Fanuc R-G2 Controller environment. Lead-through real-time method is not applicable to the Fanuc Robot due to lack of the handle which is attached to the spray gun. However, on-line programming for spray painting process is a difficult programming method. Because keeping spray gun perpendicular to the surface at a constant distance is not an easy work even for an experienced worker. In addition, to reach an optimum solution requires trial and error causing less efficiency. On the other hand, programming for spray painting process can be written easily by using a geometric modeling program for the modeling of the surface to be painted and a post processor to construct the robot program from the surface data supplied by this modeler.

For the geometric modeling of the workpiece, AutoCAD Release 12 environment with AutoSurf Release 2 surface modeling extension is used. AutoCAD is chosen due to its relatively cheap price and its widespread usage around the world. Besides, AutoCAD can be programmed by using its own programming language called AutoLISP. This peculiarity allows developing the spray painting software and the post processor in the AutoCAD environment.

Spray painting apparatus mounted on the Fanuc Robot consists of Binks automatic spray gun, pressurized paint tank, solenoid valve and oil and water extractor. This configuration supplies HVLP type of painting. However, calibration for the paint flowrate is required for the apparatus.

In this thesis, it is aimed to develop a software for spray painting process of any surface created in AutoCAD environment. The post processing is made according to the painting condition decided in the AutoCAD environment. The output of the software is a robot program text file which is ready for downloading to the Fanuc Robot.

CHAPTER 2

PAINTING MECHANICS

2.1 Painting Mechanics[1]

In air spray painting, it is important to keep the spray gun normal to the surface, at a fixed distance. Besides, the density of points must be kept uniform for quality coating. Having these conditions satisfied, uniform coating thickness and good finish are obtained.

Let the spray radius and spray area be respectively r, A. Then the following equation holds: $A = \pi r^2$. Let the flowrate at the nozzle, coating thickness, the ratio of the paint which is coated on the surface over paint flowrate at the nozzle, and infinitesimal spraying time be respectively μ, δ, σ, and t (Figure 2.1). Then, the paint amount on the surface is $Q = \mu \sigma t$. Assuming uniform dispersion, since $\mu \sigma t = A \delta$, the coating thickness, δ is

$$\delta = \mu \sigma t / \pi r^2 \qquad (1)$$

Assumptions in developing the above equations are the spray area is circle and the instantaneous coating thickness is uniform. However, as the spray gun moves, the coating thickness of the boundary becomes thinner than the center (Figure

2.2). The cross-section of coating will be a semiellipse. The maximum height of that cross-section is δ and the minimum height is zero at the boundary. Then, the average thickness δ_m is,

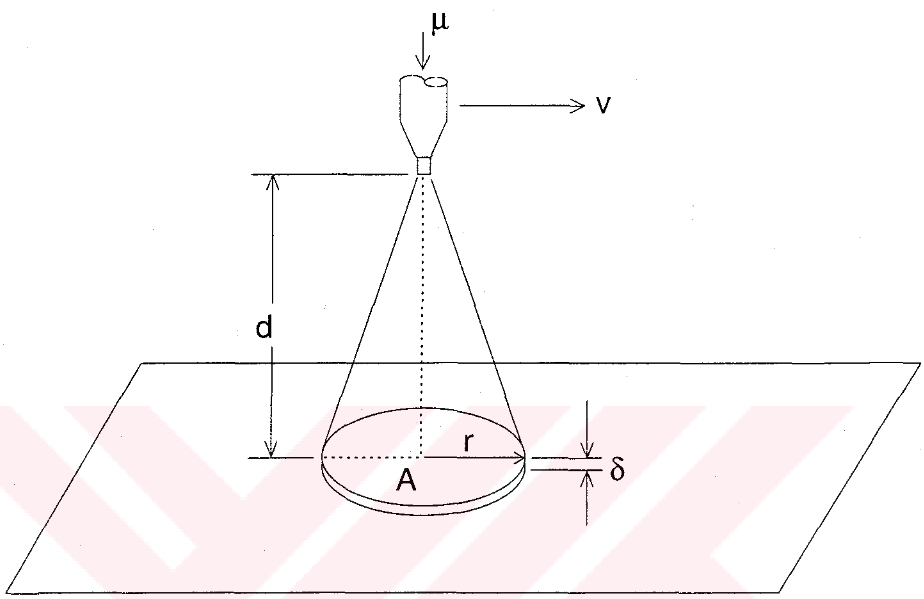


Figure 2.1 Painting parameters.

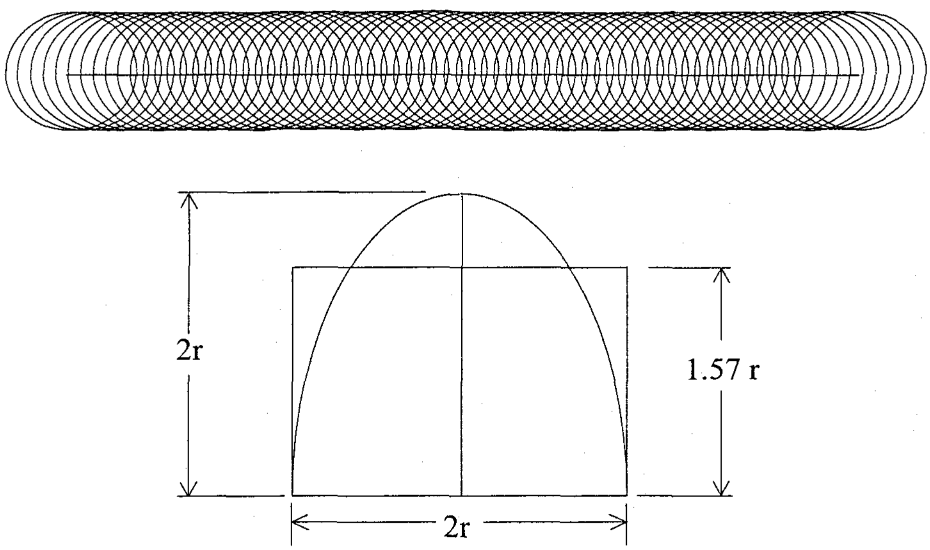


Figure 2.2 Distribution of painting thickness.

$$\delta_m = (\pi / 4)\, \delta \tag{2}$$

Let v be the painting velocity, then the time interval for which a spot is covered is 2r/v. From equations (1) and (2), the average coating thickness with the velocity of v is obtained by substituting t=2r/v. In other words, for a given painting condition μ, σ, and r, the painting velocity to obtain a coating thickness of δ_m is,

$$v = \mu\, \sigma / \; 2\, \delta_m\, r \tag{3}$$

If step increment between two paths is considered, it is logical that it should be between 0 and 2r. Recommended value for step increment is r (Figure 2.3). If step increment between two paths is taken as spray radius, r, the spray circle is overlapped and the average coating thickness is doubled, except at the boundary of the surface. Therefore, to keep the coating thickness same, the velocity, which is the sweeping speed of the surface, must be doubled.

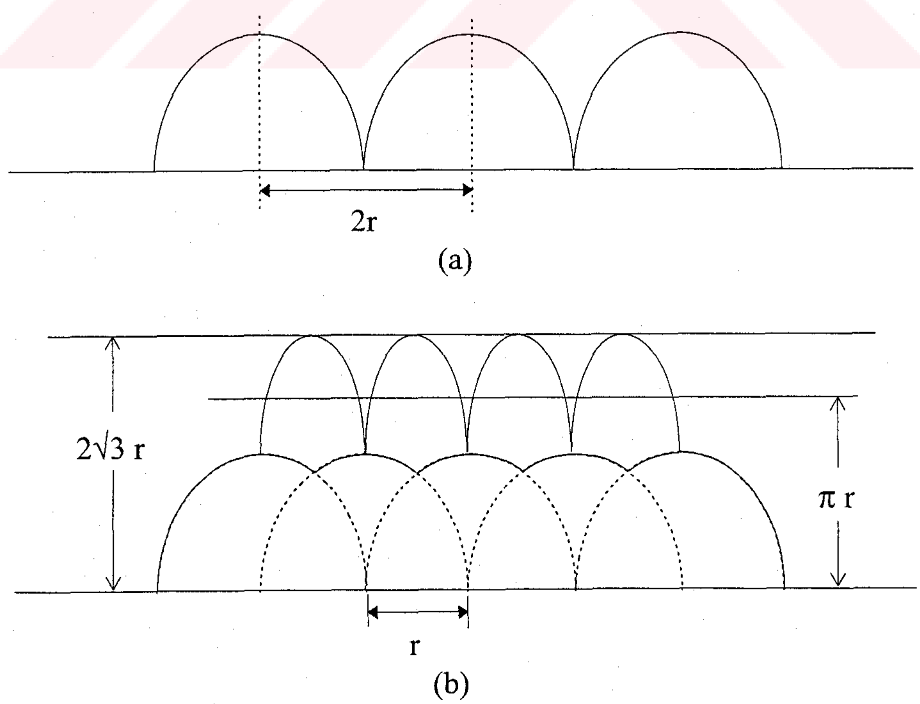


Figure 2.3 Step Increments.

$$v = \mu \, \sigma / \delta_m \, r \quad (4)$$

This velocity is the sweeping speed of the surface and should not be considered as that of the spray gun. These two speeds are the same for plane coating but considerably different for free surface coating. The above mechanism can be used not only for plane coating but also for free surface coating, as a free surface can be decomposed into a finite number of planes.

CHAPTER 3

SURFACE MODELING FOR SPRAY PAINTING

3.1 Introduction

AutoCAD is a well-known professional drawing program. It has several advantages against other drawing programs such as: It is relatively cheap, easy to use, widespread, and more importantly it is programmable.

AutoCAD package program can be equipped with other auxiliary packages according to the need of the company. Therefore, it costs less. There are a few auxiliary packages like AME (Advanced Modeling Extension), AutoVision, AutoSurf as a complementary to it.

It can be easily used with the help of its pop-up menus. In addition, it can be programmed to satisfy our needs in the most efficient way. It has an application interface language of its own, called AutoLISP.

AutoLISP lets the user to write macro programs and functions in a powerful high-level language that is well suited to graphics applications. Also, it is easy to learn and use, and is very flexible.

3.2 *Auxiliary Package : AutoSurf*

In AutoCAD, objects in all three dimensions can be modeled using entities such as polylines and splines. The result of combining these entities in 3D is referred to as a *wireframe model.* Wireframe models has many advantages over 2D representations of the same object. 3D wireframe models:

- Provide a visual environment for quick resolution of design problems and ambiguities.
- Let you extract certain information, such as spatial relationships between various features, more directly than from a multiview drawing.
- Let you output the design views as drawings.

Another technique called *surface modeling* offers additional benefits over wireframe methods. In wireframe models, the Computer Aided Design (CAD) system has limited knowledge about the implied surface between wireframe elements, while in surface modeling, objects defined by their surfaces yield more information. In surface modeling, the shell of an object for a more easily perceived view can be defined, whereas in wireframe modeling, even exact cross-sectional views of the object can not be produced.

A completely surfaced model is suitable for many engineering and manufacturing purposes, such as:

- Accurate section generation for engineering and packaging studies.
- Surface inputs for Finite Element Modeling (FEM) and Finite Element Analysis (FEA).
- Shade renderings for visualisation and presentation.
- Surface inputs for Rapid Prototyping equipment.

- Rotated surfaces for tool, mold, and die design.
- Surfaces for Numerical Control (NC) machining of models and tools.

From that point of view, AutoSurf is a high-end, free-form, curve and surface modeling system within the AutoCAD ADS environment that supports both conceptual design and manufacturing applications.

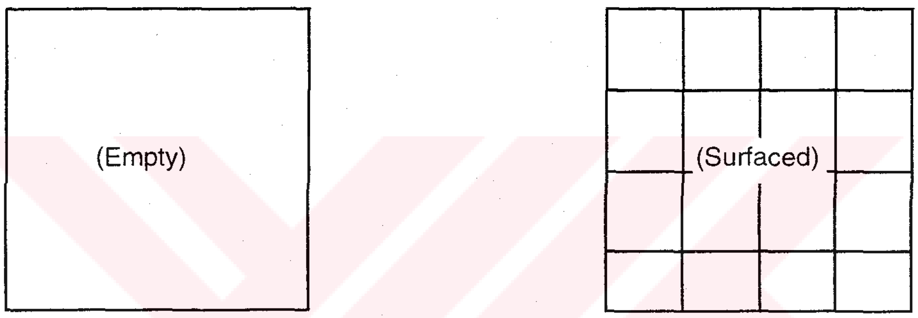
3.2.1 What Surfaces mean in AutoSurf

In AutoSurf, a surface is a mathematical equation that represents a contour or shape within a three-dimensional area. In other words, a Surface is an entity that has length and width, but essentially no thickness. A wireframe entity in 3D has no interior. For example, four lines that form a rectangle represent only the outer edge of the rectangle. By creating a surface between four lines, interior area is filled with something of substance - a Surface as shown in Figure 3.1. Created entity has a *front* and *back*. This is a key concept in surface modeling.

AutoSurf bases all surfaces on NURBS (Non-Uniform Rational B-Spline) mathematics. This is one of the most advanced forms of surface mathematics for holding curve and surface data. NURBS mathematics is used because it is capable of accurately describing the most common free-form surface types, such as *Coons Patch, Bezier* and *B-Spline*, and also exactly modeling geometric (primitive) surfaces.

There are 16 methods to create a surface in AutoSurf, although all Surfaces are NURBS-based. These methods are collected in three general surface categories: *primitive, free-form* and *derived*. An additional category, *trimmed*, is applied to all 16 surface types.

Primitive surface types have a constant geometric shape and typically include patterns such as spheres and cones, whereas free-form Surfaces follow no set geometric pattern, and often constructed from arbitrarily shaped sets of lines and curves. Therefore, free-form Surfaces are used to construct extremely complex Surfaces. There are five primitive surface types in AutoSurf, which are *Cone, Cylinder, Sphere, Torus,* and *Planar*. The seven AutoSurf free-form surface types are *Extruded, Revolved, Ruled, Lofted, Meshed, Swept*, and *Tubular*.



Wireframe

Polylines indicate extends of an empty area. The internal area is not defined.

Surface

Polylines indicate boundaries and internal shape of an area that is smoothly defined by a NURBS (Non-Uniform Rational B-Spline) Surface

Figure 3.1 Surfaced Rectangle.

Derived Surfaces are NURBS Surfaces that is generated from existing surfaces. A fillet surface between two existing surface is an example of derived surface. The four AutoSurf derived Surfaces are *Blended, Fillet, Corner*, and *Offset*.

Most of the NURBS Surfaces constructed have four smooth (regular) boundaries. However, if a Surface from irregular boundaries is constructed, the resulting NURBS Surface will not be smooth and can behave unpredictably (That is the main reason of unavailability of augmented lines with trimmed Surfaces). Because many designs include areas with irregular boundaries, AutoSurf uses *trimmed Surfaces* to create Surfaces with irregular boundaries.

Trimmed Surfaces is used to impose any irregular shaped Polyline on top of an existing smooth, four-sided NURBS Surface, thereby redefining the *borders* of the NURBS Surface. After Surface is trimmed, the border becomes the integral part of the surface definition.

3.2.2 Creating Surfaces

Surfaces can be constructed from Polylines, Splines, and Arcs. All these three input types are interchangeable from one to another.

Primitive surfaces (often called geometric surfaces) have a constant geometric shape, and typically include patterns like spheres and cones. See Table A.1 in Appendix A for primitive surface types and the commands to create them.

In contrast to primitive Surfaces, free-form Surfaces follow no set geometric pattern, and often constructed from arbitrarily shaped lines and curves. More complex Surfaces can be constructed by using free-form Surfaces.

Extruded and Revolved Surfaces are similar to primitive surface types; they can be used to create components that are added to other Surfaces.

A ruled surface can be used for surfacing wireframe designs. Often sheet metal and plastic designs include flanges for mating components. Ruled Surfaces offer a simple and quick method to surface flanges of all lengths and widths.

Swept Surfaces are one of the most commonly used free-form Surface types. A Swept Surface is used to drag a constant cross section along one or two rail curves. For example, if a sun roof opening with all of the rubber sealing surfaces is desired, first cross section representing the rubber seal is designed and then the section is dragged around the roof opening line.

Lofted and Meshed Surfaces are used when large areas defined by many sections are converted into a single smooth surface. These sets of sections originate from digitized 2D drawings and prints or scanned 3D clay models. A Lofted Surface should be used when the input is a single set of lines, and a Meshed Surface should be used when the input is two sets of lines like sections and cross-sections.

Tubular Surfaces are used in packaging design, where a surface around the center lines defining a pipe or wire is quickly rapped. Besides, Tubular Surfaces are intersected with adjacent components for interference studies. See Table A.2 in Appendix A for free-form surface types and the commands to create them.

Derived Surfaces are also free-form Surfaces. Instead of Polylines and splines as input, AutoSurf builds derived Surfaces from *existing* NURBS Surfaces. In the natural sequence of modeling or surfacing wireframes, derived surfaces are the last operation, since they are rely on the existence of other Surfaces.

A Blended Surface is used to complete the final surfaced model where holes still exist in the surfaced model. To smooth sharp edges Fillet Surface can be used and a Corner Surface can be used to cap areas where fillets intersect. Offset surface is used when a new surface that is offset from the original Surface is to be created. See Table A.3 in Appendix A for commands and the definitions.

3.2.3 Editing Surfaces

A description of six AutoSurf Surface editing commands follows.

BREAKSF	:Breaks a selected Surface into two or four new Surfaces along specified U or V lines (see Figures 3.2 and 3.3), or C0 or C1 continuity change (see Figure 3.4).
EXTENDSF	:Extends a selected Surface from a boundary by a specific distance or percentage.
NORMALSF	:Allows reversing the positive normal side of a Surface.
JOINSF	:Joins two or more adjacent untrimmed Surfaces into a new continuous Surface.
REFINESF	:Regenerates a selected Surface to a lower tolerance, reducing the number of internal patches. Also bicubic, non-rational Surfaces can be created by this command.
TRUNCSF	:Truncates a selected slab surface to the minimum size necessary to support its trimmed boundary.

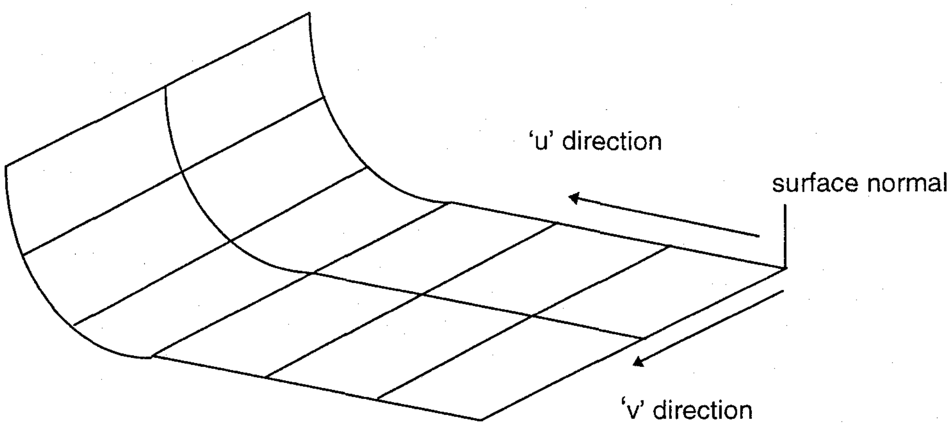


Figure 3.2 Surface with 'u' and 'v' direction surface lines and surface normal

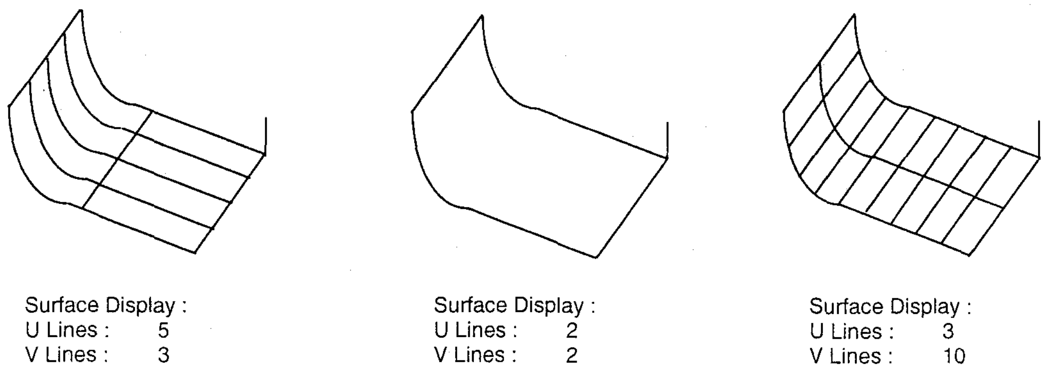
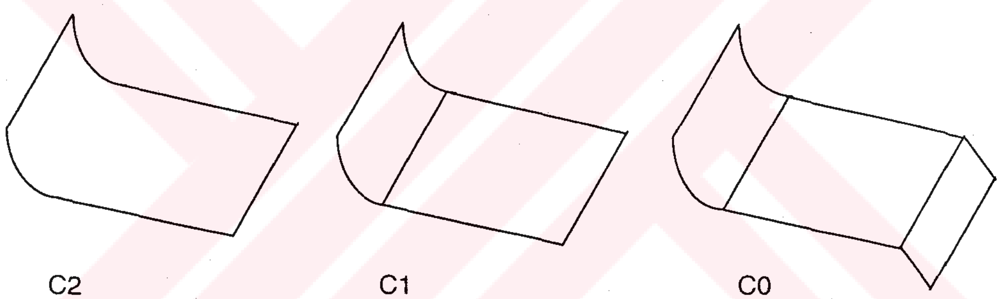


Figure 3.3 Surface display with different setting for the number of U and V lines.



C2 continuity - continuous smoothness (default)
C1 continuity - curvature change at tangents
C0 continuity - abrupt corners

Figure 3.4 Surface continuity.

The shape of the NURBS Surface or spline is directly controlled by the position of its *control points*. Normally, AutoSurf doesn't display control points. However, when the control points are turned on, they can be pulled to change the shape of the Surface or Spline. To dynamically edit Surfaces or Splines, Editsf and Editsp commands are used.

EDITSF :Used to display and move Surface control points.

EDITSP :Used to change the shape of a NURBS Spline by manipulating control points or the actual Points on the Spline.

3.2.4 Trimming Surfaces

Trimmed Surfaces is used to impose any irregular shaped Polyline on top of an existing smooth, four-sided NURBS Surface, and redefine the borders of the NURBS Surface for all subsequent operations.

The following AutoSurf commands create trimmed surface automatically as a secondary operation. AutoSurf keeps the selected part of the Surface.

INTERSF :Finds the intersection of two selected Surfaces and automatically trims one or both Surfaces to their intersection line.

PROJECTSF :Projects entities onto a Surface, and automatically trims the Surface to the shape of the Polyline and keeps the inside or outside shape.

FILLETSF :Creates fillets at the edge formed by two intersecting Surfaces, and automatically trims the Surface to the tangent of the fillet formed by the Surfaces.

CORNERSF :Creates a fillet in the corner formed by the three intersecting Surfaces, and automatically trims the Surface to the tangent of the corner fillet formed by the three Surfaces.

PLANESF :Surfaces a closed Polyline shape that lies in a plane, and automatically creates a trimmed surface.

3.3 Spray Gun Path Planning

Spray gun path planning is composed of three steps, in terms of determining sweeping direction, path interval planning, and determining intra path.

There are two sweeping directions for a given geometry of the surface, as u-surface lines and v-surface lines (Figure 3.5). Determination of the sweeping direction is somewhat arbitrary. It is generally chosen as the one with the small curvature value. The reason is that the path with the less curvature is likely to result in a desired robot motion than the path with larger curvature.

In other words, in which direction surface lines closes to the straight line, that direction should be the spray painting direction. Straight lines can be followed more accurately than the curved surfaces by the robots. In addition to the smooth robot motion, energy consumption is decreased and uniform coating is supplied while painting through less curved surface lines.

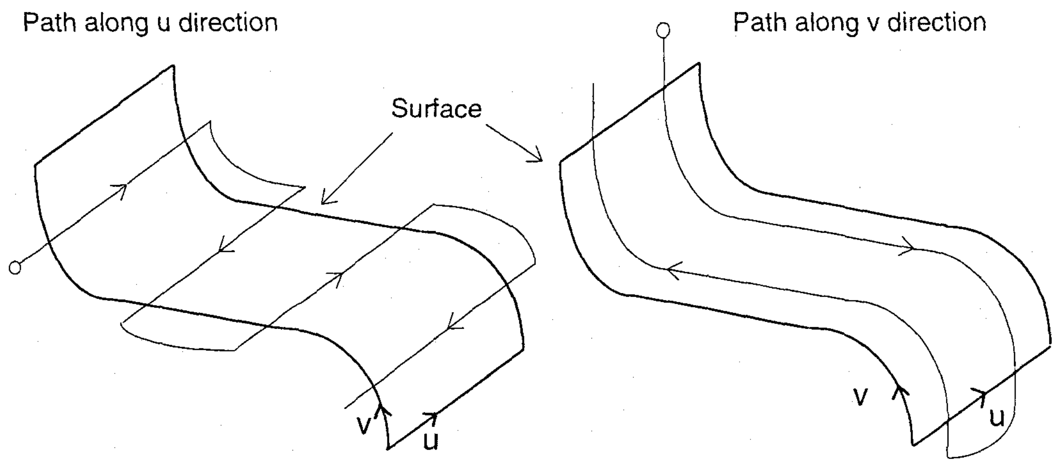


Figure 3.5 Sweeping Direction.

Path interval is the distance between two sequential path lines. It is equal to spray radius in spray painting process. Since, spray radius is dependent on painting variables, in terms of cone angle, spray distance, fluid tank pressure and spray gun's valve position, path interval is set after these variables are decided. This is done by changing surface lines number in the direction of painting for planar and cylindrical surfaces. For complex surfaces, sectioning is applied to the surface with step increment by an amount of spray radius value.

Intra path planning is to find the center of the spray gun's position on the surface within the path interval. Since the position (P_{tip}) and orientation (O_{tip}) of the spray gun is determined by,

$$P_{tip} = P_{surface} + d\, N_E \tag{5}$$

$$O_{tip} = -\, N_E \tag{6}$$

where $P_{surface}$ is the point on the surface, d is the spray distance and N_E is the unit normal vector at $P_{surface}$.

3.4 Extracting Surface Data For Spray Painting

Augmented Lines (ALs) are lines that have surface normal at every point. They have many manufacturing uses, for example 5-axis laser cutting, welding and spray painting processes. In other words, they are the most important subject if any machining operation is in consideration. With the help of these lines, surface data is extracted and they can be processed according to the desired work to be performed and the robot controller to be used.

ALs are created by CREATEAUG command. They should be selected from 'surface edge', which is the line decided as sweeping direction, for surface, since 'wireframe' option doesn't supply a detailed surface information. After selection, to decide the number of augmented vectors 'Point', 'Step' or 'Optimal' options can be selected. In spray painting process, for better surface coating, it is required to have these vectors in such a way that it will describe the surface curvature best. This is important because robot will move through these points linearly. That is to say that if vectors are placed far away from each other, possibly constant spray radius and constant spray distance will not be achieved in-between the two consecutive normal vectors for any complex surface. This is achieved by controlling *system tolerance* value.

AutoSurf stores Splines and Surfaces as infinitely smooth. However, it requires the desired accuracy while computing a new entity. The accuracy of the all AutoSurf outputs is controlled with system tolerance setting. The value used for system tolerance is called the *maximum chordal deviation*. This is the maximum perpendicular distance that line segments of a Polyline can deviate from a perfect internal mathematical representation of a Spline or Surface as shown in Figure 3.6. By setting this limit, how many points AutoSurf uses to calculate Polylines that result from any action relative to Surfaces are controlled effectively.

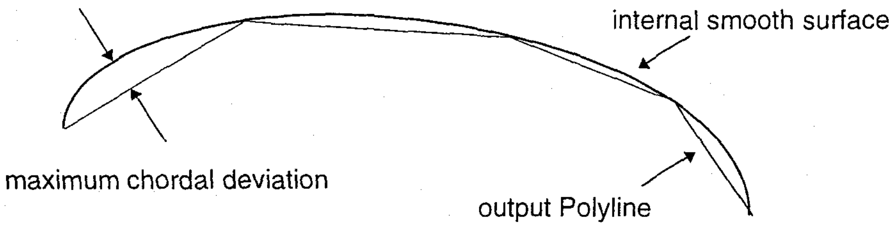


Figure 3.6 Maximum Chordal Deviation.

If 'Point' option is selected, number should be given according to the curvatures along the surface and the length of an augmented line. Point are equally spaced along the contour of the selected edge. If 'Step' is selected, keeping the step distance small always avoid from deviation from the spray gun path. Step distance is the distance between the point on the augmented line along the contour of the selected edge.

However, it is better to use 'Optimal' option. Because, by using this option, normal vectors are placed according to the *system tolerance* value. 'Optimal' defines the minimum number of points on the Augmented line to be the number of points that are needed to keep the surface edge within the surface tolerance. Hence, surface is described better. Besides, the number of normal vectors can be controlled easily by setting system tolerance value.

ALs are created normal to the surface by default. Therefore, there is no need to invoke advanced options, unless offset is desired.

After having ALs created, surface data can be saved to a file by OUTPUTAUG command. This command shows a dialog box after the selection of ALs, to enter output filename. Surface data are written to an ASCII file in the order ALs were selected.

CHAPTER 4

PROGRAMMING THE FANUC ROBOT FOR SPRAY PAINTING

4.1 Programming Fanuc Robot Arc Mate Sr. with R-G2 Controller

Fanuc Robot Arc Mate Sr. is a six-axes revolute joint manipulator. It has two offsets between links (1 - 2) and (5 - 6). Because of these offsets, to derive analytical solution for inverse kinematics of Fanuc Robot is very difficult. However, it can be programmed in w*ork coordinate system* defined by the user.

The information used to control the operation of the robot is called as robot command data. This command data is in the form of a program which is a detailed set of instructions that designate the movement and the functions of the robot required to accomplish a specific job. The program is stored in the data memory of the robot.

Each program consists of a series of steps (blocks) in which the type of movement, speed, location, program sequence, and operations involving end effector and peripheral devices are specified.

Each program has a unique program number identified with the letter O and a four-digit number. Similarly, to specify program sequence, each block has a block number identified with the letter N and a four-digit number. The block is the basic element of robot command data. This block of data includes information which defines a particular point in the robot's path of motion. Blocks are composed of G, F, and S codes.

G codes are used to control the robot path up to a target point, while F codes designate the speed to that point. The S code controls the robot end effector, peripheral devices, and program sequence.

Location specifies the exact physical position of each axis of the robot in joint angles. If program is created in work coordinate system, then point on work coordinate system and orientation is required in the position data.

4.2 Programming Fanuc Robot for Spray Painting Process

Since program is going to run on work coordinate system defined by the user, G97 'Cartesian Coordinate Input' command must be used in each block related with the surface data points. Besides, since the orientation has to be followed exactly for spray painting process, movement between the points ,which is the surface normal vector points, should be linear. To supply smooth path control, therefore, 'L' (linear) G code is used as well as 'G97'.

G97 instructs the move command using the following elements:

X,Y,Z : Position of the tool center point in the work coordinate system.
P,Q,R : Rotations of the tool coordinate system about X, Y, and Z-axis of the work coordinate system, respectively.
Operands : Attitude of the robot and angle of the wrist.

The surface data file, extracted from AutoCAD, consists of X, Y, and Z coordinates of the normal vectors in Work Coordinate and the surface normal at that point. Therefore, while post processing is performed, this surface data format is converted to the format which is required by the G97 code.

It is important that while a program is created with G97 code, tool center point, work and tool coordinate systems must be set. One of the work coordinates defined by the user is used for Cartesian coordinate. Similarly, G97 code requires one of the tool coordinates and one of the tool center points to be set before the program runs.

Here is an example of a block :

```
N0001   L   F0100M   G97,N,F,0,0,0
            X  400.00   Y  350.00   Z  200.00
            P   30.00   Q   25.00   R    0.00
          RON,1   BE
```

Operands of G97 code (N,F,0,0,0) is explained in Table 4.1.

The term 'F0100M' designate the velocity of the tool center point which is the spray gun's end point. Letter 'F' refers to the feed, as letter 'M' refers to the unit of millimeters per seconds. The four digit number shows the velocity of the robot end effectors speed up to the point defined in the position data.

The term 'RON,1' is an S code which is used for turning on output signal. This term is used to start paint flow from the spray gun. Similarly, 'ROF,1' is used to stop paint flow in spray painting process. 'BE' term indicates block end. After the last program block is reached this term should be 'PE' S code indicating the end of the program.

Since G97 code is used in almost every block, tool center point, the work and tool coordinate systems must be set carefully.

Table 4.1 G97 operands.

G97,N,F,0,0,0

Attitude configuration :

Operand	Display	Configuration
no 1 (β)	N	NO-FLIP
	F	FLIP
no 2 (θ)	F	FRONT
	B	BACK

Angle configuration :

Operand	Display	Configuration
no 3 (γ)	1	+180 < Angle ≤ +540 deg
	0	-180 < Angle ≤ +180 deg
	-1	-540 < Angle ≤ -180 deg
no 4 (β)	1	+180 < Angle ≤ +540 deg
	0	-180 < Angle ≤ +180 deg
	-1	-540 < Angle ≤ -180 deg
no 5 (α)	1	+180 < Angle ≤ +540 deg
	0	-180 < Angle ≤ +180 deg
	-1	-540 < Angle ≤ -180 deg

4.3 Setting Fanuc Robot's Coordinates for G97 Code

There are two coordinate systems related with G97 code as work and tool coordinates. When position data is given in the command data with G97 code, the robot moves to the position in work coordinate and orients the hand according to the data given in work and tool coordinate system and tool center point settings. Hence, if any of these three sets of data are mistaken, the desired position and orientation cannot be reached.

Base coordinate system is defined by the robot. The origin of the base coordinate system is the intersection point of the rotation centers of the θ and W axis. The Zo axis of base coordinate system is perpendicular to the θ axis rotation surface. The Xo axis is parallel to the arm at the zero degree position of θ axis (Figure 4.1).

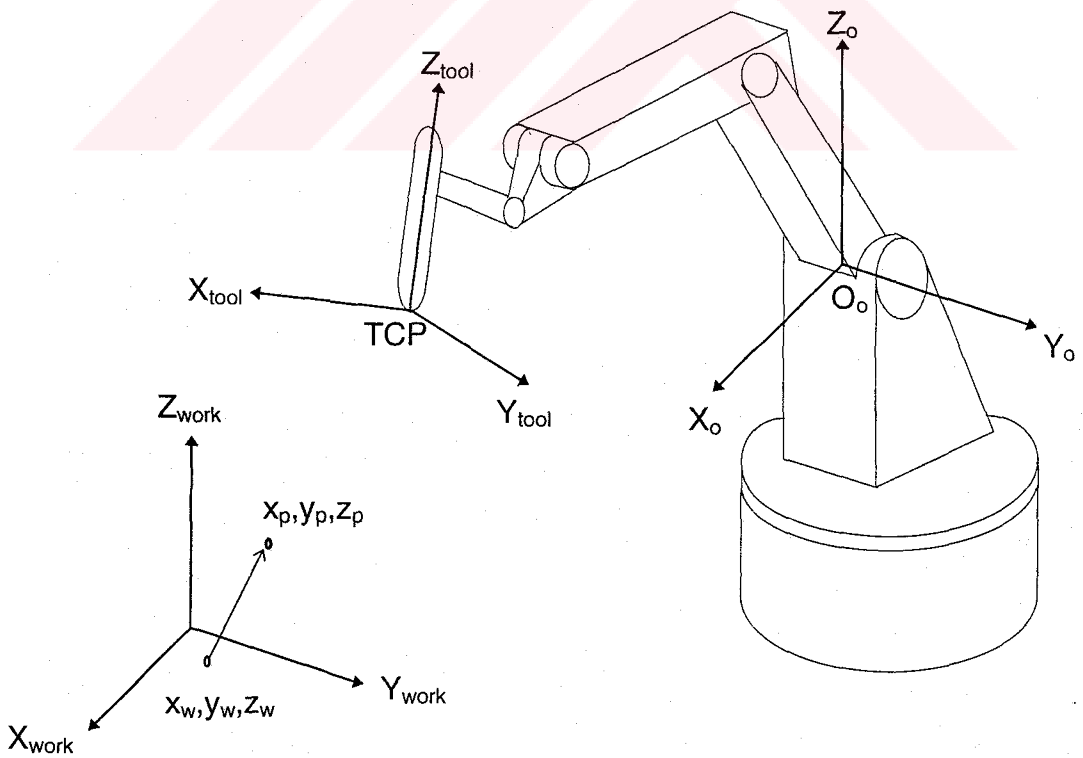


Figure 4.1 Base, work and tool coordinate systems.

To set work coordinate system, six data is required in terms of X, Y, Z, P, Q, and R. The X, Y, and Z are data defining the zero position of the work coordinate in the base coordinate system while P, Q, and R are the amount of the angles the coordinate system rotates about X, Y, and Z axis of the base coordinate system, relatively. If P, Q, and R are taken as zero then a work coordinate system parallel to the base coordinate system is obtained.

The tool coordinate system is defined by the movement of the end effector. To set it, each joint's position should be given such that each axis of the tool coordinate system becomes parallel to each axis of the base coordinate system.

Tool center point is the point at the end effector where work is executed. That point is the fluid exit at the nozzle on the spray gun for spray painting process. Tool center must be set not only for G97 code but also, for example, linear path control, since controller calculates the position data using the tool center point as a reference. The origin of the tool coordinate system is at the tool center point and, to set tool center point, distances from the base coordinate system, which is in that case defined as the center of the end effector mounting face when the robot is at the zero degree position, must be measured for the tool.

There are three tool center points, three tool coordinate systems and five work coordinate systems which can be set previously and can be changed in any program blocks. S63 code is used for that purpose.

4.4 Calculations for G97 Elements : X, Y, Z, P, Q, R

Surface data extracted from AutoCAD consists of position data (x_w, y_w, z_w) and the orientation in terms of unit cosines (cos α, cos β, cos γ). However, G97 code which is used to move in work coordinate requires position data (x_p, y_p , z_p) and orientation of the hand (P, Q, R).

To obtain the position data, (x_w, y_w, z_w) values are shifted along the normal unit vector by spray distance, d. Notice that the point (x_w, y_w, z_w) is on the surface while the point (x_p, y_p, z_p) is along the path of motion (Figure 4.1).

$$x_p = x_w + d \cos \alpha \tag{8}$$

$$y_p = y_w + d \cos \beta \tag{9}$$

$$z_p = z_w + d \cos \gamma \tag{10}$$

However, obtaining P, Q, and R values are not a straight forward task. They are calculated from geometric relations. In the Cartesian system, P, Q, and R represent the angles in 0.01° that the wrist axes will rotate from the P, Q, and R zero position defined by the tool coordinate system. They are determined as in Figure 4.2.

First, the work coordinate is rotated about its X axis by P. Second, the newly created coordinate is rotated about its Y' axis by Q. Then, this new created coordinate is rotated about its Z'' axis by R. P, Q, R values are determined when this newly created coordinate (X'''-Y'''-Z''') becomes parallel to the tool coordinate.

Since the spray gun's direction will be the same as the normal vector, whose unit cosines are known, the normal unit vector direction is taken as the Z axis of the tool coordinates as shown in Figure 4.3. Hence, to determine P, Q, and R values, projection of the unit vector to XY, YZ and ZX planes are made. For example, for the first quadrant, they are calculated by the equations as follows ;

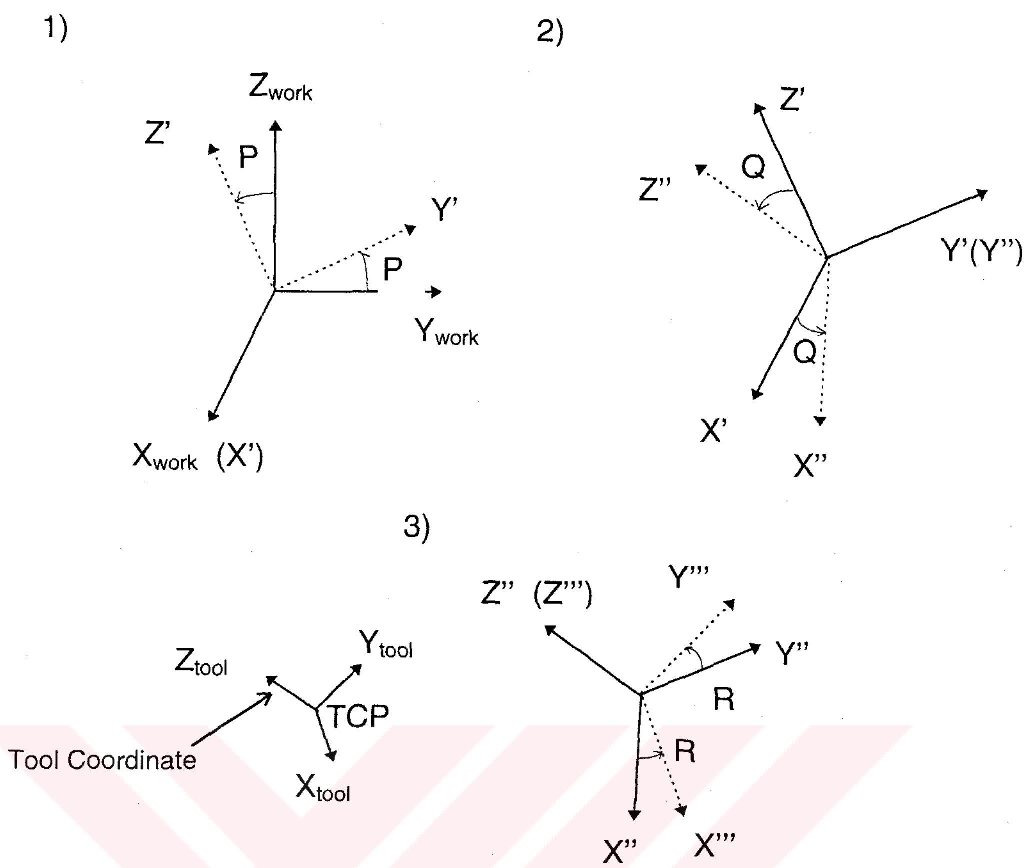


Figure 4.2 P, Q, and R rotations.

$$P = -(\tan^{-1}((\cos\beta)/(\cos\gamma))*180)/\pi \quad (11)$$

$$Q = \tan^{-1}((\cos\alpha)/(\cos\gamma))*180/\pi \quad (12)$$

$$R = \tan^{-1}((\cos\beta)/(\cos\alpha))*180/\pi \quad (13)$$

Equations (11) and (12) change with the quadrant in which the unit vector appears. Equation (13) is written by assuming the projection of Z axis (unit vector direction) and X axis of the tool coordinate system onto XY plane are on the same line. Equations for other quadrants are the same except for the signs of the P and Q angles.

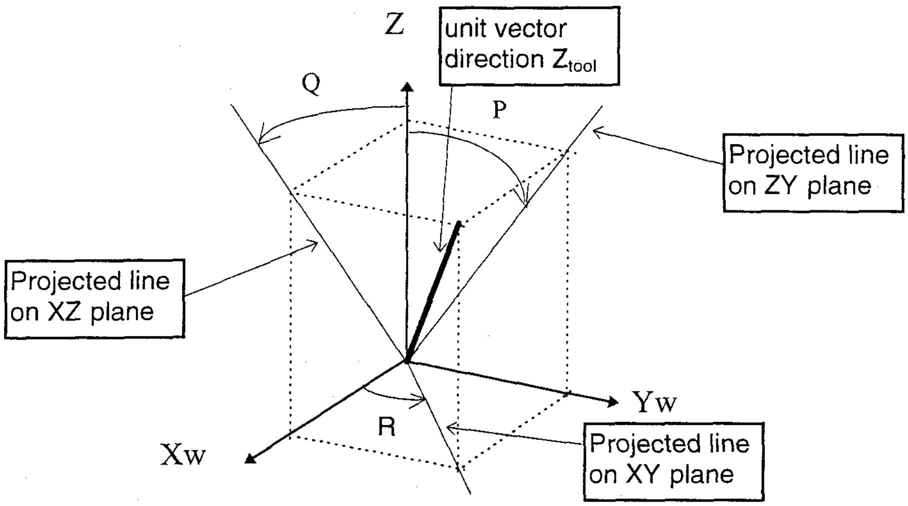


Figure 4.3 P, Q, and R values in the first quadrant.

It is obvious that the spray gun's direction is not dependent on R value. Therefore through the post processing operation, R may be taken as zero. Whether its effect is good or bad, can't be decided without inverse kinematic solution.

4.5 Spray Painting Apparatus mounted on Fanuc Robot

Spray painting apparatus consists of spray gun, fluid tank, solenoid valve, oil and water extractor, air supply, and pipes to link them to each other. Mounting configuration of spray painting apparatus is shown in Figure 4.4. In spray painting process, back pressure, nozzle opening valve position and the spray gun's velocity are important parameters.

The back pressure of the paint is adjusted from the valve upon the fluid tank. When the back pressure increased, paint flowrate increases slightly. Oil and water extractor (O&WE) does not have much effect on flowrate of the paint. However, increasing O&WE's pressure causes an increase in atomizing air pressure. Much atomizing air pressure causes reflection of the paint from the surface. Therefore spray painting process should be performed with the lowest possible O&WE pressure. The recommended pressure value for O&WE's pressure is about 4 kgf/cm^2.

Spray gun is mounted to the last link of Fanuc Robot as an end effector. To mount the spray gun, a mounting bracket is made allowing for setting it to different spraying angles. If the spray gun's position is changed, the effect of this change must be informed to the robot controller by making necessary changes in tool coordinate system and tool center point data.

Paint flowrate can be decreased or increased from spray gun steeply. Also paint spray pattern is adjusted from the spray gun, which is shown in Figure 4.5. To make the pattern look like a circle the valve should be turned in clockwise direction. However, flowrate can not be calculated exactly, since the valve on back side of the spray gun is not calibrated. This causes to an uncertainty of the paint flowrate from the nozzle and also the velocity of the spray gun, which needs to be calculated while surface data is extracted from AutoCAD. Velocity of the spray gun is one of the most important parameters in surface coating quality.

For that reason some experiments have been made to see the relation of the paint flow at different fluid tank pressures and nozzle opening valve position. Experiments are performed for fluid tank pressure values of 110 kPa, 124 kPa, 138 kPa, and 152 kPa by keeping the valve position same. The valve position is calibrated in between 0 and 60 degrees by an increment of 6 degrees.

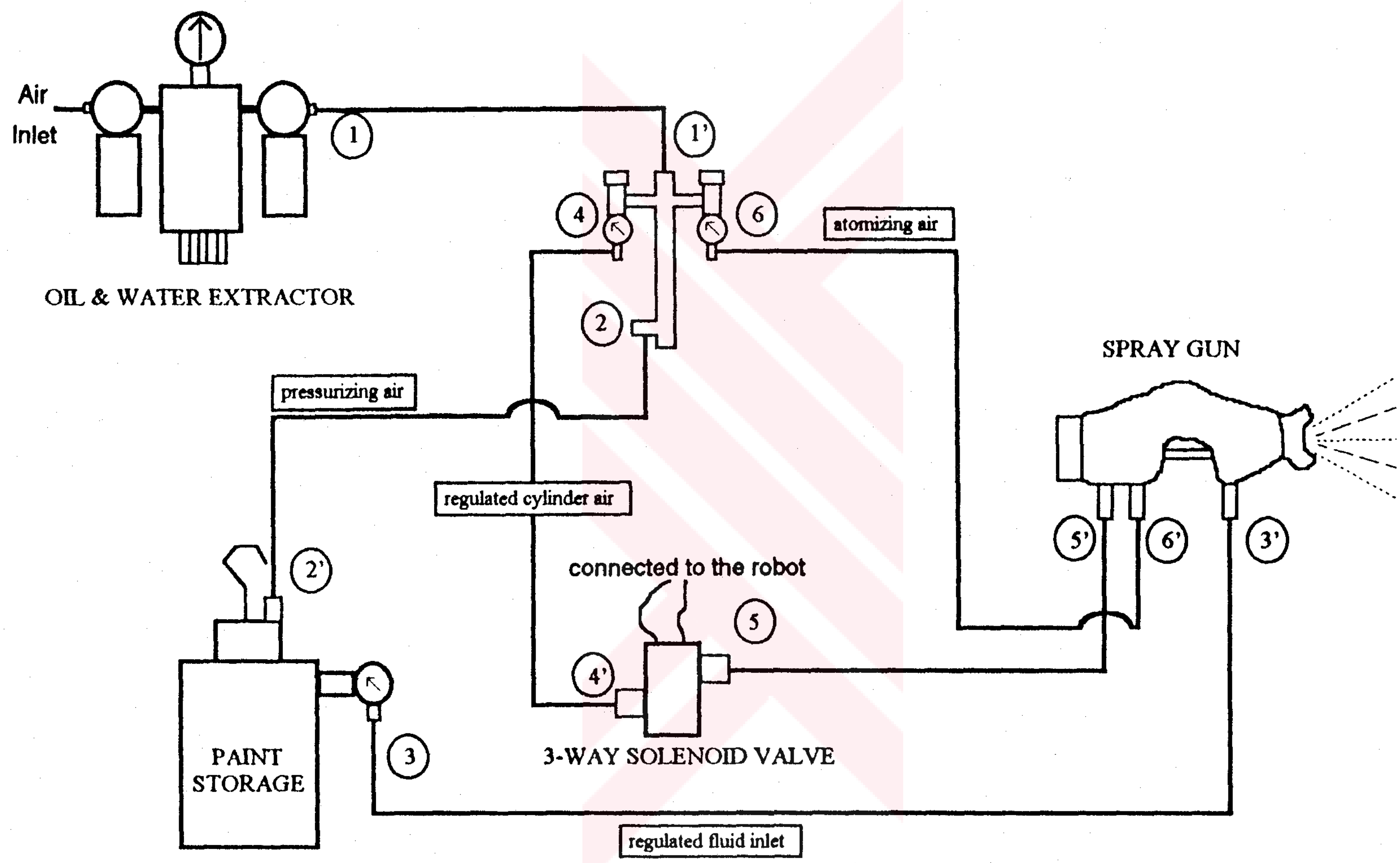


Figure 4.4 Mounting Configuration of Spray Painting Apparatus.

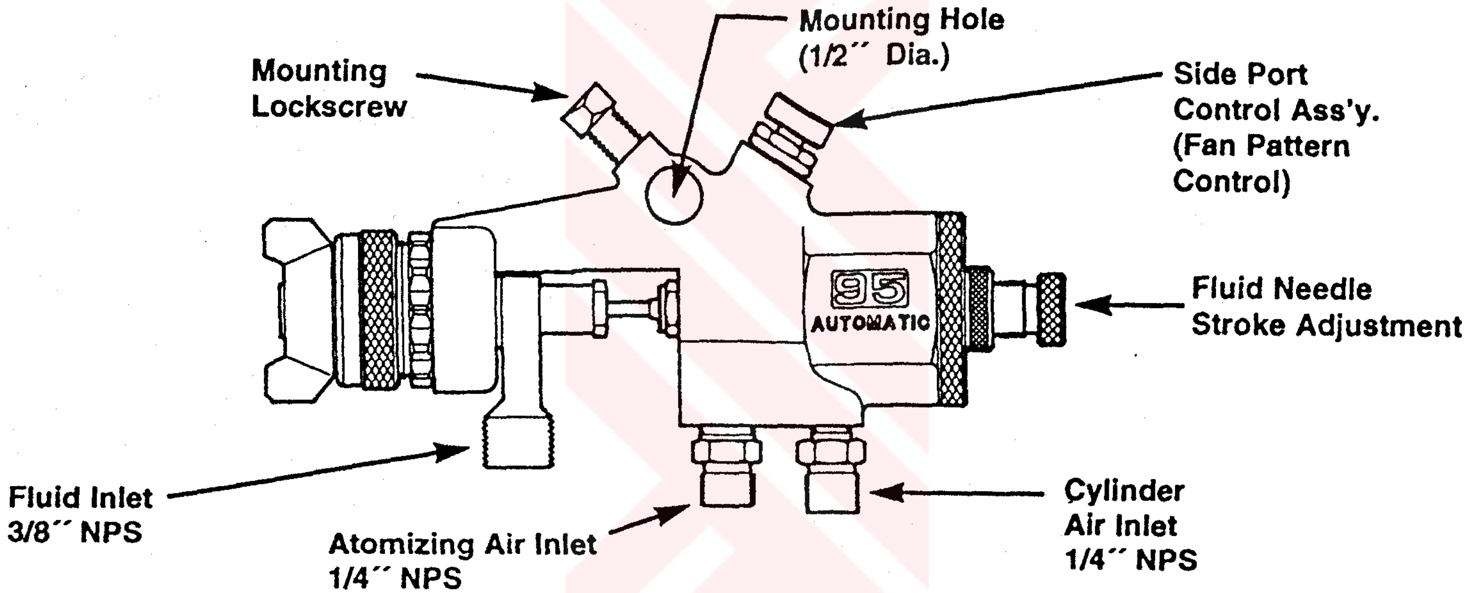


Figure 4.5 Spray Gun.

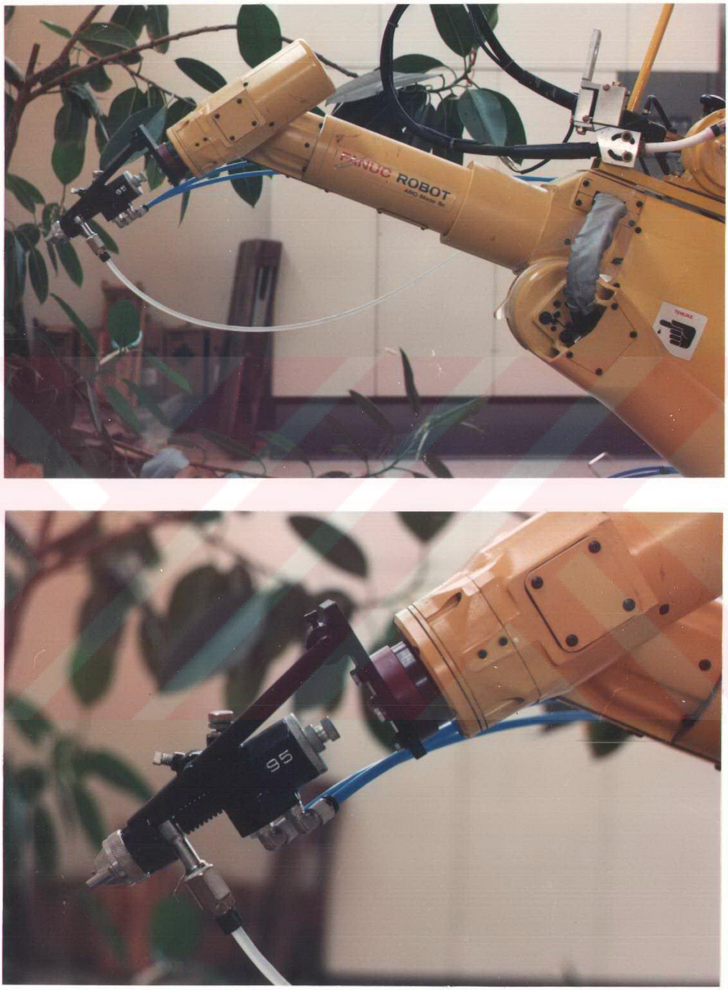


Figure 4.6 Spray Gun Mounted on Fanuc Robot.

Experiments are performed using water as the painting fluid. Water is classified as 'very thin' paint by the spray gun manufacturer. The fluid and air nozzles are selected according to the painting fluid.

The spray gun is an automatic one which can be controlled by the robot controller by using a solenoid valve. The spray gun that is mounted on Fanuc Robot is shown in Figure 4.6.

CHAPTER 5

ROBOPAINT : SOFTWARE FOR OFF-LINE PROGRAMMING OF FANUC ROBOT FOR SPRAY PAINTING PROCESS

5.1 Introduction to the Developed Software

The developed software runs under AutoCAD environment. Geometric Modeling of the surface is also performed in AutoCAD environment. After obtaining surface data information, output of the software is given in the robot's language as a text file for Fanuc R-G2 Controller. While the robot's text file is constructed, the spray parameters which are decided from spray variable dialog boxes are used.

AutoCAD is a well-known professional drawing program. It has several advantages against other drawing programs. It is relatively cheap, easily used with menus, and widespread around the world. However, main importance comes from that it is programmable. Programming in AutoCAD environment is enabled by AutoLisp which is a programming language of AutoCAD's own. By using this language, dialog boxes and functions for these dialog box terms can be written. AutoLisp requires basic knowledge of programming. It can be easily learned. Such an ability of programming allows programmer to make and develop his own AutoCAD environment.

Fanuc R-G2 Controller's programming techniques require knowledge of low level programming and a good knowledge of robot's own instruction set and, of course, theoretical background of robot motion. Fanuc R-G2 Controller's programming language is an instruction set coding with mnemonics. Post processing softwares can be developed for special processes by making use of some of these instruction sets.

Programs which are written in a special format can be downloaded from a computer to the Fanuc Robot. This peculiarity allows the off-line programming of the Fanuc Robot.

5.2 How to Run Robopaint

To run the developed software in AutoCAD environment, first of all the menu for developed software should be added, or if the user wants to use his own menu with the developed software, the source code for Robopaint menu should be added to the file "asurf.mnu". Besides, for automatic loading of the related lisp files the source codes should be added to "acadr12.lsp", otherwise these files which are used by the software (Robopaint) should be loaded while working in AutoCAD environment.

All Robopaint files should be in the "SUPPORT" directory of the AutoCAD or in one of the library directories. Since Robopaint uses AutoSurf, AutoSurf must also be loaded if not automatically loaded at the beginning of running AutoCAD. Related file is usually named as "asurf.exp".

Robopaint is mainly consists of spray variables dialog boxes and post processing. Besides, surface curvature checking exists in case of a critical radius of curvature. Also, robot's G97 code configuration and work coordinate position is shown by using a dialog box. Some of the AutoSurf commands are added for quick reachability of the user.

5.2.1 Items in Robopaint Menu related with AutoSurf

'Surface Display' item is usually used to change the number of u and v-lines of the surface selected. This is required, if surface lines will be used for surface data extraction. When spray radius is changed in the spray variable dialog box specially for planar or cylindrical surfaces, then it is important to set the correct number of surface lines. UV line numbers can be changed by picking the check box for UV lines and entering the desired number of U and/or V lines into the supplied boxes.

'Preferences' item shows a dialog box for setting surface variables such as augmented vector length and system tolerance. When working in different scales, augmented vector lengths should be set accordingly in order to pick and see them easily. Besides, for Robopaint, system tolerance must be between 1 and 5. If system tolerance value is set outside this range, the number of vectors on augmented lines becomes neither too few nor too many in order to describe the surface for spray painting process.

'Normal Direction' item is used to change the surface's unit normal vector direction. Unit normal vector indicates the surface's upper side which means the side to be spray painted. Cylindrical surfaces' unit normal vector shows always the inside of the cylinder. Therefore, direction for these surfaces should be changed unless if the inside of them is not spray painted.

If the surface is trimmed, then surface data cannot be taken by using surface lines. Therefore, 'SECTSF' command is used to create augmented lines while sectioning the surface with a step increment of spray radius. 'Apply Sectioning' item is selected from the Robopaint menu if a surface is to be sectioned.

Sectioning can be performed in two types in terms of Radial and Plane. 'Radial' option creates planar cross-section lines radially on a surface or multiple surfaces. It requires a plane by picking two points. The First point is taken as the origin of the plane that rotates on the point. The second point is the location of the cut plane at zero degrees. AutoSurf requests a Start angle and End angle. The Start angle is the degree, measured from the line represented by the two selected points, to start the cuts. The End angle is the increment between successive cuts. This option is used usually for cylindrical surfaces.

Command : **sectsf**
To invoke advanced options enter "??"...
Select surfaces : (Select the surface from which the cross section created)
Selection type Radial/<Plane> : **r**
Select first location : (Origin- center of the cylinder)
Select second location : (Locate a point- any point on the circle)
Start Angle <0> : (enter a number from -180 through 180)
End Angle <0> : (enter a number from -180 through 180)
Step Value <0> : (enter a number from -180 through 180)

'Plane' option creates X, Y, Z or cross section lines based on a selected plane from a selected surface or multiple surfaces. When this option is used to make X, Y, or Z section cuts, the cuts are made on planes that are normal to the specified axis at the specified positions.

When section cuts are made from a selected plane this option defines the section positions as the distance in the positive or negative normal direction from the selected plane. The selected plane represents the zero cutting plane, and the normal for the plane indicates the positive direction. If a trimmed surface is selected, the section lines begin and end at the surface borders.

Command : **sectsf**
To invoke advanced options enter "??"...
Select surfaces : (Select the surface from which the cross section created)
Selection type Radial/<Plane> : (press return for plane option) **p**
Plane by Entity/Last/View/Xy/Yz/ZAxis/Zx/<3 points> : (enter an option)
Start Value <0> : (enter the distance from the cutting plane where you want the cross section to start)
End Value <0> : (enter the distance from the cutting plane where you want the cross section to end)
Step Value <0> : (distance between successive cut planes- spray radius value)
Group name prefix <ASG> : (enter a group name prefix)

To create augmented lines with 'SECTSF' command advanced option is used by entering '??' in the 'Select surface' command line. It shows a dialog box as shown in Figure 5.1.

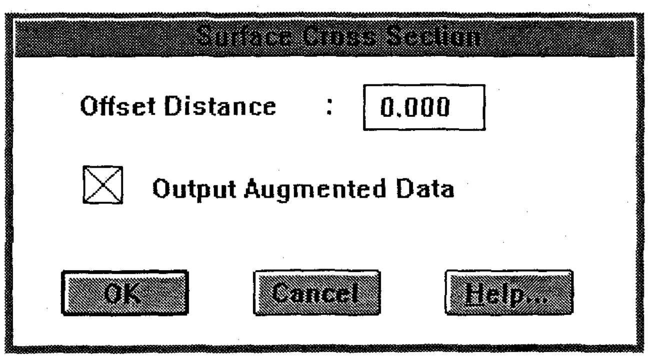


Figure 5.1 'SECTSF' command advanced option dialog box.

If surface lines is used to create Augmented lines then 'Create Aug. Lines' item in the Robopaint menu is used. ALs should be selected from the surface edges and the density of the vectors on an augmented line must be properly set for curved surfaces. This is because of the robot's motion following the created vector points linearly. This linearity causes deflection from the path, hence spray distance changes as shown in Figure 5.2.

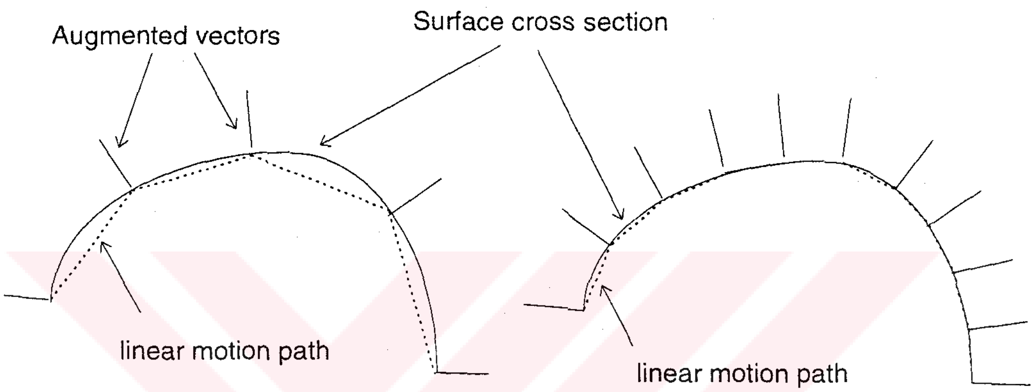


Figure 5.2 Effect of number of augmented vectors.

'CREATEAUG' command creates an Augmented line from wireframe entities or at the edge of a Surface. In spray painting, only the surface edges is used for creation of them. This option lets the control of the number of point on the Augmented lines created on the surface edge. There are three options for the determination of the points. It is better to use 'Optimal' option, which defines the minimum number of points on the Augmented line to be the number of points that are needed to keep the surface edge within the surface tolerance.

Command : **createaug**
To invoke advanced options enter "??"...
Create from wireframe/<Surface edge> : (press return for Surface edge option)
Select objects : (select surface edges)
Augmented by Points/Step/<Optimal> : (press return for Optimal option)

If the Surface is trimmed, this command can't be used to create Augmented lines from surface edges. While creating ALs, a group name to the created lines can be given for easily saving the point and vector data to an ASCII file.

Surface data can be extracted by 'Output Aug. Lines' item. 'OUTPUTAUG' command saves the Augmented line data (point and vector) to an ASCII file. This data is then used to create the robot program file for spray painting process of Fanuc Robot.

Command : **outputaug**
Select augmented lines : **g**
Select augmented lines : (pick one augmented lines to be saved to a file)
(enter the filename ,inwhich the data will be saved)

If a group name prefix is given to ALs, by entering 'group' (g) in the command line they can be selected by picking only one of the group member.

5.2.2 Items in Robopaint Menu related with Spray Painting

When a surface is composed of two or more primitive surfaces, then radius of curvature check should be performed to see if it is smaller than the spray diameter. 'Curvature check' item is used for that purpose. Before doing that surface data in both direction must be saved to the files " C: \ Painting \ u.aug " and " C: \ Painting \ v.aug ". Radius of curvature of the surface is investigated by using these data files. A dialog box shows the results and offers a painting direction as shown in Figure 5.3.

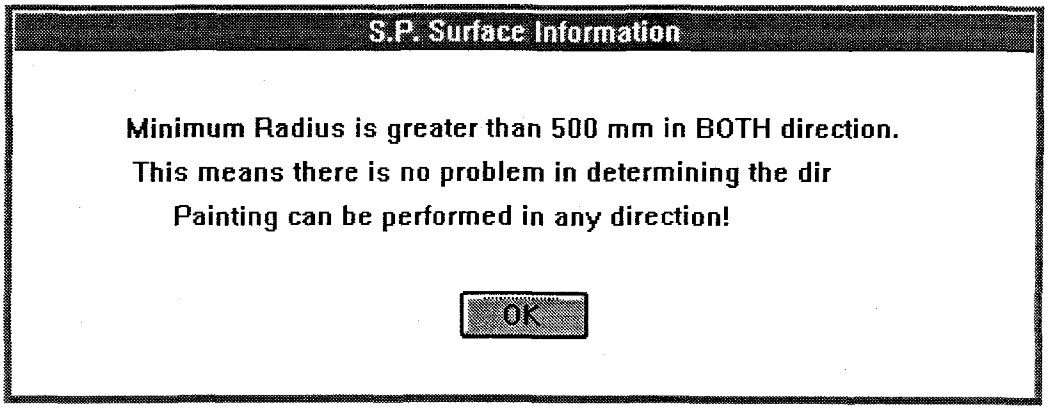


Figure 5.3 Radius of Curvature Information Dialog Box

'G97 Configuration' item is related with the configuration of the Cartesian coordinate input command (G97). Besides, the work coordinate data for the Fanuc Robot's settings is also be seen in that dialog box as shown in Figure 5.4.

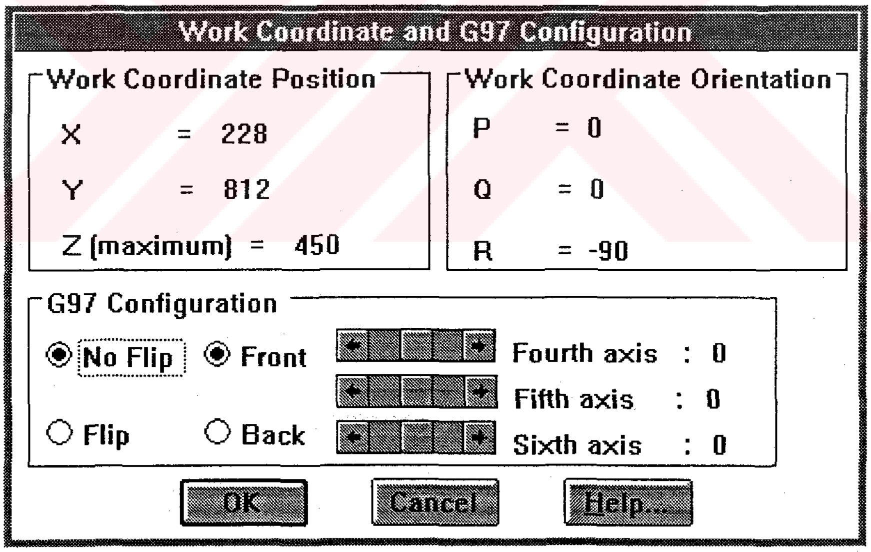


Figure 5.4 Configuration Dialog Box

In this dialog box, work coordinate system X, Y, and Z point data is calculated from the data file such that the middle point of the surface placed at

(700,0) in base coordinate. This point is approximately the middle point in the reachable area of the Fanuc Robot. All position data given in base coordinate. P, Q, and R values represents the orientation of the work coordinate which is parallel to the base coordinate except that it is rotated -90 degrees about Z axis. While post processing, G97 configuration is taken as in this dialog box.

Spray gun's velocity is an important parameter in spray painting process. It is dependent on the paint flowrate, the desired paint thickness and the spray pattern's radius. Paint flowrate depends on the spray gun's nozzle opening valve position and the paint tank pressure. To take the loss of the paint into account, paint flowrate is multiplied with a constant value called ratio of the paint coated on the surface over the paint flowrate at the nozzle. This ratio is about 50 percent for a regular spray gun. Spray pattern's radius is dependent on the half cone angle and the spray distance. The spray pattern's cone angle changes also with the paint tank pressure and the nozzle opening valve position.

'Spray Variables' item invokes a dialog box to decide the painting condition as shown in Figure 5.5.

The output of this dialog box is the spray gun's velocity which is required to supply the desired paint coating thickness under the given painting condition in terms of paint tank pressure and valve position.

The relations between the tank pressure, the valve position and dependent variables are obtained from experiments. The tank pressure and the valve position effect the paint flowrate at the nozzle and the spray pattern's half cone angle. The experiments showed that when either valve position or tank pressure is increased, paint flowrate and cone angle increases linearly. Results of the experiments are shown in Appendix B.

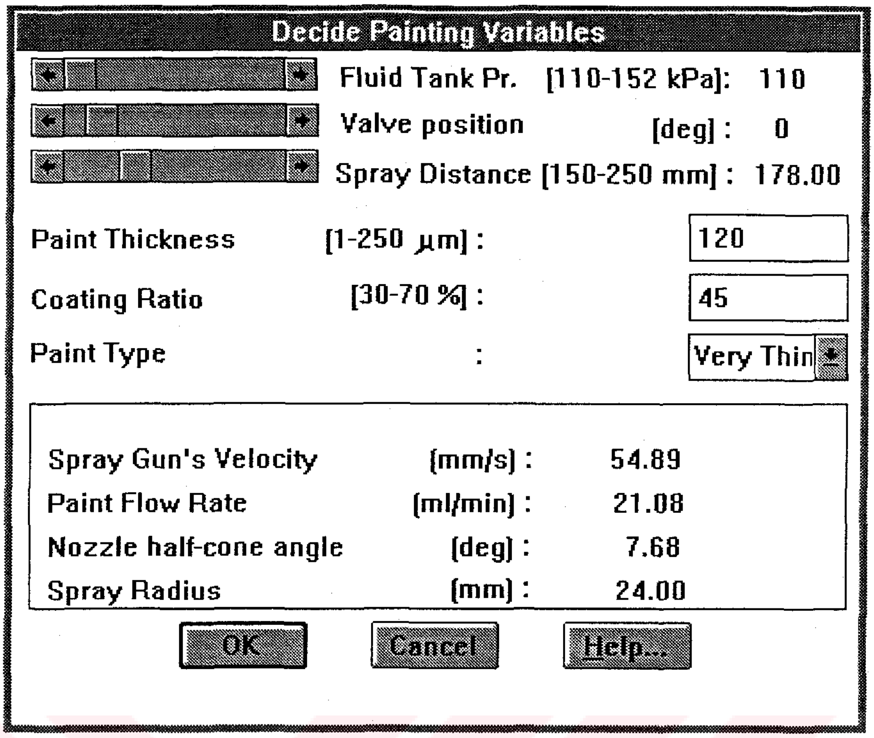


Figure 5.5 Complex Surface Dialog Box

The surface is sectioned according to the step value given in the dialog box. Sectioning is performed with 'Output Augment Data' advanced option of the 'SECTSF' command.

Having surface data file created, post processing can be performed to obtain the robot language text file for Fanuc Robot Arc Mate Sr. with R-G2 Controller.

5.3 Post Processing for Spray Painting Process

After surface data extraction, post processing takes place. There are two post processing options in terms of 'Linear' and 'Complex'.

Post processing operation for spray painting is written in AutoLISP. Two post processing operations are almost the same in terms of algorithm and process. Flowchart for post processor for free-formed surfaces is shown in Figure 5.6.

For the complex and planar surfaces' post processing operation, similar algorithm is used except the determining the beginning and end of the path. To determine the change of path line, angle between position vectors for three data points are investigated. In planar surfaces, the angle becomes zero if the point is in the Augmented line and 90 degrees if the path changes, as shown in Figure 5.7. When Augmented lines are created with sectioning, this value changes according to the curvature of the surface. However, it is kept almost constant through out the path due to the system tolerance used in determining the number of points. Therefore, change of path line for curved surfaces are decided according to the deflection from that angle.

In building robot program file, each lines corresponds to a block of the program. These blocks are composed of position control (P) or linear position control (L), motion speed, G97 configuration, position and inclination data, opening (RON,1) or closing (ROF,1) of spraying process, and finally the term defining the block end (BE) or program end (PE).

Linear position control is used while moving in the intra path, whereas position control is used to connect two successive intra paths. The reason is that while spray painting performed, robot should follow the path closely. If P control is used between the vectors in the intra path then path doesn't followed exactly. Because the path that the robot travels when under P control is usually nonlinear.

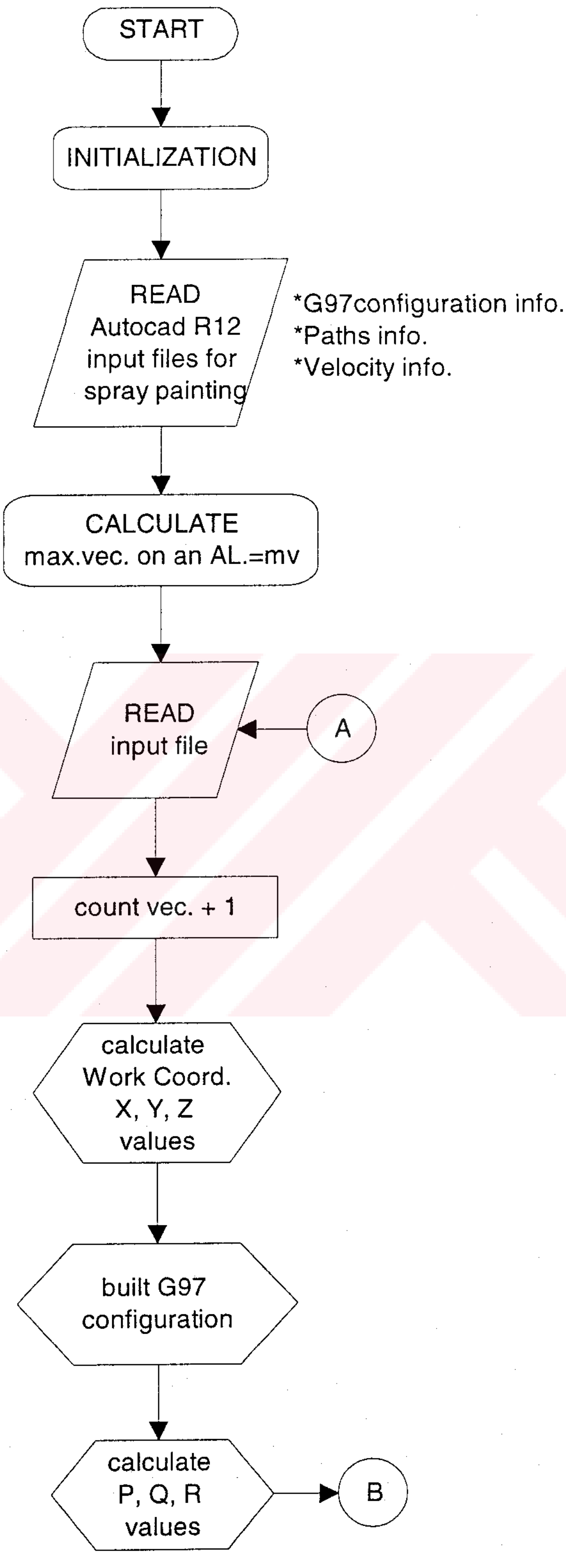


Figure 5.6 Flowchart for post-processing of free-formed surface.

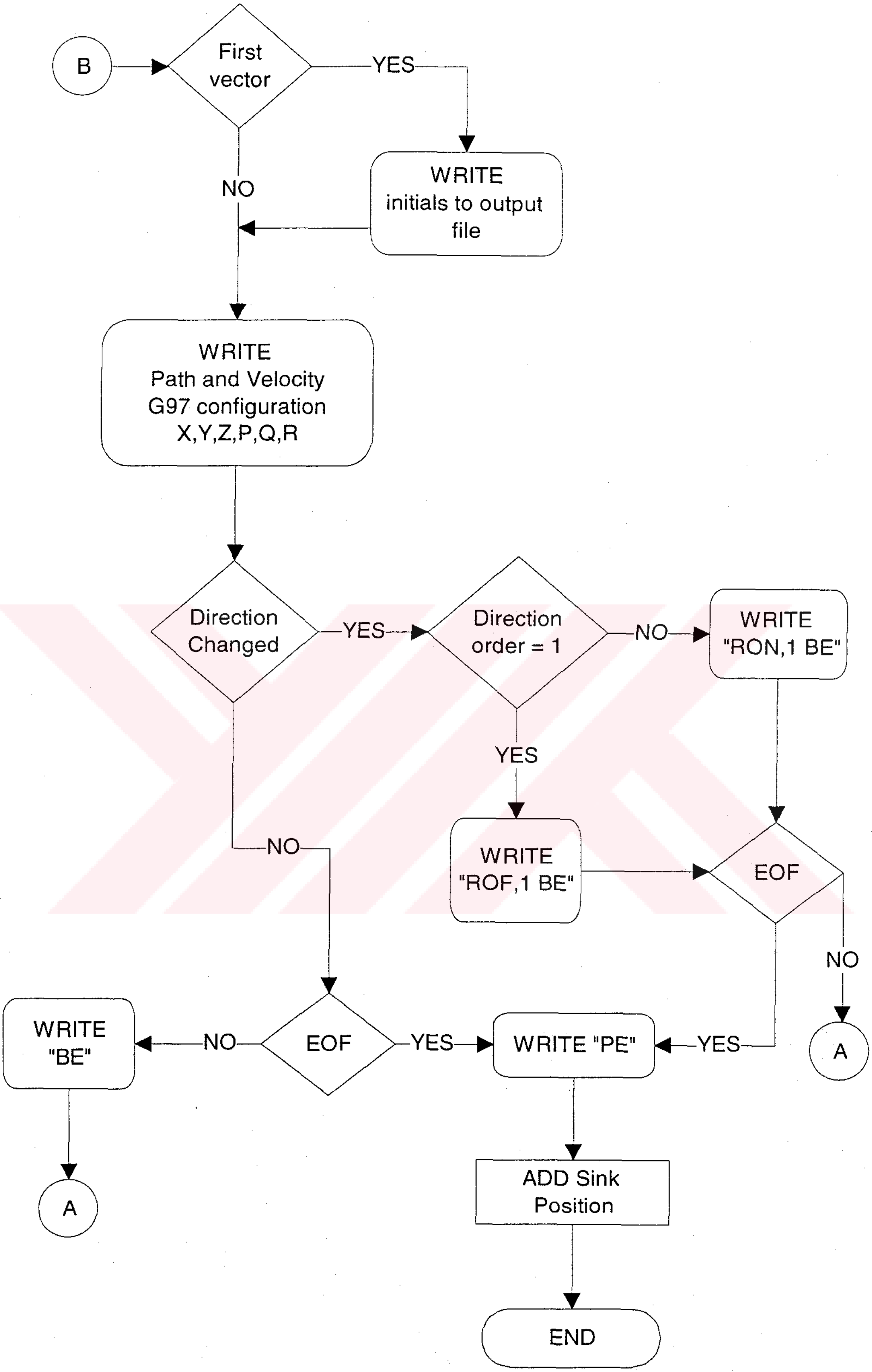


Figure 5.6 (Continued)

G97 configuration is kept constant throughout the program. If configuration is changed from one point to the other, the orientation of the spray gun changes absurdly causing a deflection from the path.

The position and orientation data is calculated from the surface data as explained in section 4.2. The opening and closing of the paint flow is decided according to beginning and end of the path. With the first point paint flow is allowed by 'RON,1' term and when the path changes paint flow is closed by 'ROF,1' term.

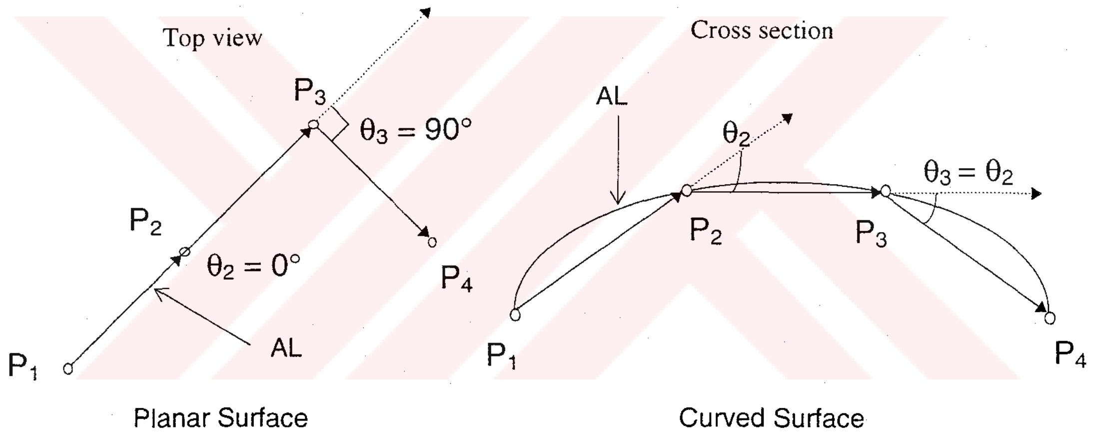


Figure 5.7 Determining the change of path lines with the angle between position vectors.

5.4 Downloading the Robot Program to the Fanuc Robot

To Download the robot program file to the Fanuc Robot, there is procedure followed for the computer and the robot side. For the Computer side the procedure to be followed is as follows:

1. Copy the robot program file to the "C:\>ROBOT> " directory,
2. Run the program named 'fanuccom.exe' in the "C:\>ROBOT> " directory,
3. Select the send file option,
4. Select program file option,
5. Input the file name and press enter.

The computer is now ready to send the file. By using the teach pendant this procedure must be followed in order to save the file to the robot :

1. Press the function button,
2. Select device option,
3. Select host option,
4. Select load option,
5. Select program option,
6. Enter the program number where the file is to be downloaded.

While downloading there are two important points. There should not be any program having the same program number entered and there must be enough free-space for the robot program to be saved into the robot's memory. (Free-space is shown while O-MAP). Free-space shows the maximum number of blocks that the program can have. If there were not enough memory left for downloading, then some of the programs in the robot's memory should be deleted or saved to the computer's hard disc for later usage. To save a program to the computer, similar procedure should be followed.

To save a program to the computer these procedures must be followed.

Computer side :

1. Run the program named fanuccom.exe in the "C:\>ROBOT> " directory
2. Select the receive file option,
3. Select program file option,
4. Enter file name to which robot program is going to be saved,
5. Press enter.

Robot side :

1. Press the function button,

2. Select device option,
3. Select host option,
4. Select save option,
5. Select program option,
6. Enter the program number to be saved.

Before running the program, paint tank pressure and valve position values must be set to the values that are decided in the AutoCAD session. Workpiece must be placed at the same coordinates in the work coordinate as it was placed in the drawing. First work coordinate settings is used while running the program.

5.5 Limits of the Software and Painting Strategy

In spray painting process, transfer efficiency is between 45 and 55 %. Better transfer efficiencies requires spray guns with teflon needle [5]. These guns have transfer efficiencies between 60 and 90 % depending on the application. The spray gun mounted on the Fanuc Robot is supposed to have a transfer efficiency of 50 %. The experiments showed that with this transfer efficiency there is not much difference in painting a curved or a plane surface in the recommended spray distance limits. However, the curved surface's radius of curvature must be higher than the spray radius available by the spray gun's nozzle and painting condition. Hence, only the surfaces, which have smaller radius of curvature values than the spray radius available by the painting configuration, can not be spray painted.

In off-line programming for spray painting process, the most important thing is to get the surface data that gives the required surface equation. However, in AutoSurf, surface lines are used to get the surface data, instead of surface equation. Extracting surface data from surface lines causes deflection from the spray radius in complex surfaces. In complex surfaces, due to the curvature of the surface and the differences in the edge lengths, the distance between two sequential surface lines doesn't become constant through the path.

One of the most important parameter in spray painting is the spray gun's velocity. The velocity is dependent on the paint flowrate in turn the spray painting variables in terms of paint tank pressure and spray gun's valve position. The maximum velocity limit depends on the experimentation with different painting conditions.

Planar surfaces are the easiest surfaces to be spray painted. However, sharp corners built by planar surfaces should be cared. These surfaces must be treated separately.

Primitive surfaces like cylindrical, spherical, and conical, must be painted in two steps. They must be separated into two part such as front and back side or left and right side. Surface data should be extracted from one of these parts and post processing must be performed for that part.

While determining the painting direction for primitive surfaces, always the parallel surface lines must be taken into account. For a cylindrical surface, surface lines in either direction can be taken as painting paths, whereas for conical and spherical surfaces only the surface lines parallel to each other must be taken as painting paths. This should be applied to complex surfaces as well.

CHAPTER 6

SAMPLE SOLUTIONS

6.1 Spray Painting of a Complex Surface

To create a surface of revolution, there must be a path curve or profile and an axis around which the profile will be rotated. If path curve is created by using polylines as in Figure 6.1, the resulted surface becomes a continuous one as in Figure 6.2. If Lines are used for the path curve, then for each line a surface is created by surface of revolution as shown in Figure 6.3.

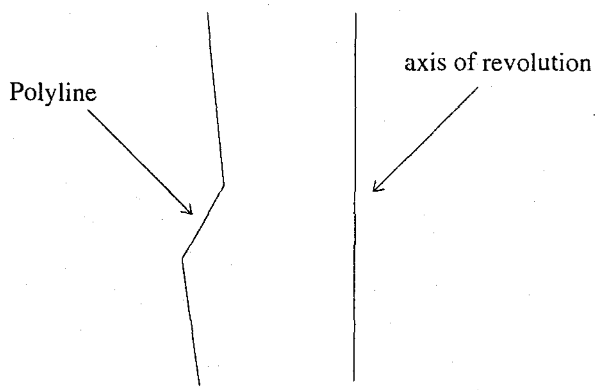


Figure 6.1 Elements of Revolved Surfaces

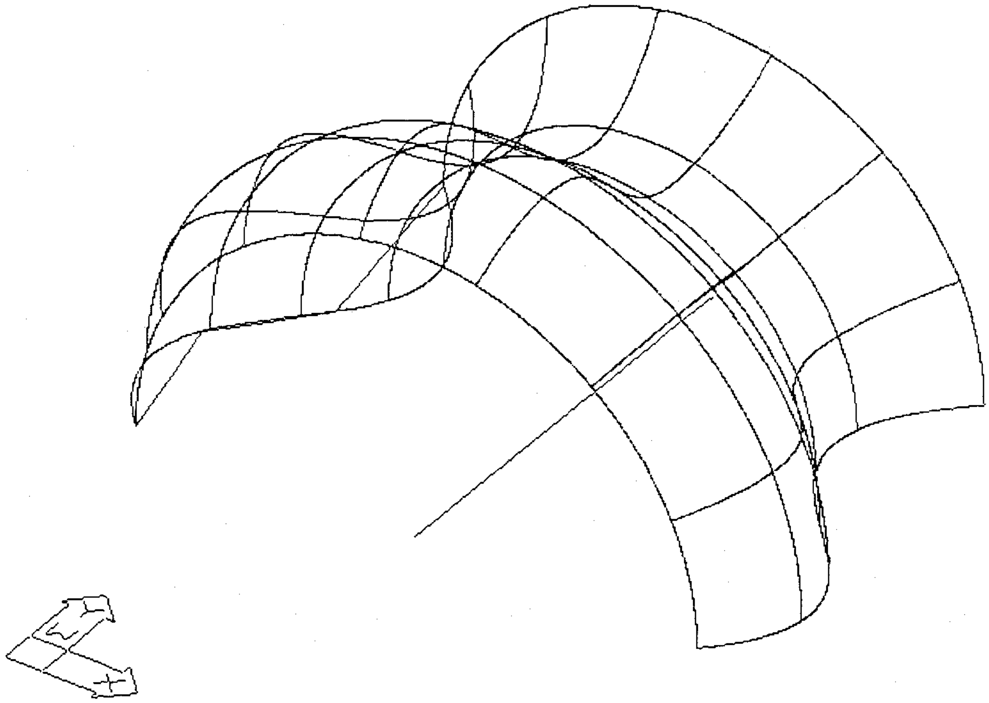


Figure 6.2 Revolved Surface resulted from Polyline profile.

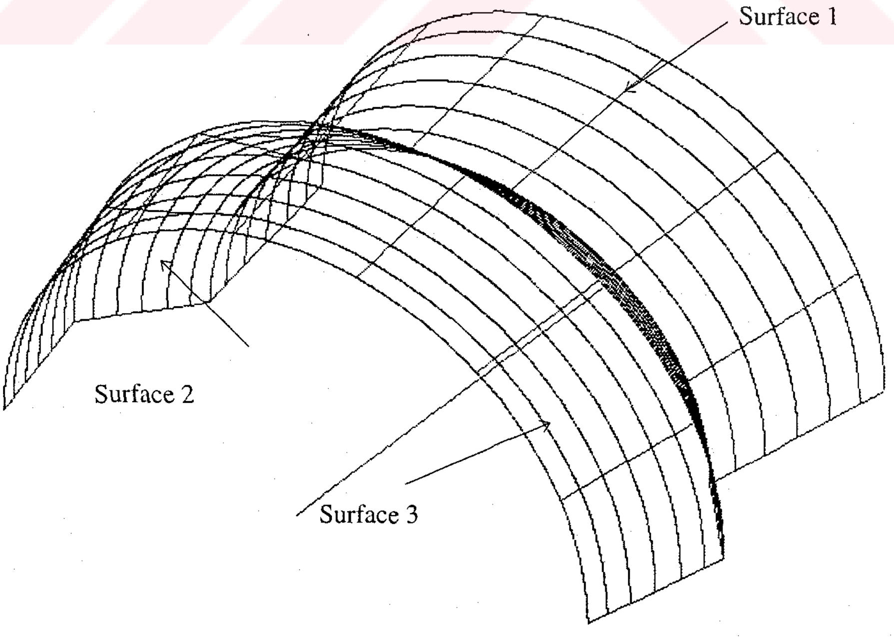


Figure 6.3 Revolved Surface resulted from Line path.

The surface in Figure 6.2 is created from entities that is shown in Figure 6.1 by selecting 'Create > Surface > Revolve' from asurf menu. Then, in the command line, the procedure follows :

Command : _revolvesf
To involve advanced options enter "??"...
Select profile curves : (pick Polyline)
Select profile curves : (press return to end selection)
Axis by Entity/<Locations>: **e**
Select line or polyline : (pick axis of revolution)
Start angle : <0> : **0**
End angle : <360> : **180**

After surface is created, to decide the painting direction radius of curvature check should be performed in the two possible painting direction. To do that surface data along these directions must be saved to the files named as 'u.aug' and 'v.aug' in the'C:\PAINTING\' directory.

To extract surface data in X- and Y-axis direction, sectioning may be performed, or surface lines can be used for creation of the augmented lines. If sectioning applied, procedure is followed as such:

Command : **sectsf**
To invoke advanced options enter "??"...
Select surfaces : ??
(select Output Augmented Line option from the dialog box)
Select surfaces : (pick the surface in Figure 6.2)
Select surfaces : (end selection)
Selection type Radial/<Plane> : (press return for plane option) **p**
Plane by Entity/Last/View/Xy/Yz/ZAxis/Zx/<3 points> : **y**
Point on XY plane : **0,0** (a point on the right side of the figure)
Start Value <0> : (press return)
End Value <0> : 1000,0
Step Value <0> : 40
Group name prefix <ASG> : udirection

Similarly, entering 'z' in the command line for the below command augmented lines parallel to X-axis are created.

Plane by Entity/Last/View/Xy/Yz/ZAxis/Zx/<3 points> : **z**

To save the surface data in the files 'Output Aug. Lines' Robopaint menu item is selected or 'CREATEAUG' command is entered at the command line. The surface data parallel to X axis direction is saved to 'C:\PAINTING\U.AUG'. Similarly, the surface data parallel to Y-axis direction is saved to 'C:\PAINTING\V.AUG' file. To select all vectors 'group' option is entered in the command line.

Command : **outputaug**
Select augmented lines : **g**
Select augmented lines : (pick one augmented lines to be saved to a file)
Select augmented lines : (press return to end selection)
(select the file c:\painting\u.aug in the dialog box that appears.)

After saving the surface data in to the mentioned files, 'Curvature Check' item in Robopaint menu is selected to start radius of curvature check of the surface.

The surface data files are investigated and the minimum radius of curvature is given in the surface information dialog box as shown in Figure 6.4.

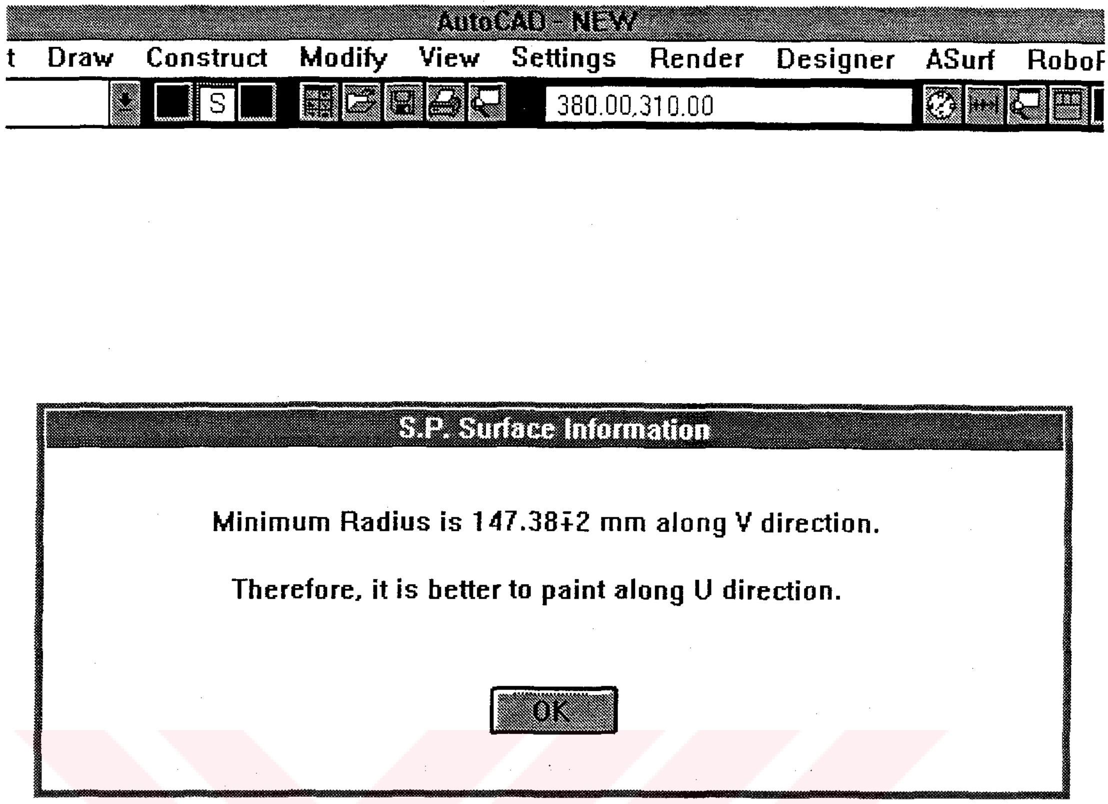


Figure 6.4 Surface Information Dialog Box.

Having decided the painting direction, the Surface should be sectioned again with a step value equal to the spray radius. Spray radius value is set while painting condition is decided. Painting condition is decided by selecting 'Spray Variables' item in Robopaint menu, which shows a dialog box shown in Figure 6.5.

After the painting condition is decided, by pressing 'OK' button the step value for sectioning and painting condition is shown briefly with a dialog box as shown in Figure 6.6.

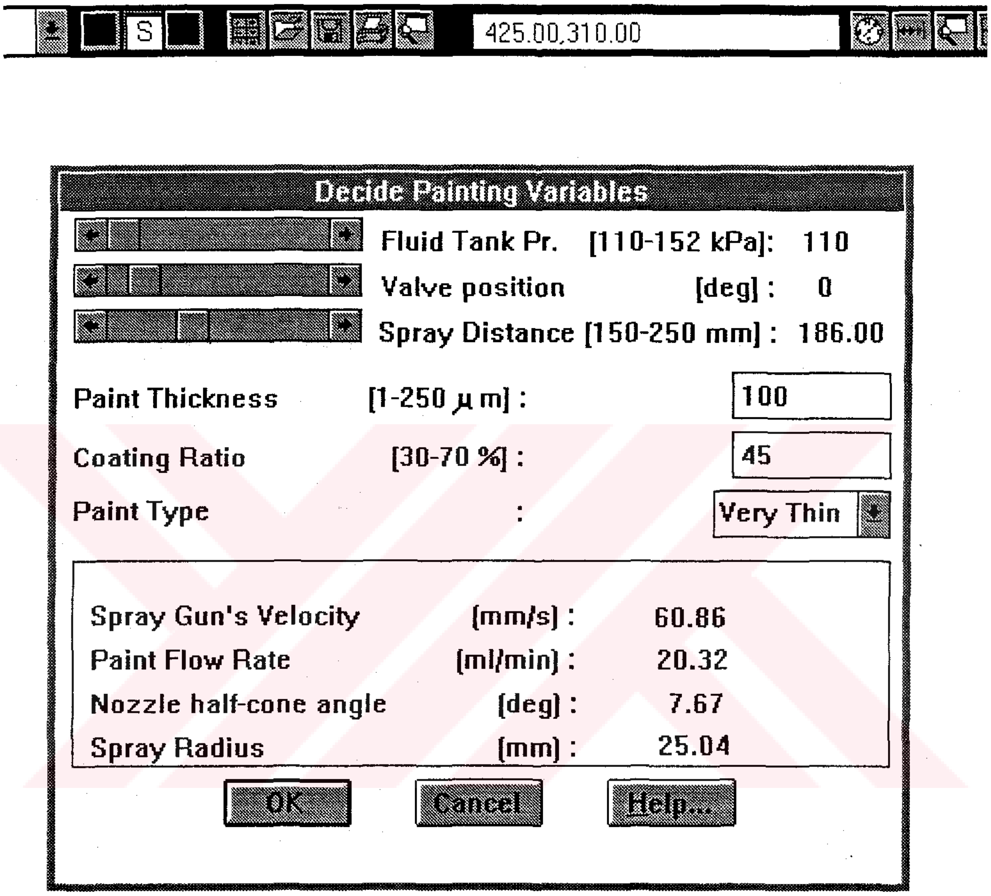


Figure 6.5 Spray Variables Dialog Box.

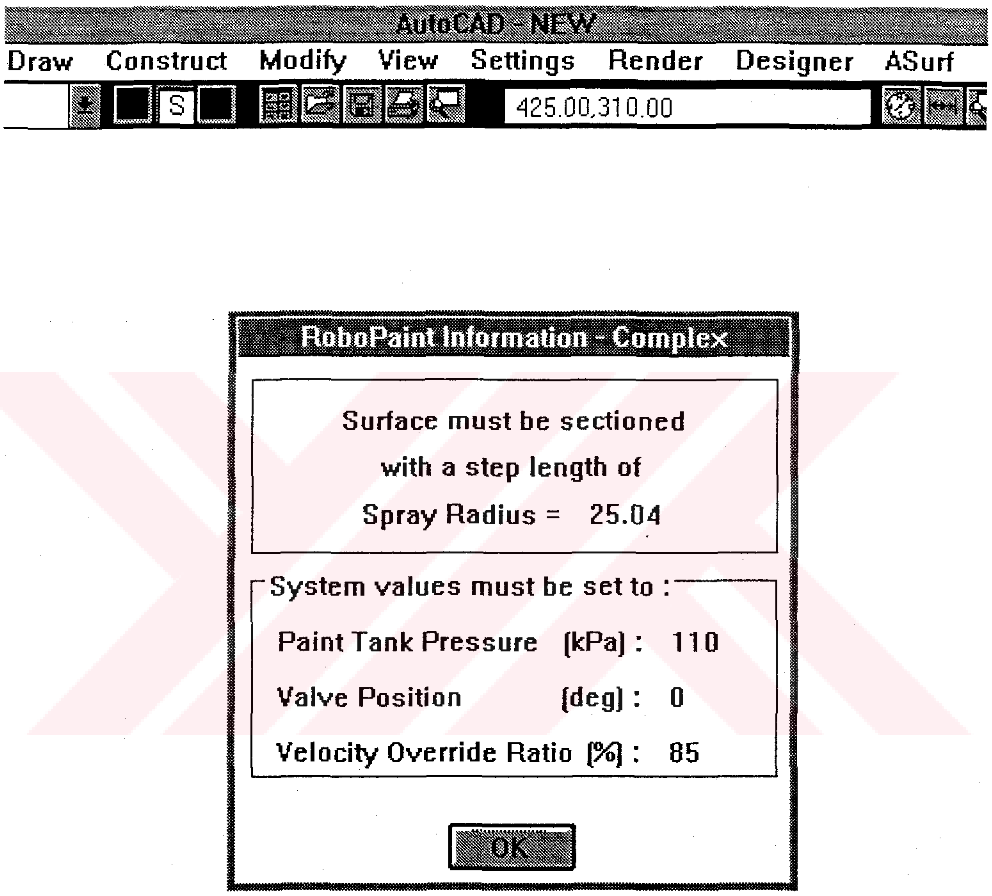


Figure 6.6 Painting Condition Information Dialog Box.

Surface must be sectioned parallel to X-axis (u line direction) with a step increment of spray radius value. Similarly, surface data must be saved to a file. The sectioned surface with augmented lines is given in Figure 6.7.

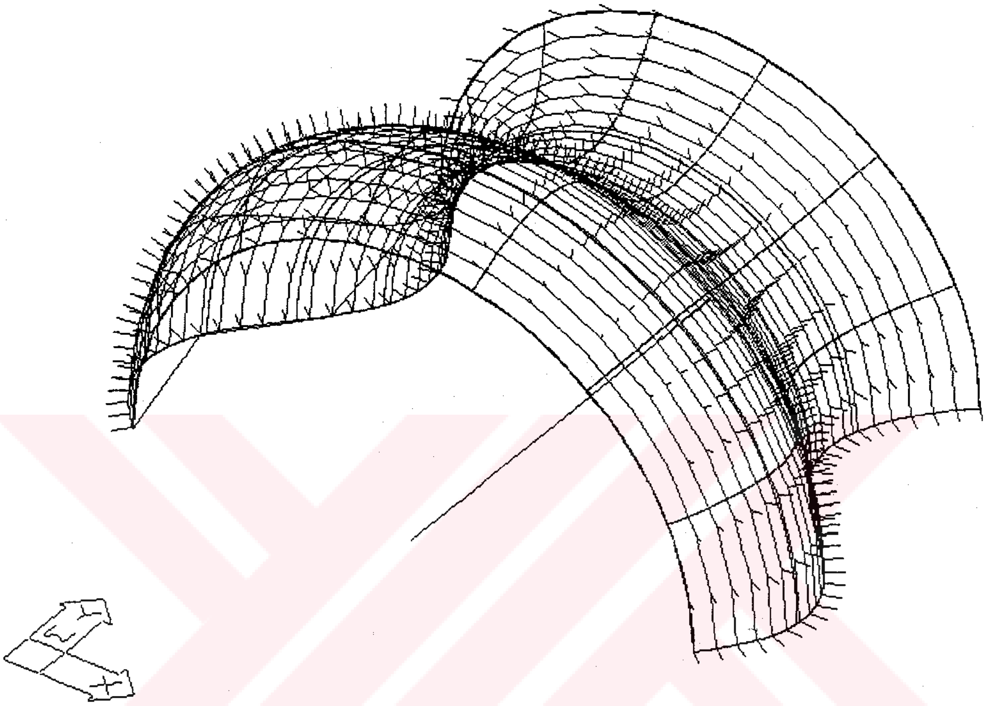


Figure 6.7 Sectioned surface with Augmented lines.

After having surface data, before post processing the surface data file name and the robot program file name must be entered by selecting the Robopaint menu item 'I/O Filenames'. It shows a general AutoCAD select file dialog box as shown in Figure 6.8.

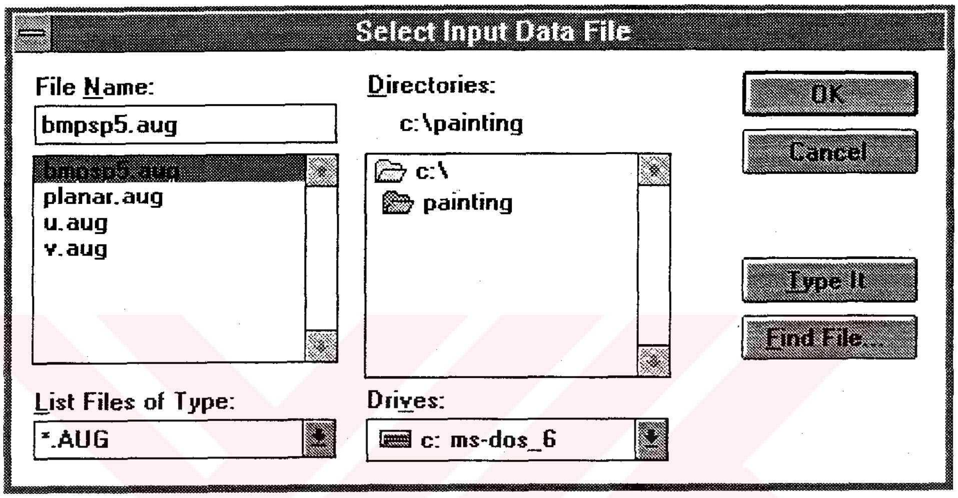


Figure 6.8 Dialog Box for entering Surface Data Filename.

Finally, post processing is performed by selecting 'Post Process' option in the Robopaint menu. It shows a dialog box first, as shown in Figure 6.9. The resulted robot program file is shown by a dialog box as shown in Figure 6.10 by selecting the 'Show Output' menu item.

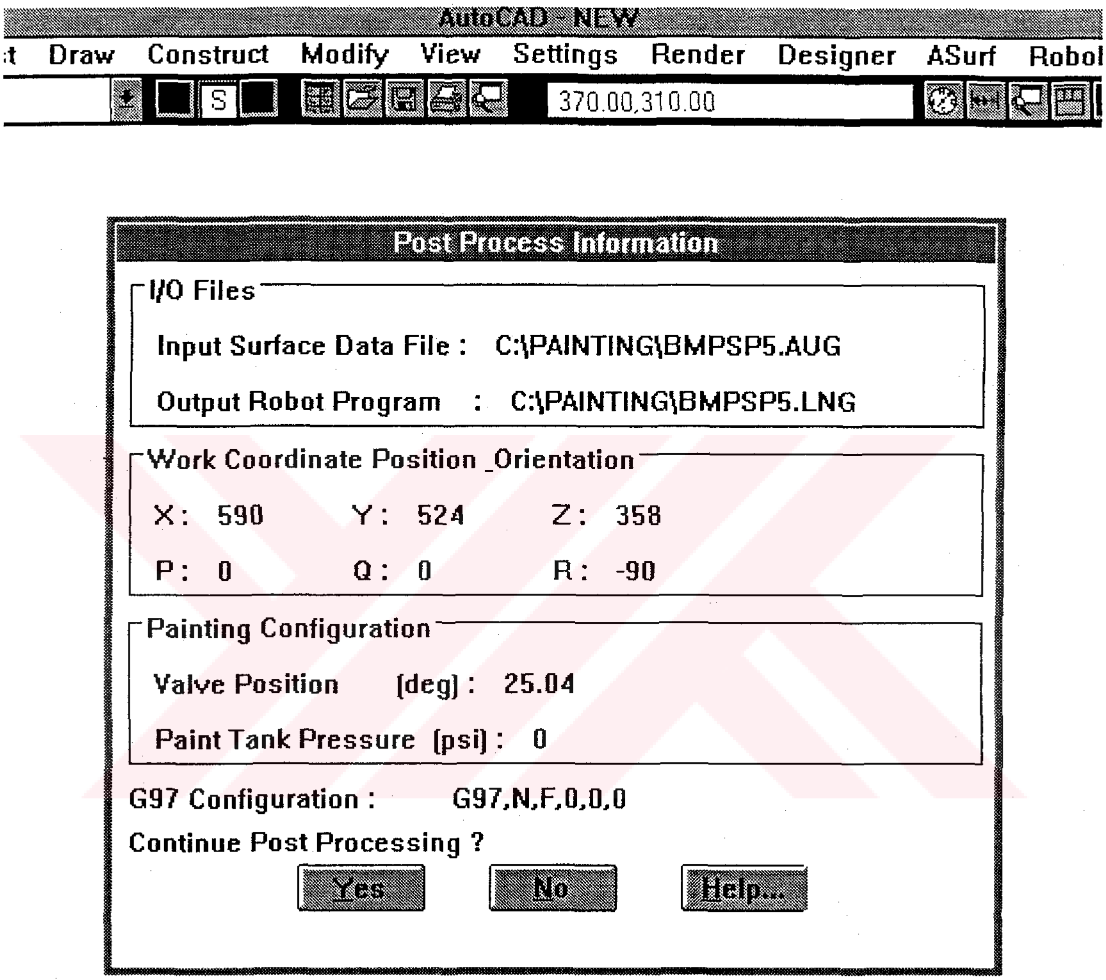


Figure 6.9 Post Process Information Dialog Box.

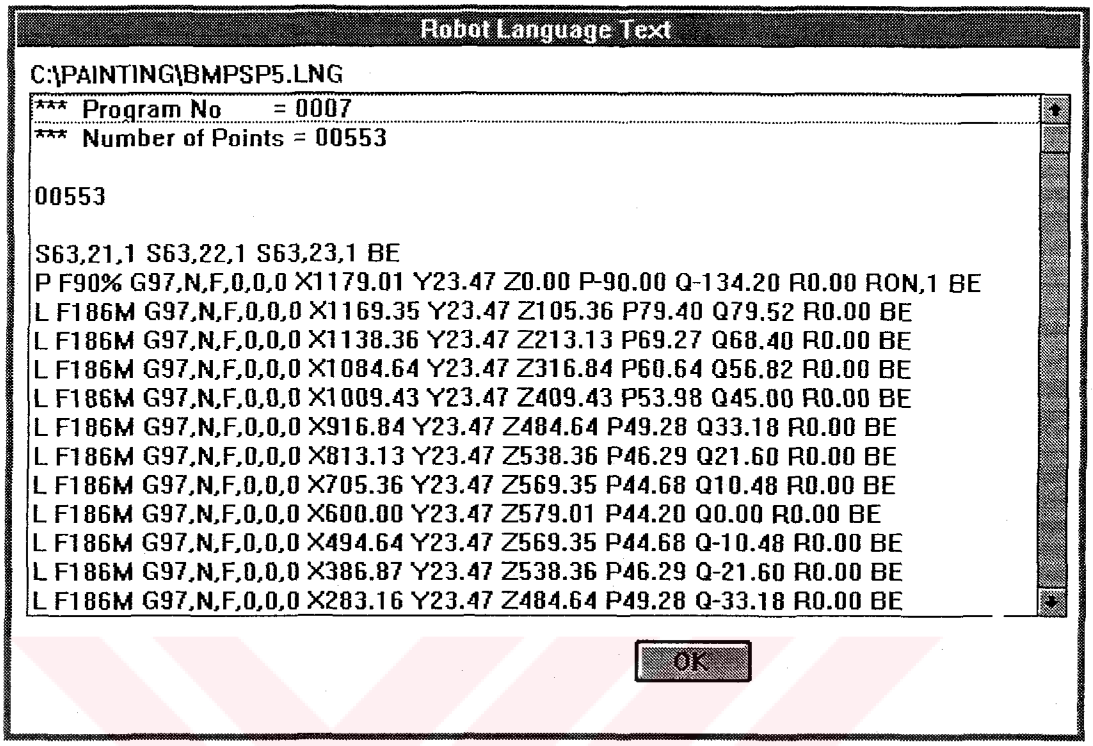


Figure 6.10 Dialog Box for showing Robot Program ASCII File

6.2 Spray Painting of Half of a Cylindrical Surface

Cylindrical surface is a primitive surface and can be created by selecting the "Create>Cylinder" option in asurf menu. However, this command can not create half of a cylinder. The other way to create a cylinder is to revolve a line around an axis line about 360 degrees. Similarly, half of a cylinder can be created by revolving a line around an axis line about 180 degrees as shown in Figure 6.11.

There are two ways of extracting surface data for that surface. First one is to use the surface lines and the other is to make section with output augmented line option by using SECTSF command. The surface lines in any direction can be used but line paths must be the first choice. Therefore, surface lines parallel to the center line must be chosen for the painting paths.

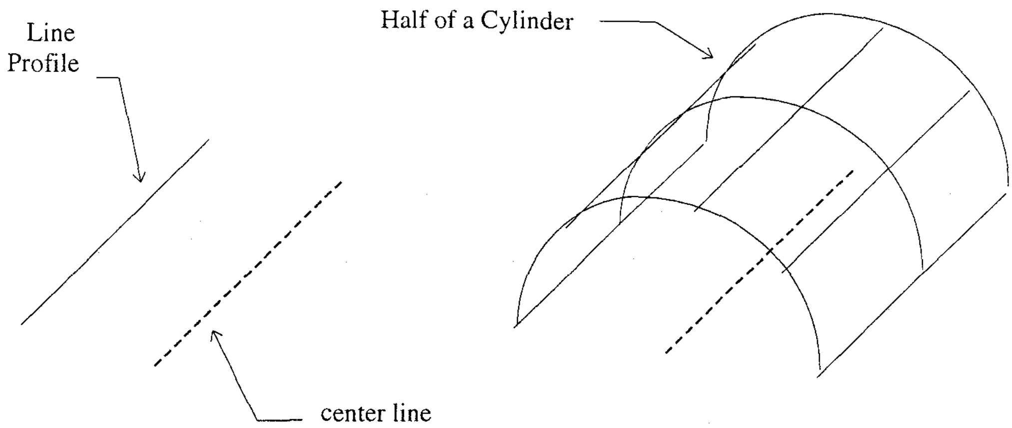


Figure 6.11 Creating half of a Cylinder.

If the surface lines are used for the extraction of the surface data, first of all the number of the lines must be decided. This is done by deciding painting condition for cylindrical surface with spray variables dialog box as shown in Figure 6.12.

After deciding variables, a dialog box shows the number of surface lines to be set. The number of surface lines can be set to the desired number by selecting the surface and selecting the 'Display Surface' option in the Robopaint menu. A sectioned surface with output augmented line option is shown in Figure 6.13. If sectioning is to be applied, it is performed about Z-axis and increment is given in degrees corresponding to the arc length amount of the spray radius value.

When the surface lines are set to the correct number, creation of the augmented lines is performed by using 'Create Aug. Lines' option in Robopaint menu. All surface edges parallel to the center line is selected. Optimum option is used to decide the number of point on an augmented line. The next step is to save the surface data into an ASCII file. 'Output Aug. Lines' option is selected to make saving. Just before post processing, input surface data file name and the desired robot program file name must be entered by selecting 'I/O Filenames' option in the Robopaint menu.

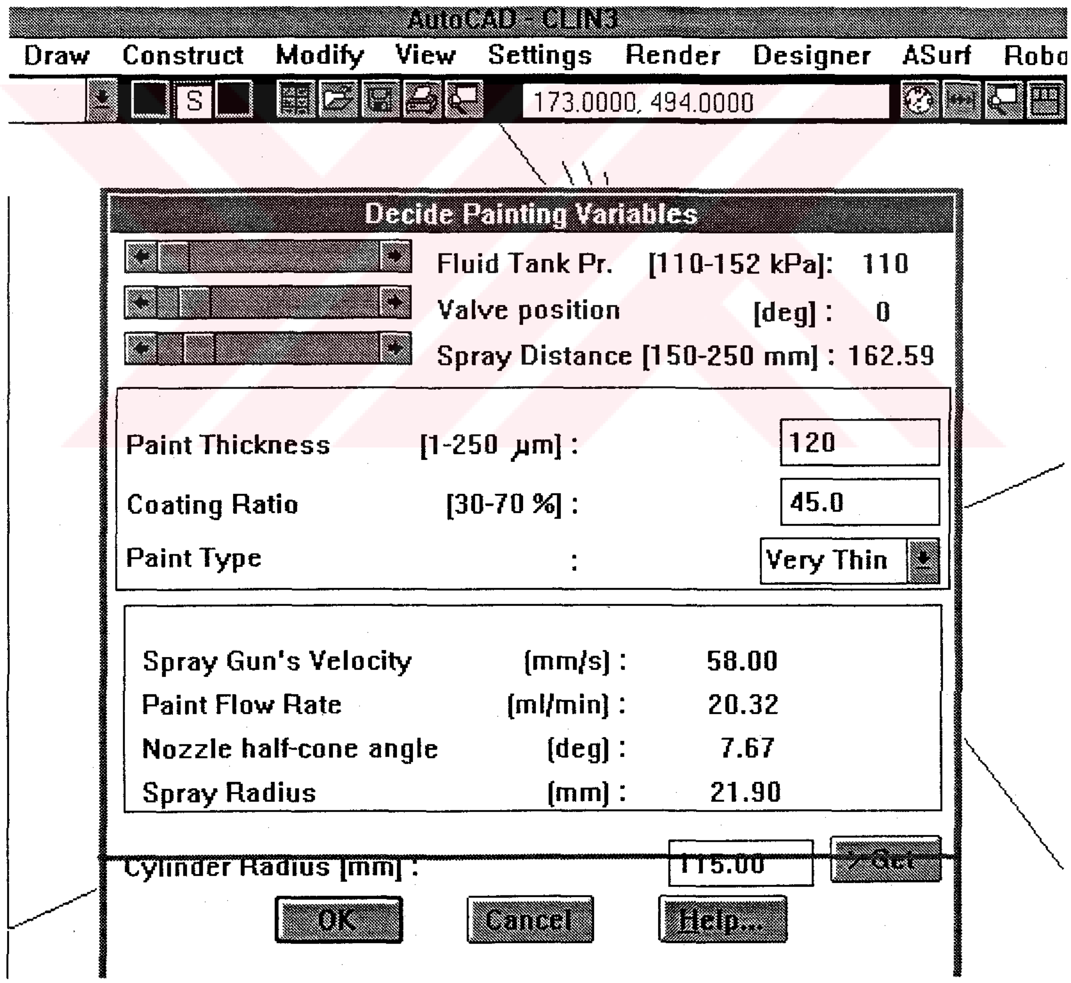


Figure 6.12 Spray Variables Dialog Box for Cylindrical Surface.

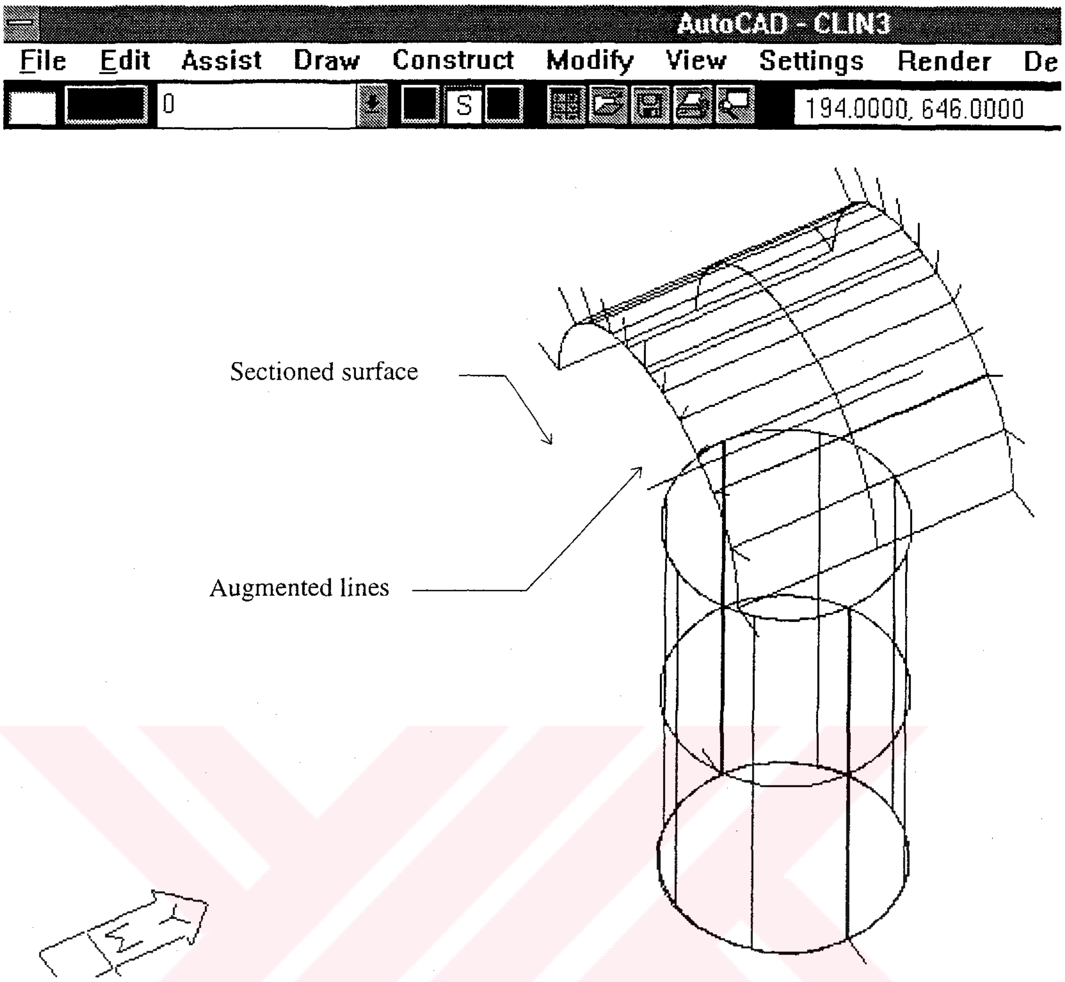
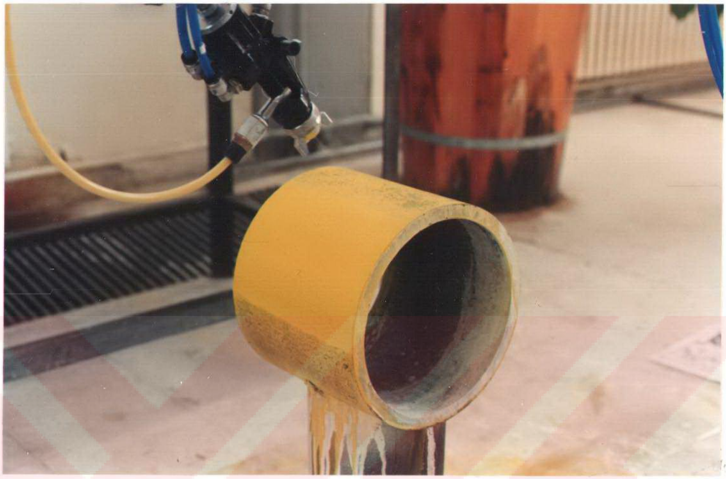


Figure 6.13 Sectioning of Half of a Cylinder with Output Aug. Line Option.

After entering input and output file names, by selecting the 'Linear' post process option the robot program file is constructed.

To spray paint the surface, the output text file is downloaded to the Fanuc Robot. Before running the program paint tank pressure, spray gun's valve position are set to the decided values and the work piece is placed at the same position and orientation as in the drawing. First work coordinate setting is used while running. The cylindrical work piece is shown in Figures 6.14 while spray painting process is performed.



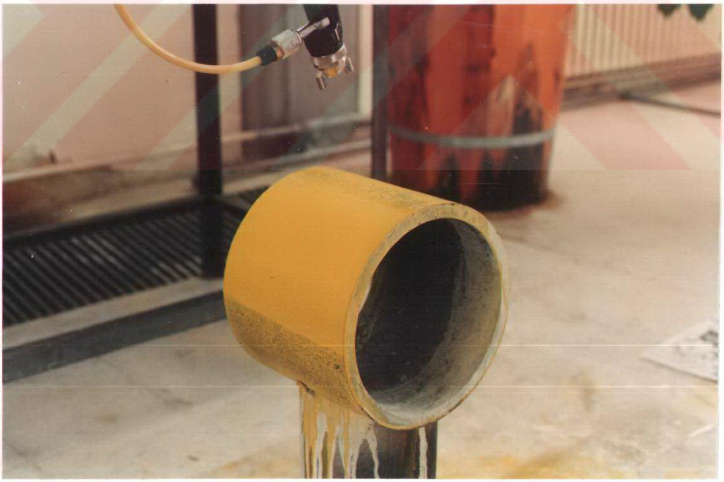


Figure 6.14 Spray painting of half of cylindrical surface, two sequential paths.

6.3 Spray Painting of a Planar Surface

A planar surface can be created by using 'Create>Surface>Planar' option in AutoSurf menu as shown in Figure 6.15.

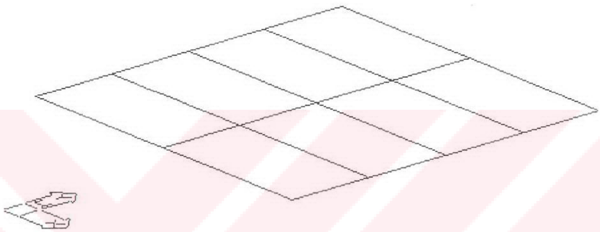


Figure 6.15 Planar Surface.

After the creation of the surface, if painting condition is not decided, by selecting 'Spray Variables' item in Robopaint menu as shown in Figure 6.16 the dialog box to decide painting condition is invoked as shown in Figure 6.17.

Having decided the painting condition, to decide the number of surface lines or the sectioning step value for the planar surface 'Surface Lines>Planar' item in Robopaint menu is selected. The edge length that is perpendicular to the painting direction has to be entered. Then, the sectioning step length and the number of surface lines that are required for equally placed augmented lines are given in a dialog box.

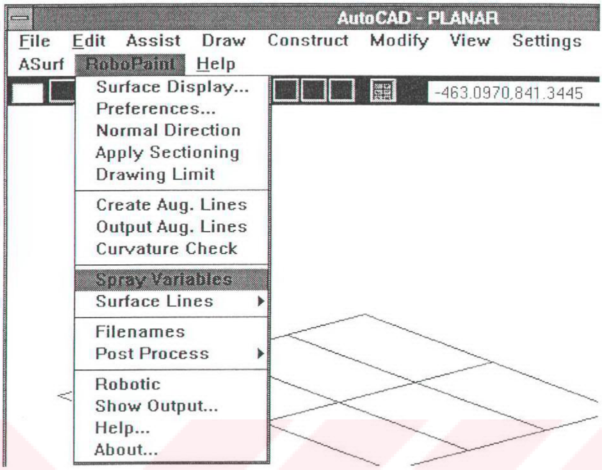


Figure 6.16 'Spray Variables' menu item.

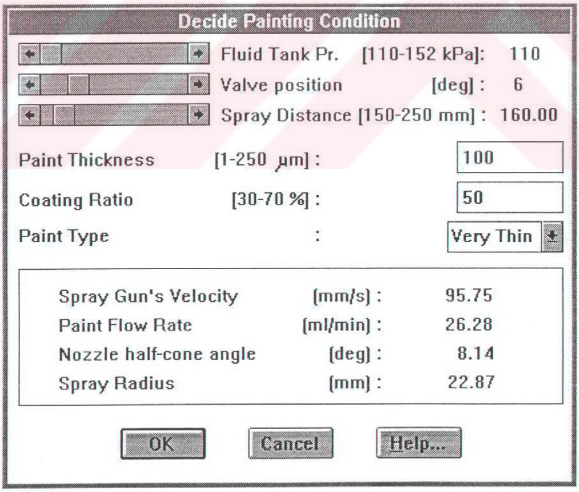


Figure 6.17 The dialog box to decide painting condition.

Either surface lines or sectioning can be used to extract the surface data. To make sections of the planar surface, 'Apply Sectioning' item is selected. In the command line the procedure follows;

Command : _**sectsf**
To invoke advanced options enter "??"...
Select surfaces : ??
(select Output Augmented Line option from the dialog box)
Select surfaces : (pick the surface in Figure 6.15)
Select surfaces : (end selection)
Selection type Radial/<Plane> : (press return for plane option)
Plane by Entity/Last/View/Xy/Yz/ZAxis/Zx/<3 points> : **z**
Point on ZX plane : **200,200,0** (for the surface in Figure 6.15 corners '200,200 and 600,600')
Start Value <0> : (press return)
End Value <0> : 400 (edge length.)
Step Value <0> : 23.5294 (value given after 'Surface Lines' selection.)
Group name prefix <ASG> : (give a group name)

Augmented lines are created by this procedure as in Figure 6.18.

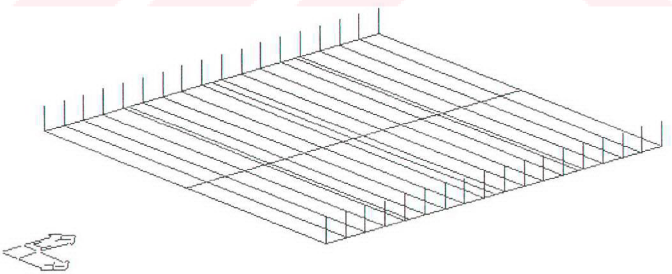


Figure 6.18 Sectioned planar surface with augmented lines.

The surface data is extracted after augmented lines are created by selecting 'Output Aug. Lines' item. When the selection of the augmented lines ends, a dialog box is invoked to enter the data file name. This data file must be selected after 'I/O Filenames' option is invoked just before post processing.

Having extracted the surface data and defined the data file name, by selecting 'Post Process>Linear' item in the Robopaint menu as shown in Figure 6.19 the robot program text is constructed for downloading to the robot. The planar surface is shown in Figure 6.20 during spray painting process.

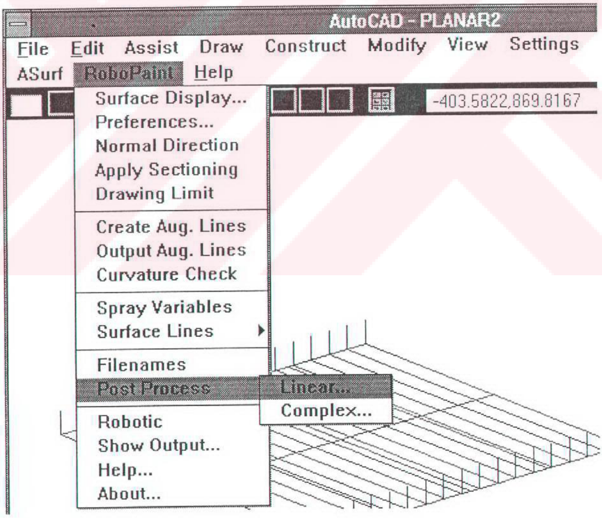


Figure 6.19 Post Process options.

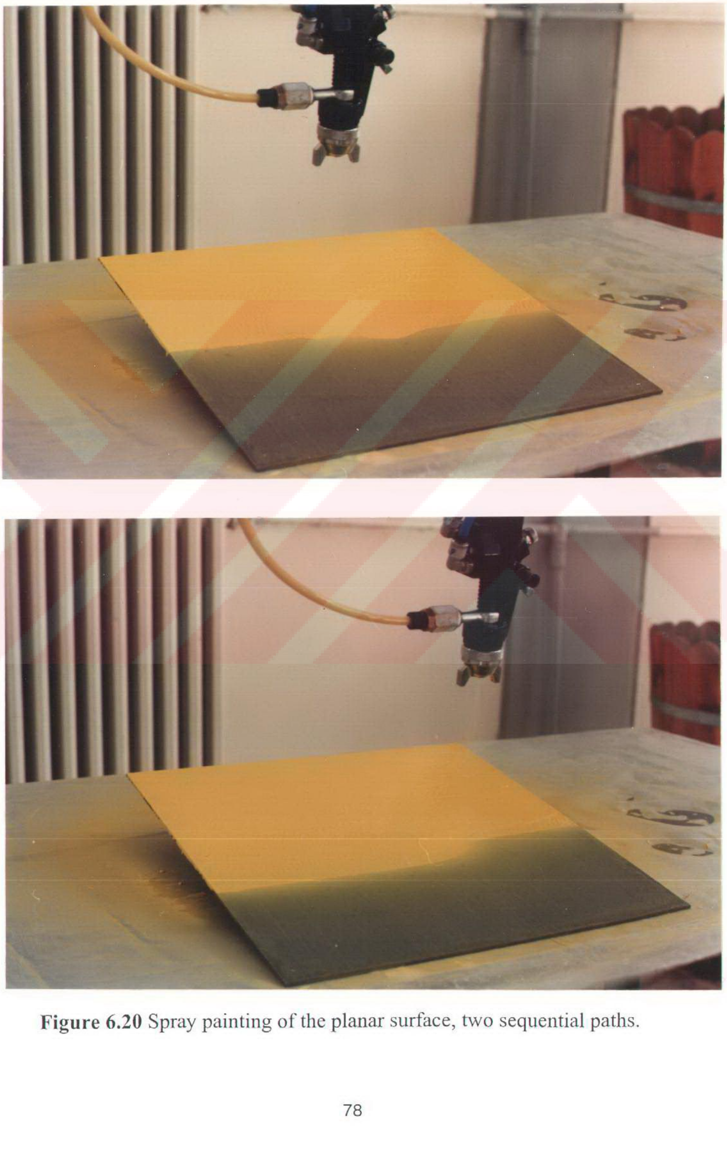


Figure 6.20 Spray painting of the planar surface, two sequential paths.

CHAPTER 7

CONCLUSION

7.1 The Developed Software and Fanuc Robot

In this thesis, a software is developed for spray painting process of an industrial robot. The software is developed for Fanuc Arc Mate Sr. with R-G2 Controller environment for low-level programmers.

Software runs under AutoCAD Release 12 environment and it requires AutoSurf auxiliary for surface manipulations. Surfaces created in AutoSurf can be painted by the software's output robot language text written for the Fanuc Robot's Controller.

The developed software runs perfectly as far as the AutoCAD Release 12 and Fanuc Robot's Controller are considered. The main problem occurs in the complex surfaces' data supplied by AutoSurf. Complex surfaces are sectioned in order to extract the surface data. That causes a difference in spray radius on the surface between two sequential path lines due to the curvature of the surface.

In spray painting processes, surface equation is the main input required for the planing of the path of the spray gun. If surface equation is supplied, spray radius and spray gun motion path can be calculated and decided more easily even in complex surfaces. In AutoSurf, if two opposite boundary lines, from which surfaces are created, are at different dimensions, spray radius changes along an AL. That brings us to select an average spray radius which is undesired.

In addition, in AutoSurf surface data for trimmed surfaces can't be extracted from surface lines. This problem is partially solved by using 'sectsf' command.

One of the deficiencies comes from using the work coordinate in programming the Fanuc Robot. 'Impossible configuration' problem occurs while working with work coordinate system. It is related with G97 Cartesian coordinate input code. 'Impossible configuration' means that the robot can not move to the target point with the desired hand position and orientation. That may be because of the position which is out of the reachable range in any hand orientation, or the hand orientation is impossible for that position.

7.2 Future Development of The Software

By using the same post processing algorithms, post processors can be written for different robot controllers. Hence, the software should not be thought as a limited one with the Fanuc Robot R-G2 Controller.

Surfaces created by other drawing programs can be transferred to AutoCAD environment. Therefore, surface creation in other programs should not be a problem.

On the other hand, in any way, if there is an other program for creation and surface data extraction of surfaces, using the same procedure not only in post processing but also in AutoCAD environment, a software with its own environment can be programmed easily.

Off-line simulation is the most important feature in the process of design and optimization of robot cells and fast evaluation of robot programs. It is a must to have a simulation program for any type of process for an industrial robot. Robopaint creates a robot program file for spray painting of the desired surface but it doesn't ensure that the robot can reach every target point in the program blocks. Therefore, it needs an off-line simulation program.

The calculations for spray radius, spray gun's velocity, cone angle and flowrate are all dependent on the fluid tank pressure and nozzle opening valve. However, nozzle opening valve on the spray gun was not calibrated.

The spray gun's nozzle opening valve (fluid needle stroke adjustment) is calibrated to find these relationships. Results showed that the nozzle opening valve and paint tank pressure effects the spray pattern's cone angle and the flowrate. However, only a limited portion of the valve is calibrated. In determining the painting condition, therefore, small spray gun velocity values can't be reached. Calibration of the nozzle opening valve should be finished for small values of the nozzle opening valve position.

Experiments were conducted by using water as paint. Therefore, for a better surface coating these experiments must be performed with paint whose type is categorized as very thin. Similar experiments must also be performed for different viscosity fluids in terms of medium and heavy type with appropriate fluid and air nozzles available.

Finally, coating thickness experiment should be carried out not only to check the surface coating uniformity but also to check the theory used to calculate the spray gun's velocity. Also, the experiments should be performed for different painting conditions to see the maximum spray gun's velocity allowing good coating uniformity.

REFERENCES

[1] Suh, S.H., Woo, I.K., Noh, S.K., *Development of An Automatic Trajectory Planning System (ATPS) for Spray Painting Robots*, Proceedings of the 1991 IEEE, pp.1948-1955, 1991.

[2] Duelen, G., Stahlmann, H.D., and Liu, X., *An Off-Line Planning and Simulation System for the Programming of Coating Robots*, Annals of CIRP , 38, pp.369-372, 1989.

[3] Ngo, P.A.P., Cheever, G.D., Ottoviani, R.A. and Malinski, T., *Development of a Predictive Model for the Changes in Roughness that Occurs During the Painting Process*, Journal of Coatings Technology, 65, pp.29-37, 1993.

[4] Anon, *Spraying Their Way to Success*, Industrial Robot, 16, pp.13-16, 1989.

[5] Schreiber, R.R., *Portrait of a Painting Robot*, Manufacturing Engineering, 107, pp.55-61, 1991.

[6] Ueno, T., Kajioka, Y., Sato, H., Yoshida, T., and Tsutsui, R., *Automation in Construction*, Robotics, 6, pp.327-335, 1990.

[7] Fanuc Robot Manual, Fanuc Ltd., Japan, 1991.

[8] AutoSurf Reference Manual, Autodesk, Inc., Switzerland, 1992.

[9] AutoLISP Programmers Reference, Autodesk, Inc., Switzerland, 1992.

[10] AutoCAD Customization Manual, Autodesk, Inc., Switzerland, 1992.

[11] Spitzbart, A., Calculus with Analytic Geometry, Scott, Foresman and Company, Glenview, Ill., 765,1969.

[12] Thomas, G.B., JR., Calculus and Analytic Geometry, Addison Wesley Publishing Company, Inc., Massachusettes, USA, 1010, 1961.

[13] Schilling, R.J., Fundamentals of Robotics Analysis and Control, Prentice-Hall, Inc., New Jersey, USA, 425, 1990.

[14] Rogers, D.F., Adams, J.A., Mathematical Elements for Computer Graphics, McGraw-Hill Publishing Co., Singapore, 611, 1990.

APPENDIX A

AUTOSURF SURFACE COMMANDS

Table A.1 The five AutoSurf primitive surface types and the commands to create them.

Surface Type	Command	Description
Cone	PRIMSF, COne	Creates a cone or partial cone.
Cylinder	PRIMSF, CYlinder	Creates a cylinder or partial cylinder.
Sphere	PRIMSF, Sphere	Creates a sphere or partial sphere.
Torus	PRIMSF, Torus	Creates a toroidal tube or partial toroidal tube.
Planar	PLANESF	Creates a planar surface from various input options. The extends of the surface are dependent upon the creation method used. This command also creates a trimmed planar surface from any closed entity that lies on a plane.

Table A.2 The seven AutoSurf free form surface types and the commands to create them.

Surface Type	**Command**	**Description**
Extruded	EXTRUDESF	Creates a surface by extruding (adding depth to) the input curve. The depth can be input directly, or captured from another selected entity. Draft (at an angle) can be created automatically on the extruded walls.
Revolved	REVOLVESF	Creates a surface by rotating a curve about an axis.
Ruled	RULESF	Creates a surface between two input entities. The surface is characterized by "straight elements" between and held *Normal* to, each input curve. This type of surface is sometimes called a "ribbon" surface.
Swept	SWEPTSF	Creates a surface from a set of cross section lines that are swept along one or more "rails". Two or more cross sections (at different positions along the rails) may be used as input. The surface is then created by blending between the cross sections.
Lofted	LOFTSF	Creates a surface from any number of curves that are spatially parallel and do not cross each other. The surface is "fit" through the input curves.
Meshed	MESHSF	Creates a surface from two sets of any number of curves that are spatially parallel and do not cross each other. Each set of lines must cross the other set of lines, and may be "parametrically square".
Tubular	TUBESF	Creates a variable diameter tubular surface around a selected line that becomes the center of the tube.

Table A.3 The four AutoSurf derived surface types and the commands used to create them.

Surface Type	**Command**	**Description**
Blended	BLENDSF	Creates a surface that smoothly joins 2, 3, or 4 existing surfaces or curve entities, and is tangent to each parent surface.
Fillet	FILLETSF	Creates a surface that rounds sharp corners. Fillet surfaces have arcs of the same radius for its cross sections, with the arc cross sections tangent to two cross sections.
Corner (Ball)	CORNERSF	Creates a blended fillet at the intersection of three fillets. This allows different radii at each surface edge.
Offset	OFFSETSF	Creates a surface by offsetting a given surface by a value. The offset is applied normal (perpendicular) to the original surface.

APPENDIX B

SPRAY GUN CHARACTERISTICS

The experiments are performed to calculate the relationship between the paint tank pressure, nozzle opening valve position and paint flowrate.

Also, similar experiments are conducted to find the relationship between the pattern diameter and the paint tank pressure, and nozzle opening valve position.

According to the paint type, air and fluid nozzles are selected and mounted to the spray gun as shown in Table B.3. Water is used as paint during experimentation and the air and fluid nozzles are selected accordingly.

It is seen that paint tank pressure has a small effect on the cone angle and the paint flowrate, whereas valve position effects the cone angle and the paint flowrate considerably.

The cone angle and the paint flowrate increase linearly as paint tank pressure and valve position increase. Therefore, linear curve fitting is used to formulate the relations between them. Interpolation is used to obtain the flowrate and the cone angle for the other paint tank pressures and valve positions, when programming the 'Spray Variables' dialog box.

The results are shown in the following pages.

Table B.1 Flowrate data set obtained for the calibration of the spray gun.

	Tank Pressure (kPa)			
Valve (deg)	110	124	138	152
-6	16.3	18.8	20.6	22.5
0	20.3	23.9	26.4	28.9
6	25.4	29.0	32.2	35.3
12	31.3	35.0	38.5	42.0
18	37.2	41.0	44.8	48.7
24	43.1	47.0	51.2	55.3
30	49.0	53.0	57.5	62.0
36	57.5	61.8	66.5	71.7
42	62.3	67.1	71.9	76.7
48	68.0	73.1	78.7	84.3
54	75.0	80.7	86.7	92.7

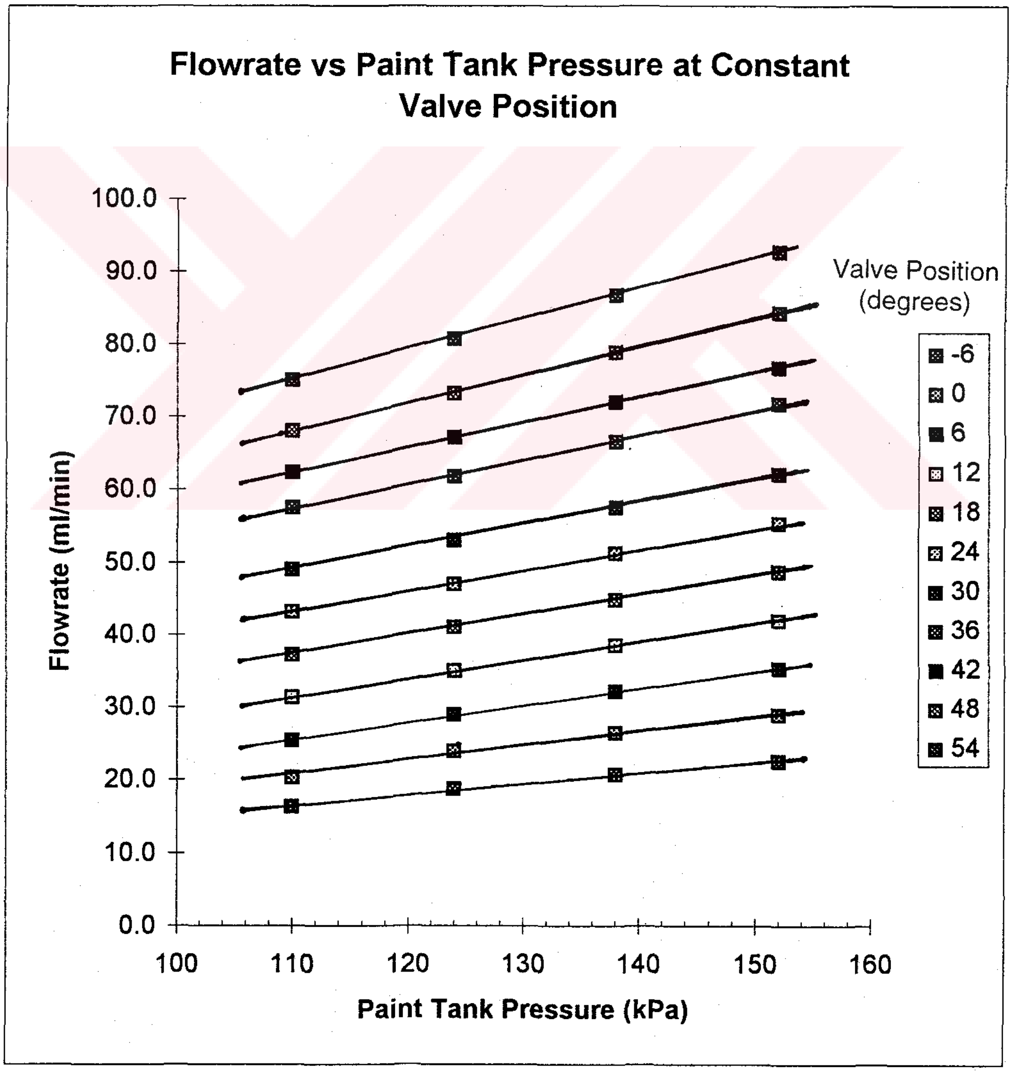


Figure B.1 Flowrate vs. Paint Tank Pressure graph.

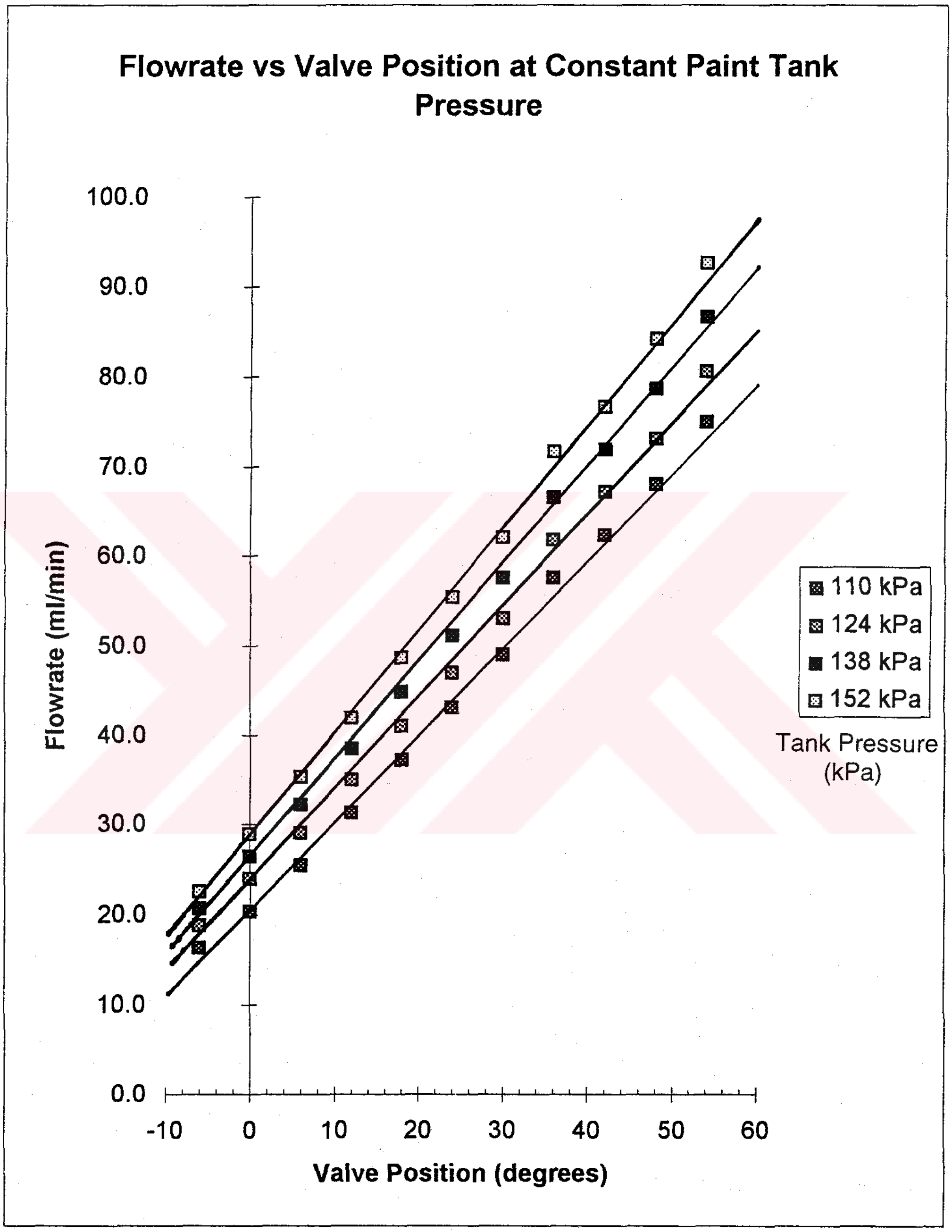


Figure B.2 Flowrate vs. Valve Position graph.

Table B.2 Spray pattern diameter at a spray distance of 20 cm for various painting condition.

	Tank (kPa)			
Valve (deg)	**110**	**124**	**138**	**152**
-6	50.50	51.00	54.90	56.30
6	58.65	59.80	62.70	65.75
18	63.00	64.30	66.10	72.80
30	69.00	71.60	74.70	78.80
42	76.70	78.00	79.80	83.60
54	86.50	87.50	88.00	88.50

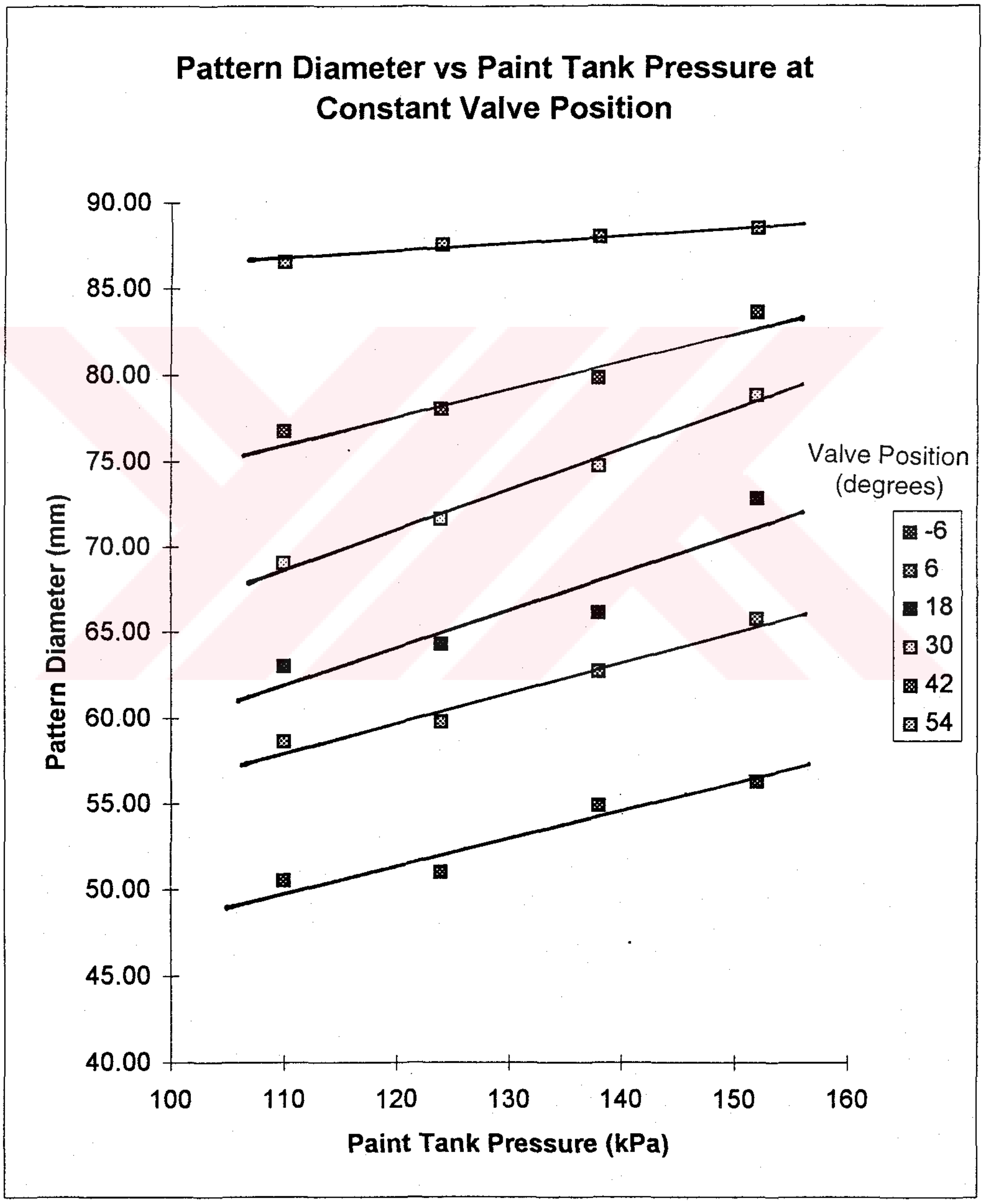


Figure B.3 Pattern Diameter vs. Paint Tank Pressure graph.

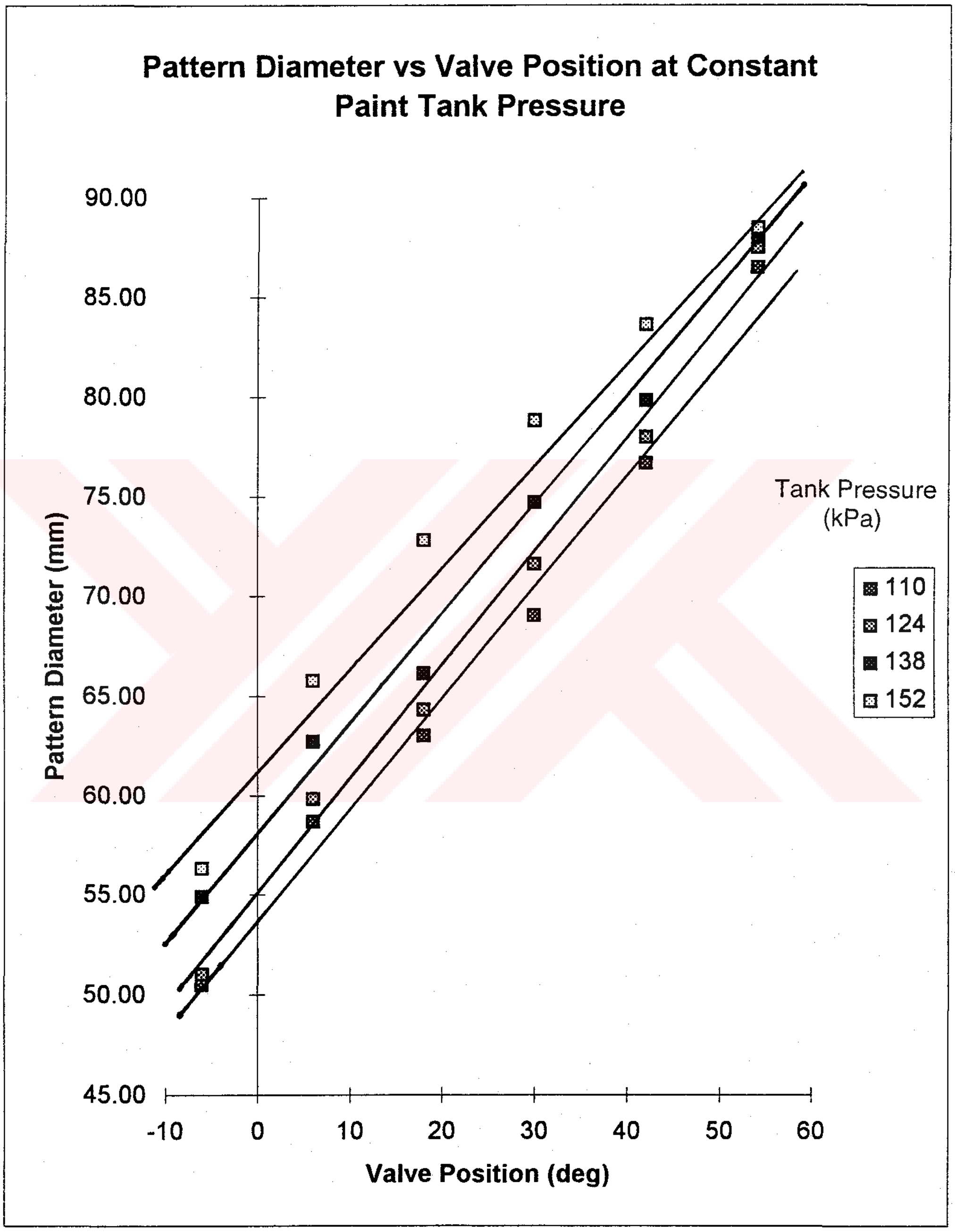


Figure B.4 Pattern Diameter vs. Valve Position graph.

Table B.3 Nozzle and needle selection.

TYPE OF FLUID TO BE SPRAYED	Fluid X Air Nozzle available	Nozzle Type	Max. Pattern at 25 cm	Fluid Needle
VERY THIN 14 -16 secs -- No 2 ZAHN WASH PRIMERS DYES STAINS SOLVENTS WATER INKS	63BSS X 63PB	PE	35.5 cm	763A
MEDIUM 19 -30 secs -- No 2 ZAHN SYN. ENAMELS VARNISHES SHELLACS FILLERS PRIMERS EPOXIES URETHANES LUBRICANTS WAX EMULSIONS	65SS X 63PK	PE	38 cm	765
HEAVY (CREAM-LIKE) OVER 28 SECS -- No 4 FORD HOUSE PAINT WALL PAINT (OIL-LATEX) BLOCK SEALERS MILL WHITES VINYLS ACRYLICS EPOXIES	66SS X 63PB	PE	35.5 cm	765

APPENDIX C

USER'S MANUAL

1. Installing Robopaint Software

To run the Robopaint in AutoCAD environment, first of all the Robopaint menu should be added to the menu which is named "asurf.mnu". The source code for the Robopaint menu is :

```
***POP12
[RoboPaint]
[Preferences...]^C^C_surfvar
[Surface Display...] ^C^C_dispsf
[Normal Direction] ^C^C _normalsf
[Drawing Limits] ^C^C splimits
[--]
[Spray Variables...]^C^Cspvars
[->Surface Lines]
  [Planar]^C^Csplane
  [->Half Cylinder]
   [Radius...]^C^Cspcyln
   [<-Height...]^C^Cspclin
  [Half Cone]^C^Cspcone
  [<-Half Sphere]^C^Csphere
[--]
[Apply Sectioning] ^C^C _sectsf
[Create Aug. Lines] ^C^C_createaug
[Output Aug. Lines] ^C^C _outputaug
[--]
[I/O Filenames...]^C^Cfilenames
[->Post Process]
```

```
[Linear...]^C^Cinorder;postproc1
[<-Complex...]^C^Cpostproc2
[G97 Configuration...]^C^Crobotic
[--]
[Show Output...]^C^Cshowit
[About...]^C^Cspabout
[Help...]^C^C(acad_helpdlg "robopain" "")
```

Table 1. Source files used by Robopaint.

Used by menu item	*Dialog file (.dcl)*	*Lisp file (.lsp)*
Drawing Limit		splimit
Curvature Check	spinfo1	spradchk
Spray Variables	spvars	spvars
Surface Lines/ Plane	spinfo3	spslpla
Surface Lines/ Cylinder/ Radius	spinfo4	spslcyl
Surface Lines/ Cone	spinfo5	spslcon
Surface Lines/ Sphere	spinfo6	spslsph
Surface Lines/ Cylinder/ Height	spinfo7	spslcln
G97 Configuration	sprobo	sprobo
I/O Filenames		spfile
Post Process/ Linear	spinfo2	sprep,spostpr1
Post Process/ Complex	spinfo2	spostpr2
Show Output	spshow	spshow
About	spabout	spabout
Help		robopain.hlp

All Robopaint files given in Table 1 should be in the “SUPPORT” directory or in one of the library directories of the AutoCAD. These lisp files must be loaded while starting to a new AutoCAD session. For automatic loading of these files, following lines must be added to the "acadr12.lsp" file.

```
(autoload "spabout" '("spabout"))
(autoload "spradchk" '("radiuscheck"))
(autoload "spvary" '("spvars"))
(autoload "spslpla" '("splane"))
(autoload "spslcyl" '("spcyln"))
(autoload "spslcln" '("spclin"))
(autoload "spslcon" '("spcone"))
(autoload "spslsph" '("sphere"))
(autoload "spfile" '("filenames"))
(autoload "sprep" '("inorder"))
(autoload "spostpr1" '("postproc1"))
(autoload "spostpr2" '("postproc2"))
(autoload "sprobo" '("robotic"))
(autoload "spshow" '("showit"))
(autoload "splimit" '("spviewport"))
```

Because AutoSurf is needed for surface data handling, it must also be loaded if it is not automatically loaded at the beginning of the AutoCAD session. Related file is usually named as "asurf.exp".

2. Surface Creation in AutoSurf for Spray Painting

There are five primitive surface types in AutoSurf, which are *Cone, Cylinder, Sphere, Torus,* and *Planar*. Although they are primitive, only half of them can be painted easily except for planar surfaces. If all of the surface is tried to be spray painted, then most probably there occurs a problem of reaching for the robot. For the first four type surfaces one thing should be noticed which is the minimum radius of curvature of the surface. In spray painting process, most of the problem comes from the curvature of the surface. Surfaces with small radius of curvatures

Seven AutoSurf free-form surface types exists. These are *Extruded, Revolved, Ruled, Lofted, Meshed, Swept*, and *Tubular* surfaces. When EXTRUDESF command is used, first entity which is picked is the shape which will be extruded. Then the direction and magnitude is indicated by the second entity.

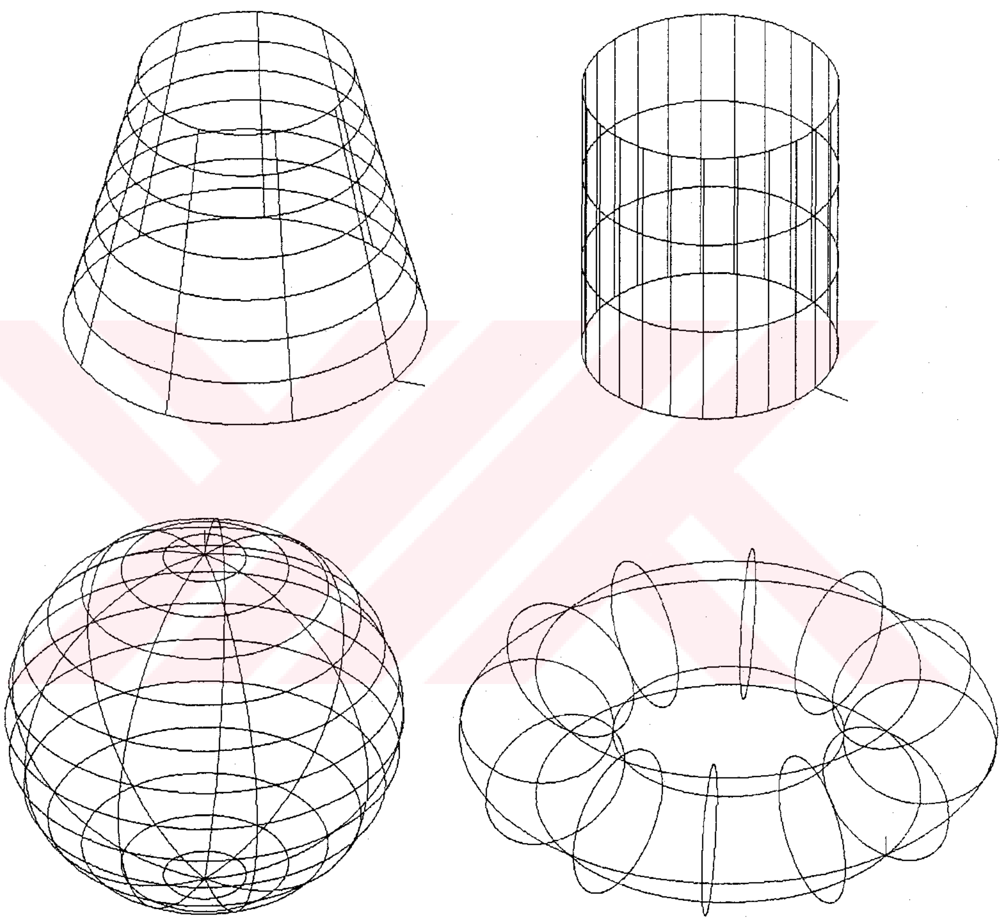


Figure 1. Primitive surfaces in AutoSurf.

REVOLVESF command is used to create a surface of revolution. In that case the first entity picked is the shape to be revolved and the second is the axis of the revolution, such as a Spline or Line.

A ruled surface can be created by any combination of two Lines, Polylines, Arcs, or Splines. Ruled surface uses a straight edge to blend from one shape to another. Many surfaces are created from ruled surfaces.

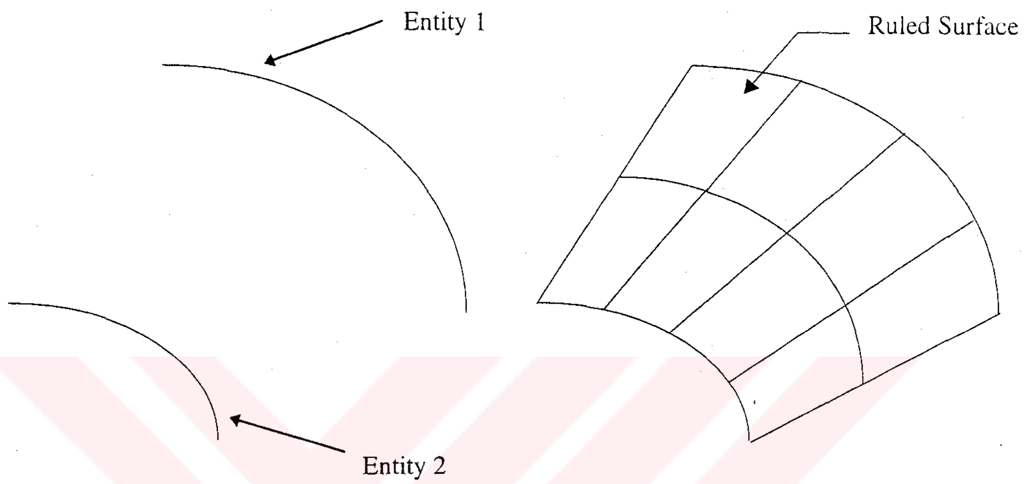


Figure 2. Ruled surface creation.

LOFTSF creates a Surface from U lines. The U lines can be any combination of Lines, Arcs, Splines, Augmented lines, or Polylines. At least two U lines must be picked. The lines must be selected in the order. A skipped U line should not be selected. Also, U lines should not cross each other.

MESHSF creates a Surface from two sets of input lines. The set can be any combination of Lines, Arcs, or Splines and are referred as U and V lines to indicate their opposing directions. Each set of lines defines the bounds and contours of the Surface in its direction. The lines in each set should be consistently spaced (nearly parallel), not cross, and contain no undesired inflections. The two sets of lines must be orthogonal, that is, each line in one set must cross every line in the other set.

SWEPTSF is used to create a surface from a set of cross section lines that are swept along one or more rails. If two or more cross sections are selected, the Surface is created by blending between the cross sections. To do this, select all the cross sections in the order and select the rail curves.

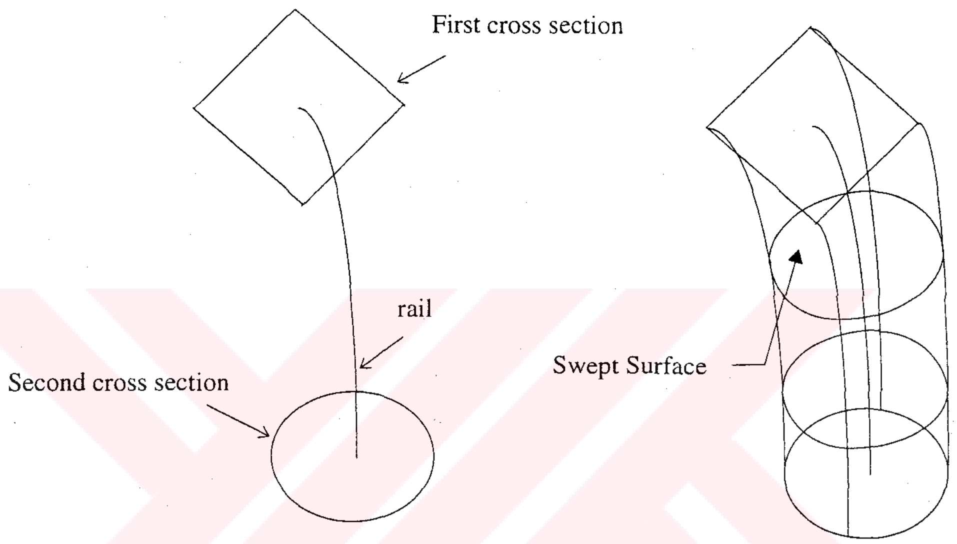


Figure 3. Creating a swept surface.

TUBESF creates a tubular Surface around a selected Line that is defined as the axis of the tube, or a series of Surfaces that form a tubular Surface around a Polyline that is defined as the axis of the tube.

By using these commands many surfaces can be created easily. Also, by using advanced options for these commands, different surfaces can be created even by using the same entities.

Spray painting process becomes easier if the surface is close to a planar surface. If the surface is a circular one then a problem of painting occurs while painting along the path perpendicular to the center line. Surfaces that have a large curvature values, which means radius of curvature is small, may not be painted due to the availability of the spray pattern radius in the range of the recommended spray distance. Therefore, if a surface is not a planar surface then a curvature check must be performed for the surface. For that reason, first of all, a surface data of the surface must be taken by using surface lines. Hence, augmented lines must be created on the surface lines.

3. Some of the Auxiliary AutoSurf Commands

There are a few AutoSurf commands which are used not to create the surface but to effect the surface's environment in terms of surface line numbers, surface normal direction and number of points on a surface line.

DISPSF command is usually used to change the number of u and v-lines of the surface. This is required, if surface lines is used for surface data extraction. 'Display Surface' item in Robopaint menu uses this command.

SURFVAR command shows a dialog box for setting surface variables such as augmented vector length and system tolerance. When working in different scales, augmented vector lengths should be set accordingly in order to pick and see them easily. The accuracy of the all AutoSurf outputs is controlled with system tolerance setting. It is better to set system tolerance value in the range 1 and 5 for Robopaint. If it is set outside this range, the number of vectors on augmented lines becomes neither too few nor too many in order to describe the surface for spray painting process. Besides, smaller or bigger system tolerances causes error in the calculation of the radius of curvature of the surface. 'Preferences' item in the Robopaint menu uses this command.

NORMALSF command is used to change the surface's normal vector direction. Normal vector indicates the surface's upper side which means the side to be spray painted. In creation of cylindrical surfaces, normal vector shows always the inside of the cylinder. Therefore, direction for these surfaces should be changed unless the inside of them is spray painted. 'Normal Direction' item in Robopaint menu invokes this command.

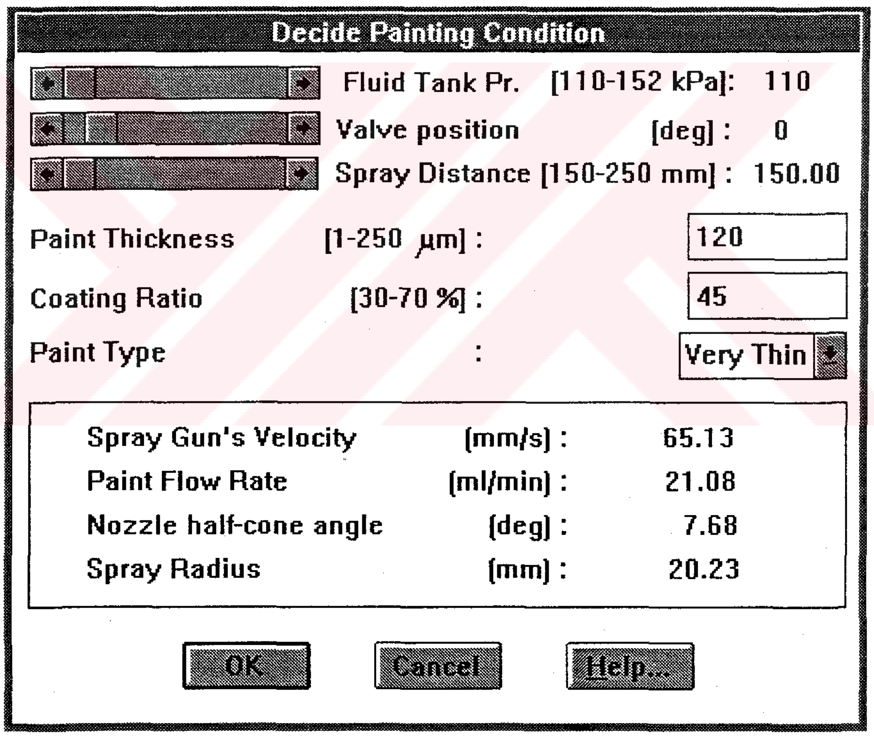


Figure 4. Determination of the Painting Condition with Spray Variables Dialog Box.

4. Deciding Painting Condition and Surface Line Number

Spray gun's velocity is an important parameter in spray painting process. It is dependent on the paint flowrate, the desired paint thickness and the spray pattern's radius. Paint flowrate depends on the spray gun's nozzle opening valve position and the paint tank pressure. To take the loss of the paint into account, paint flowrate is multiplied with a constant value called ratio of the paint coated on the surface. This ratio is about 50 percent for a regular spray gun. Spray pattern's radius is dependent on the half cone angle and the spray distance. The spray pattern's cone angle changes also with the paint tank pressure and the nozzle opening valve position.

'Spray Variables' item invokes a dialog box to decide the painting condition as shown in Figure 4.

The output of this dialog box is the spray gun's velocity which is required to supply the desired paint coating thickness under the given painting condition in terms of paint tank pressure and valve position.

The recommended values for the spray distance is 20 cm in spray painting process. In the dialog box, spray distance can be decided by a slider in the range 15 and 25 cm. Paint tank pressure value is decided similarly in range 110 kPa and 152 kPa with an increment of 7 kPa. Valve position value changes from -6 to 54 degrees with an increment of 3 degrees. The ranges and incrementation for paint tank pressure and valve position values are chosen according to the calibration of the paint tank and spray gun's nozzle opening.

Coating ratio value shows the ratio of the paint coated on the surface. This ratio is about 50 % with a regular spray painting gun. It must rely on the experimentation with different paint types and painting condition. However, there isn't any relation for now.

Paint type shows the fluid type which will be used in the spray painting process. This is required to decide air and paint nozzle that will be mounted to the spray gun. Experimentation for fluid flow rate is performed only for 'very thin' type of paints. Therefore, only this option is available in the dialog box.

The four dependent variable is shown in the dialog box as information ; spray gun's velocity, fluid flow rate, half cone angle, and spray radius. However, if complex surfaces are to be spray painted, spray radius should be closely investigated. Because, complex surface are sectioned to extract the surface information and spray radius is used as the step value while making sectioning.

On the other hand, if one of the primitive surfaces is to be spray painted, surface line number which is perpendicular to the path lines, or sectioning step value has to be calculated. For that reason, in Robopaint menu, "surface line>" item with corresponding surface type should be selected as shown in Figure 5. For planar, conical and spherical surfaces, planar surface's edge length, conical surface's height, and radius of spherical surface are required to be entered respectively.

Also, for cylindrical surfaces there may be to input according to the painting direction. If it is decided to paint the cylindrical surface along surface lines parallel to the center line, cylinder radius is needed, otherwise cylinder's height is required to calculate the surface line number or sectioning step value. The result is given in a dialog box which shows the number of surface lines to be set and the sectioning step value as shown in Figure 6.

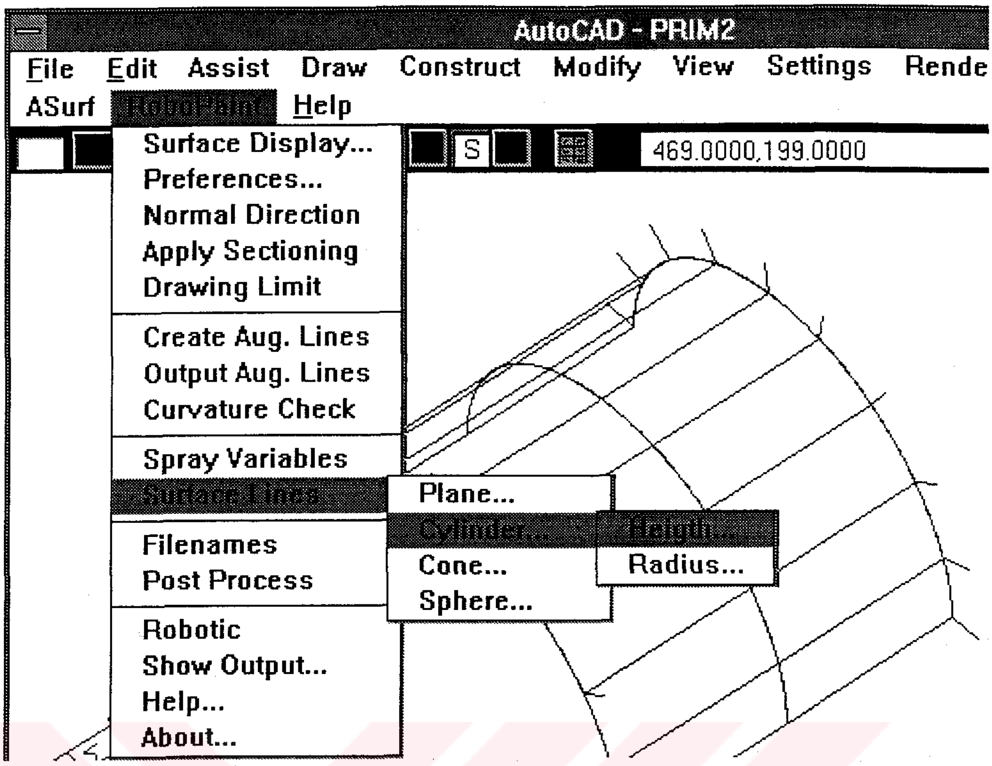


Figure 5. Surface Lines item in the Robopaint menu.

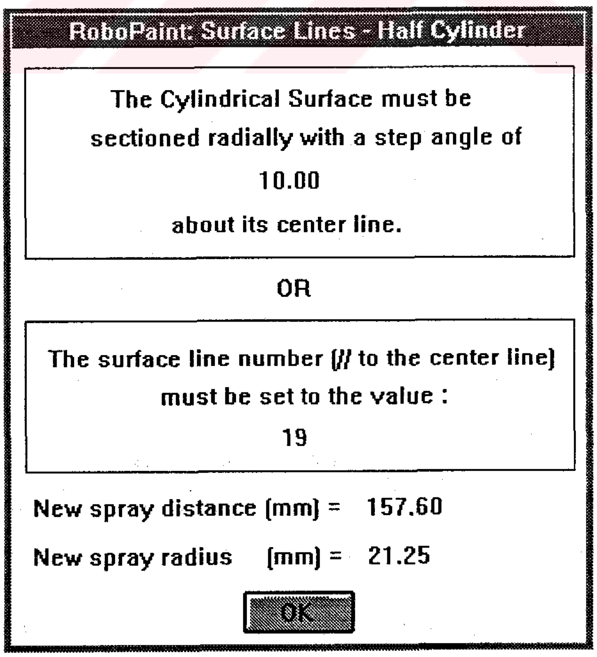


Figure 6. Dialog box showing the result for ‘Surface Line>Radius’ item.

The results given in this dialog box considers always half of the primitive surfaces except for the planar surfaces as shown in Figure 7.

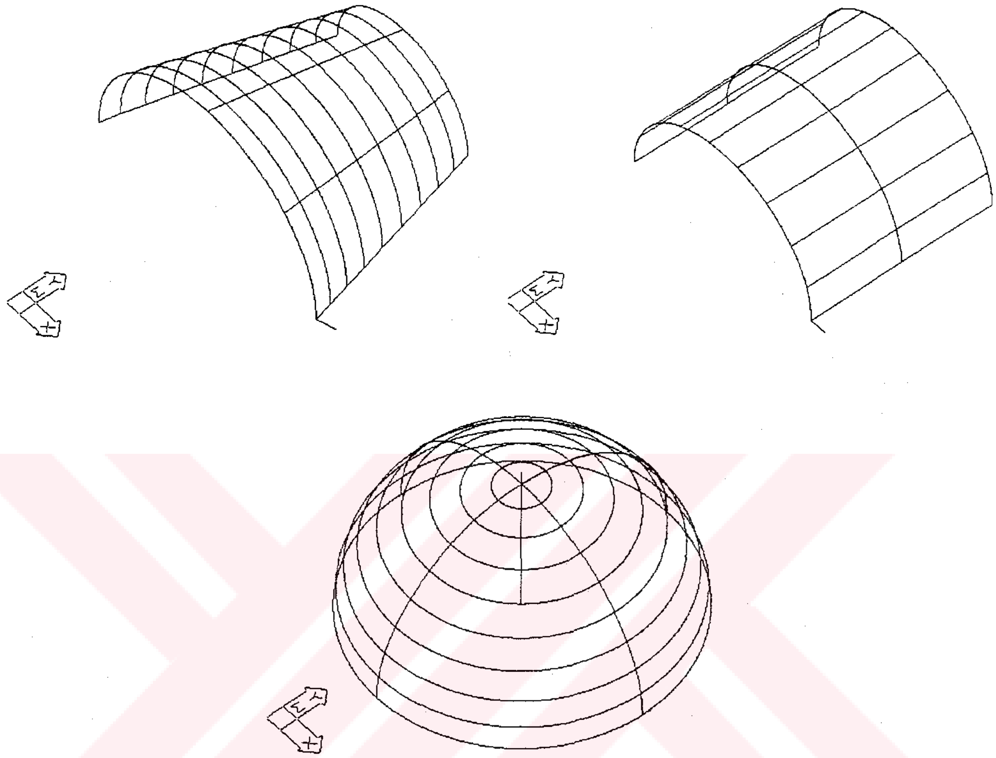


Figure 7. Example of half of the primitive surfaces.

5. *Surface Data Extraction and Sectioning*

When painting condition is decided, the next step is to decide the number of surface lines if the surface is one of the primitive types. After having the sectioning step value and number of surface line information, the next step is to save the surface data into a file. For that reason, first of all augmented lines must be created either by using or by sectioning of the surface.

If surface lines are to be used in extracting surface data, then correct number of surface lines must be set. This is done by selecting 'Display Surface' menu item in Robopaint menu and the surface. To create augmented lines from surface edges, 'Create Aug. Lines' menu item is selected or 'createaug' is entered in the command line. Then, surface lines for which augmented lines are to be created are selected.

Command : **createaug**
To invoke advanced options enter "??"...
Create from wireframe/<Surface edge> : (press return for surface edge option)
Select objects : (select surface edges)
Augmented by Points/Step/<Optimal> : (press return for Optimal option, be sure that system tolerance value is between 1 and 5)

If sectioning is applied to the surface then there is no need to set the correct number of surface lines. To make section either complex or primitive surface, 'Apply sectioning' item in Robopaint menu is picked. To create augmented lines while sectioning, advanced option must be invoked.

Sectioning can be done in to ways in terms of radial and plane. In radial sectioning, a line profile is rotated about Z axis which is placed at a point given by the user.

Command : **sectsf**
To invoke advanced options enter "??"...
Select surfaces : **??**
(select output augmented lines option in the dialog box appeared)
Select surfaces : (Select the surface from which the cross section created)
Selection type Radial/<Plane> : **r**
Select first location : P1 in Figure 7
Select second location : P2 in Figure 7
Start Angle <0> : **0**
End Angle <0> : **180**
Step Value <0> : (step angle given in Surface Lines result dialog box)
Group name prefix <ASG> : (enter a group name prefix)

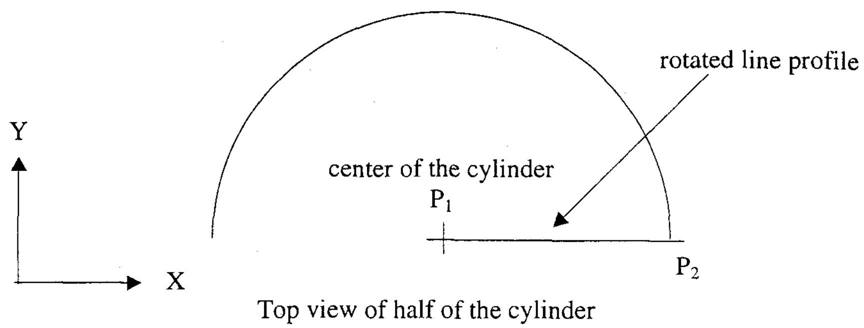


Figure 8. Radial sectioning of a cylindrical surface.

In plane sectioning, a profile is used to cut the section in any plane given by the user.

Command : **sectsf**
To invoke advanced options enter "??"...
Select surfaces : **??**
(select output augmented lines option in the dialog box)
Select surfaces : (Select the surface from which the cross section created)
Selection type Radial/<Plane> : (press return for plane option) **p**
Plane by Entity/Last/View/Xy/Yz/ZAxis/Zx/<3 points> : (enter an option) **x**
X axis start point (0,0,0) : (pick the corner close to the x axis)
Start Value <0> : **0**
End Value <0> : (enter the edge length perpendicular to the path lines)
Step Value <0> : (enter spray radius value)
Group name prefix <ASG> : (enter a group name prefix)

After the creation of the augmented lines, surface data is saved to an ASCII file by selecting the augmented lines. 'Output Aug. Lines' item in Robopaint menu is selected for that purpose or OUTPUTAUG is entered at the command prompt.

Command : **outputaug**
Select augmented lines : **g** (group 'g' is used if sectioning applied otherwise augmented lines must be selected one by one)
Select augmented lines : (pick one augmented lines)
(enter the output data filename that appears after selection ends)

6. Radius of Curvature Check for Complex Surfaces

If the surface which will be painted is a complex surface and the minimum radius of curvature is not known, radius of curvature check for that surface must be made to decide either it can be painted or not. Also, painting direction is decided according to the minimum radius of curvature of the surface. This is done by selecting 'Curvature Check' item in Robopaint menu. Before selecting this item, surface data in two possible painting directions must be saved in two files named as 'u.aug' and 'v.aug' in the 'c:\painting\' directory. These two files are checked by selecting the 'Curvature Check' item. Result of this check is given by a dialog box as shown in Figure 9.

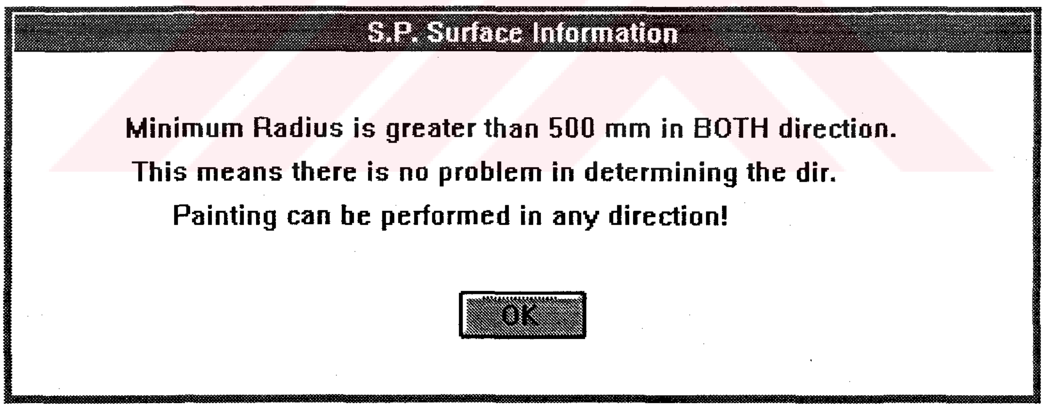


Figure 9. Radius of curvature check result dialog box.

Sectioning can be applied to the surface in two plane as XZ and YZ. If step value is taken as the spray radius decided in spray variables dialog box, then according to the result obtained after radius check one of these files can be used to for postprocessing. Otherwise, one more sectioning is applied with a step value equal to spray radius.

On the other hand, surface lines can also be used to take the surface data for that purpose. However, it requires much work for saving surface data because all of the augmented lines are picked on by one.

7. Post Processing

Before postprocessing, surface data file name which is saved previously and the desired output file name must be entered by selecting 'I/O Filenames' item in the Robopaint menu.

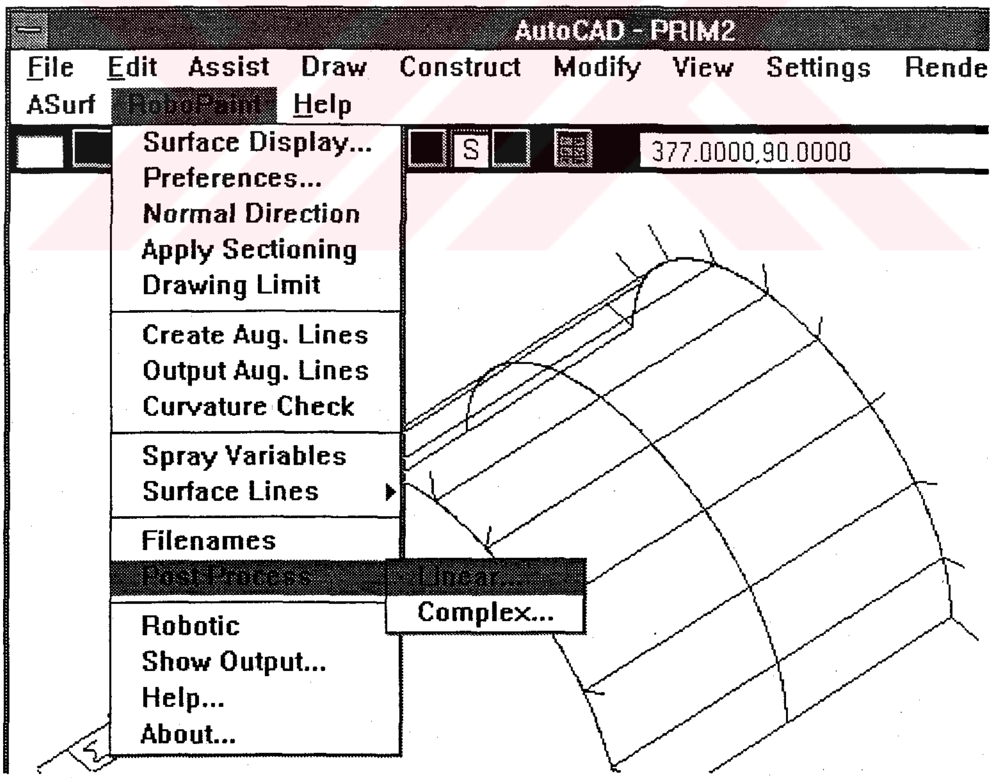


Figure 10. Post Process Options.

There are two post processing option in terms of Linear and Complex as shown in Figure 10. Linear option must be used for planar surfaces and cylindrical surfaces for which the augmented lines are created parallel to the center line. Complex option must be selected for the rest of the surface types.

Before the output robot program text is created, a dialog box is appeared as shown in Figure 11. If 'Yes' button is selected the post processing is made otherwise it is canceled.

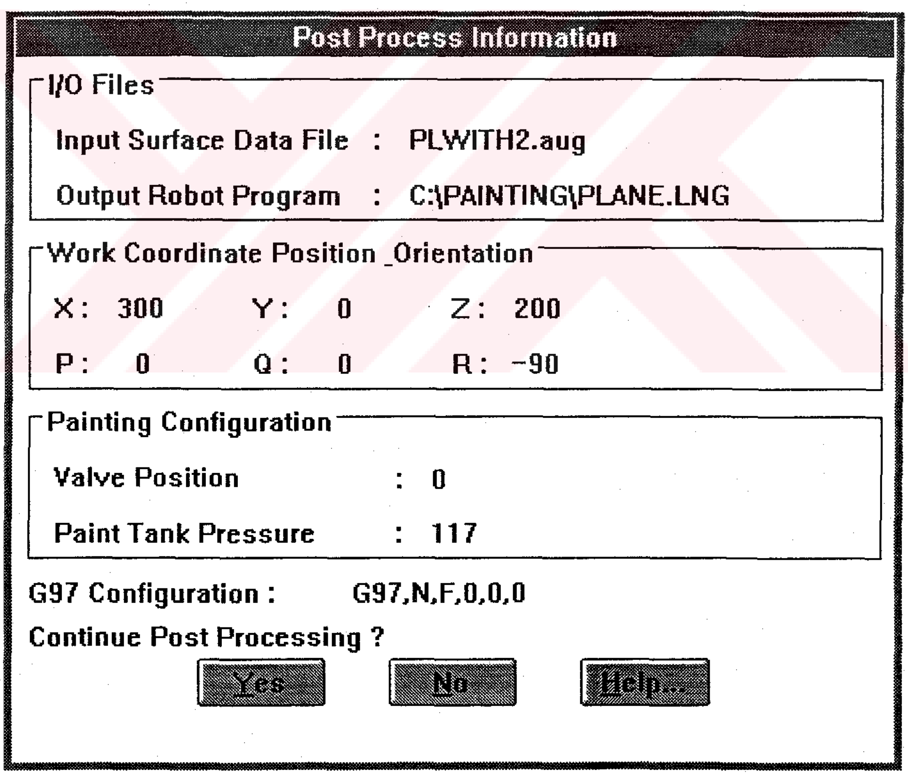


Figure 11. Post process information dialog box.

8. Downloading the Robot Program to the Fanuc Robot

To Download the robot program file to the Fanuc Robot, there is a procedure followed for the computer and the robot side. For the Computer side the procedure is as follows:

1. Copy the robot program file to the "C:\>ROBOT> " directory,
2. Run the program named 'fanuccom.exe' in the "C:\>ROBOT>" directory,
3. Select the send file option,
4. Select program file option,
5. Input the file name and press enter.

The computer is now ready to send the file. By using the teach pendant this procedure must be followed in order to save the file to the robot's memory :

1. Press the function button,
2. Select device option,
3. Select host option,
4. Select load option,
5. Select program option,
6. Enter the program number where the file is to be downloaded.

While downloading there are two important points. There should not be any program having the same program number and there must be enough free-space for the robot program to be saved into the robot's memory. Free-space shows the maximum number of blocks of the program. It can be seen by selecting O-MAP on the teach pendant. If there were not enough memory left for downloading, then some of the programs in the robot's memory should be deleted or saved to the computer's hard disc for backup.