Numerical analysis of the temperature distribution and Offline programming of industrial robot for thermal spraying

Final Thesis

by

Alejandro Frutos

at

Institute for Manufacturing Technologies of Ceramics Components and Composites- IMTCCC
University of Stuttgart

Stuttgart, January 2009

Acknowlegedment

This Final Thesis has been developed in the Institute for Manufacturing Technologies of Ceramic Components and Composites (IFKB, University of Stuttgart).
I would like to sincerely thank Prof. Dr. Rainer Gadow for giving me this opportunity to realize my final thesis at the IFKB as well as Dipl.-Ing. Antonio Candel and Dipl._Ing Esther Dongmo by the interest showed in this work.

I would like to show my gratitude to D. José Conde del Teso as my advisor and D. José Andrés Moreno Nicolás as my teacher in the University of Spain.

Stuttgart, 25th January 2009

Resumen del proyecto

Para asegurar la alta reproducibilidad en el proceso de producción así como las propiedades del recubrimiento, máquinas herramienta avanzadas y sistemas robot son usados para modernas aplicaciones de spray térmico. Los robots industriales pueden desarrollar tareas repetitivas y además ser usados en atmósferas que pueden resultar peligrosas para la salud humana.

Debido a la alta energía cinética y térmica de los gases y las partículas impactando, la trayectoria y velocidad de la pistola automáticamente guiada tiene una importancia crucial en los procesos de transferencia de calor y masa durante la producción de los recubrimientos de spray térmico. Las propiedades como espesor, porosidad, microdureza y la distribución de tensión térmica son influenciadas significativamente por la distancia de spray, velocidad y trayectoria de la pistola. En principio la trayectoria de la herramienta es el parámetro más complejo de calcular en el robot industrial.

La definición de la trayectoria de la pistola es un paso a realizar en la programación del robot, normalmente resulta una tarea laboriosa, ya que es realizada midiendo los puntos en la celda de trabajo del robot mediante el método de teach-in. Así, para el desarrollo de trayectorias del robot para la producción de recubrimientos sobre piezas complejas en un tiempo eficaz son necesarias herramientas específicas de programación off-line.

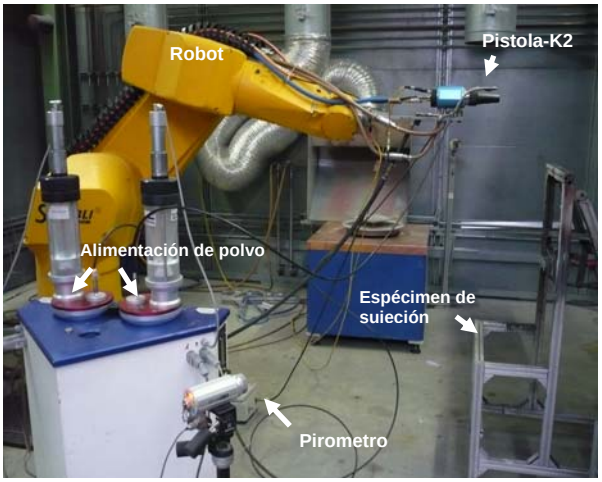


Figura 1 Robot RX 170 para recubrimiento en IFKB

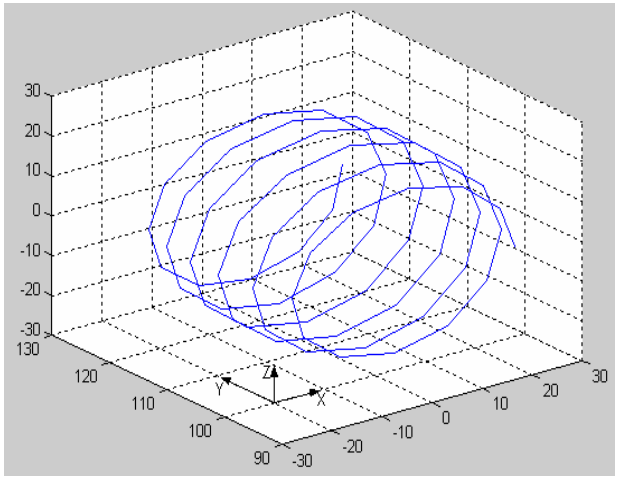


Figura 2 Simulación de la trayectoria

Por medio del apropiado acoplamiento de las trayectorias generadas automáticamente con la cercanía a la realidad de los modelos de elementos finitos de la pieza de trabajo y los procesos de recubrimiento, un flujo de información bidireccional entre la programación off-line del robot y la simulación de la transferencia de calor/masa pueden ser alcanzados.
El trabajo se enfocará en la optimización de este flujo de información para que el proceso de recubrimiento con las trayectorias generadas pueda ser simulado con programas FEM (métodos de elementos finitos) así como ANSYS y ABAQUS. Resultando que las distribuciones térmicas deberían contribuir a la optimización de las trayectorias del robot.

Para obtener la trayectoria de la pistola del robot pueden utilizarse fundamentalmente tres métodos, así como teach-in, ingeniería inversa y diseño asistido por ordenador. Estos tres procesos son utilizados a lo largo del proyecto, pero el que presenta mayor interés es el CAD (Computer Aided Design).

1er Objetivo del proyecto

Generación de la trayectoria del robot a partir de la información geométrica del modelo en CAD de una pieza del tren de alta velocidad (AVE).

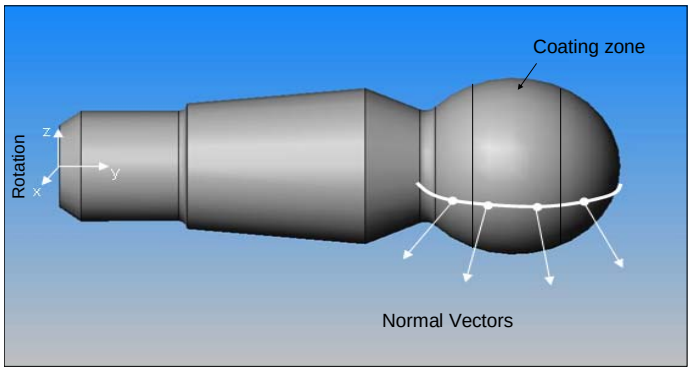


Figura 3 Descripción de la trayectoria y del área de recubrimiento

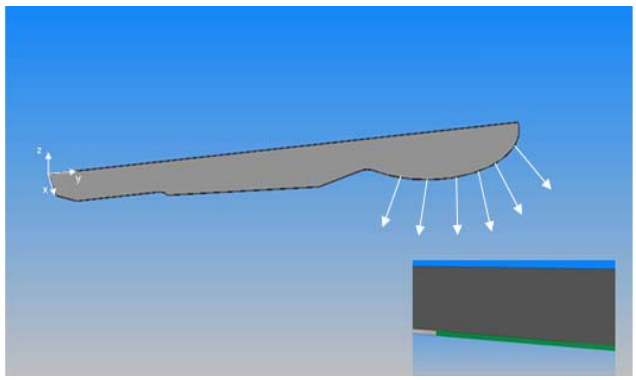


Figura 4 Sección de la pieza

A partir del la generación del modelo en CAD (fig.3), el software aplicado es SolidWorks, es posible realizar las secciones necesarias que definan la trayectoria de la pistola del robot (ver fig.4). Con una única sección es posible definir esta trayectoria ya que esta pieza posee un movimiento de rotación en el eje Y.
No todas las piezas pueden ser definidas con una única sección, por lo general son necesarias varias, aquí se ha desarrollado también otra pieza para probar esta teoría.

La superficie que define el movimiento de la pistola del robot es salvada en el formato STL, que permite realizar un mallado triangular de la misma, entregando las coordenadas de los tres vértices y del vector normal a la superficie del triangulo.

La representación de los vértices correspondientes al mallado de esta superficie puede realizarse por medio de la subrutina 4.4 de Matlab (ver fig.5). Tras un filtrado de los puntos es posible generar la senda del robot como una función de los puntos que componen la sección superficial de la esfera. Las coordenadas de estos puntos están referidas al sistema de coordenadas del software de diseño por lo que es necesario adaptarlas al sistema de coordenadas del robot.

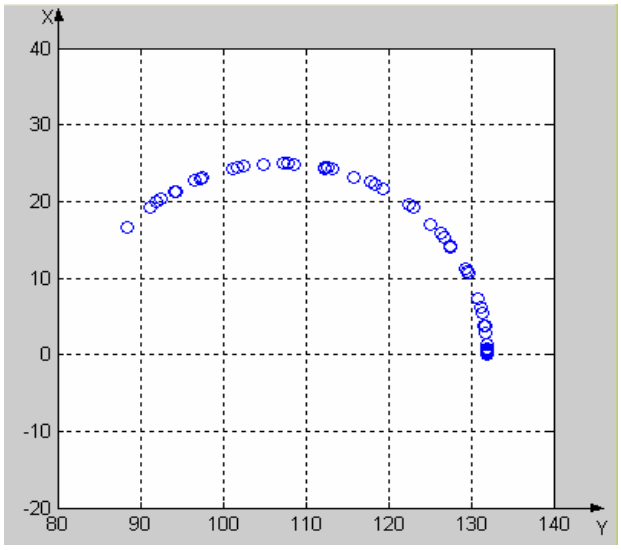


Figura 5 Representación de puntos en XY

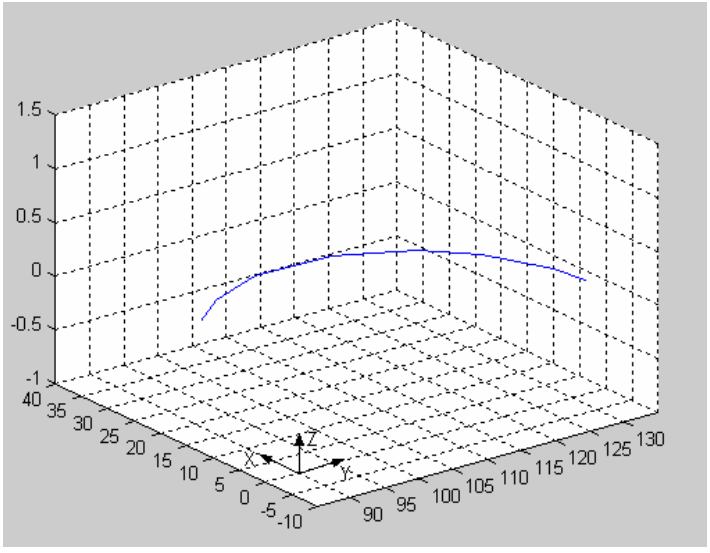


Figura 6 Simulación de la trayectoria de la pistola del robot

En este caso de estudio, la senda del robot es una circunferencia (ver fig.6). Este movimiento de la pistola acompañado con la rotación de la pieza permite alcanzar el proceso de recubrimiento de la pieza. La senda generada esta desarrollada en el código de la subrutina Kugelzapfen3. La velocidad del robot, la reposición de la herramienta y las posiciones de igniciones son incluidas en esta subrutina.

2º Objetivo del proyecto

El análisis numérico de la transferencia de calor de la llama y de las partículas calientes impactando en la superficie de la pieza durante el proceso de recubrimiento mediante HVOF (High Velocity Oxigen-Fuel).
El software "ABAQUS 6.7" es usado en este estudio para el modelado, análisis, control y optimización de estos fenómenos de transferencia de calor.

Tras la importación del modelo generado en CAD obtenemos el modelo de la pieza de trabajo en el software de elementos finitos (ver fig.7).
El modelo necesita que se le definan la transferencia de calor (cargas térmicas), las propiedades del material para el substrato y el recubrimiento, las condiciones iniciales y el mallado del modelo.

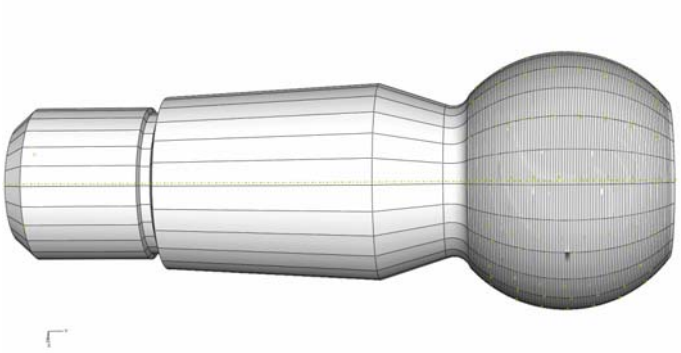


Figura 7 Modelo de la pieza en “ABAQUS 6.7”

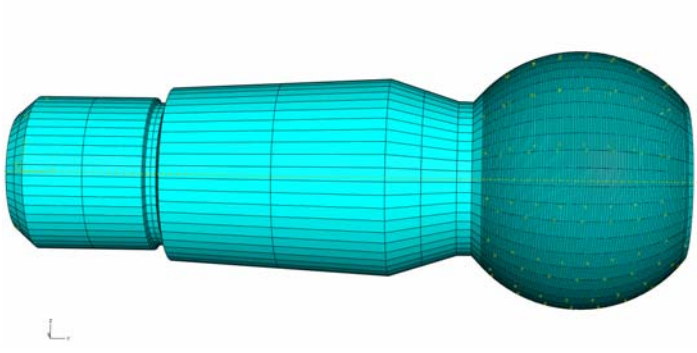


Figura 8 Mallado del modelo

La aplicación de las cargas es el proceso más laborioso debido al movimiento de rotación que experimenta la pieza, que acompañado con el movimiento de translación de la pistola del robot da como resultado un movimiento en espiral (ver fig.2).

En la tabla 5.8 se muestran los parámetros del movimiento utilizados para el cálculo de la duración de las cargas térmicas en los steps, resultando un ciclo de 0,4 segundos. Cada anillo se compone de 12 nodos, cada 30º existe un nodo (ver fig.9). La figura 10 muestra las cargas térmicas aplicadas en un anillo.

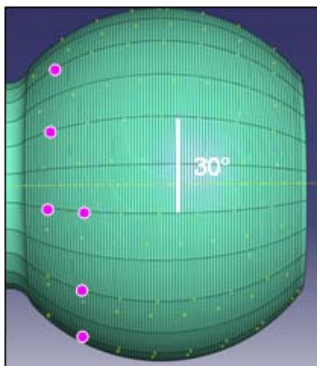


Figura 9 Nodos en el primer anillo

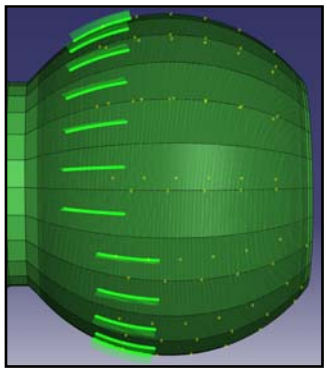


Figura 10 Anillo con cargas térmicas

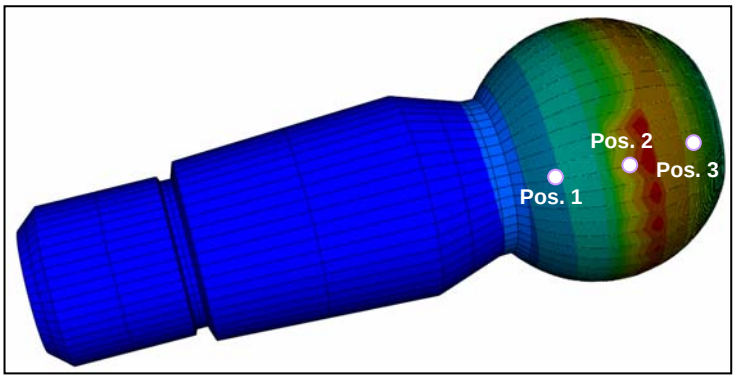


Figura 11 Posiciones de los nodos seleccionados

Los resultados de este análisis se pueden procesar con la evolución de la temperatura, los flujos de calor y la evolución de las tensiones.
La figura 13 muestra la evolución de la temperatura en tres nodos sobre la superficie de la pieza. Los tres nodos seleccionados se encuentran localizados en el principio, mitad y parte final de la superficie a recubrir (ver fig.11).

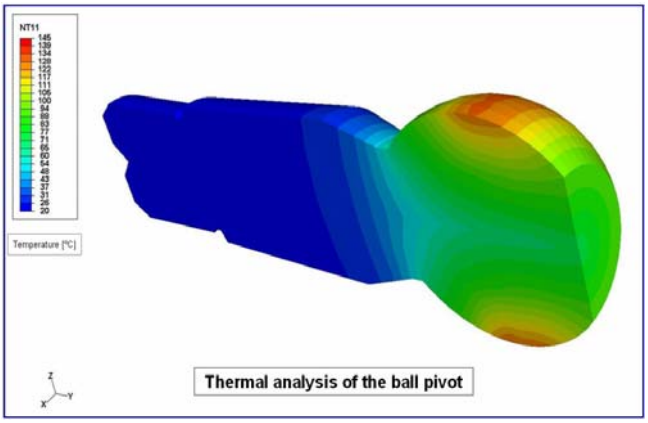


Figura 12 Campo de temperatura en una sección de la pieza

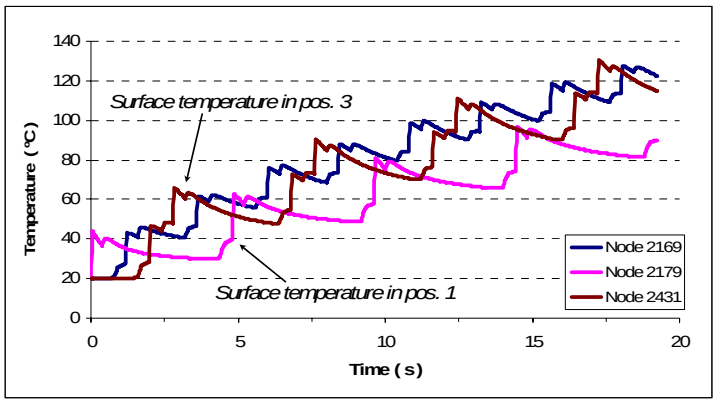


Figura 13 Evolución de la temperatura nodal durante la simulación del proceso de recubrimiento

El tiempo de simulación son 20 segundos, siendo representados cuatro ciclos. Durante un ciclo el chorro de la fuente de calor se mueve por todos los puntos de la superficie, dibujando una espiral como trayectoria. En la figura 13 es posible distinguir cada ciclo para los nodos primero y tercero. Para el segundo aparecen ocho picos debido a que el chorro pasa por este nodo dos veces por ciclo.

En cada ciclo se incrementa el valor de la temperatura, aunque el valor del incremento es decreciente con el nuevo ciclo. Por esta razón, no es posible prever la temperatura alcanzada durante la aplicación de varios ciclos sin la ayuda de la simulación numérica.
Al final de la aplicación de los cuatro ciclos la temperatura máxima alcanzada es de 145º C. El proceso de recubrimiento de la pieza tiene aproximadamente una duración de tres minutos y las medidas de temperatura realizadas con la cámara termográfica proporcionan un valor máximo de 220º C.

Conclusiones

Los sistemas robot son introducidos por modernas instalaciones para los procesos de spray térmico. La trayectoria del robot está llegando a ser un parámetro de proceso importante. Este proyecto desarrolla diferentes posibilidades para generar trayectorias y programas de robots industrial para aplicaciones de spray térmico, con el foco en la programación off-line soportada por CAD.
En orden de considerar la geometría real del componente herramientas CAD, ópticos o táctiles sistemas de medida de coordenadas pueden ser usados para predecir los datos requeridos por la programación off-line. Específicas herramientas de software fueron desarrolladas para procesar la geometría del substrato, la transformación de coordenadas y el cálculo de los vectores normales y sus trayectorias.

Además es necesario controlar la alta carga térmica sobre la superficie a recubrir. Se desarrollaron simulaciones numéricas para evaluar la distribución de temperatura y tensión durante el recubrimiento. Así la planificación de la trayectoria de la pistola del robot se acopló con la simulación en el método de elementos finitos.
Para verificar los códigos desarrollados, las subrutinas programadas se aplicaron al recubrimiento de la rótula usada en el tren de alta velocidad (AVE). La evolución de la temperatura y la tensión en el substrato durante el proceso de recubrimiento de 20 segundos se calculó y discutió.

Index

1 Motivation

In order to ensure high reproducibility in the production process as well as net-shape coating properties, advanced machine tools and robot systems are used for modern thermal spraying applications. Robot systems have an important presence in modern industry. They can develop repetitive tasks and can be used in atmospheres which are dangerous for the human health.

Due to high kinetic and thermal energy of the impinging gases and particles, the trajectory and velocity of the automatically guided torch has a crucial importance in the heat and mass transfer processes during the production of thermally sprayed coatings. Properties such as coating thickness, porosity, microhardness and thermal stress distribution are therefore significantly influenced by the spraying distance, velocity and the spatial trajectory of the torch. In principle the trajectory of the tool is the parameter more complex of calculate in the industrial robot.

The definition of the trajectory of the torch is a working step in the robot programming, which is normally a laborious task. In most cases, it is realized by measuring points in the manufacturing cell by teach-in, which results in a low continuous production.
Thus, a time efficient development of the robot trajectories for the production of net shaped coatings on complex geometries requires specific off-line programming tools.
Off-line programming is much comfortable, increase of work safety and efficiency, low time to program, continuous production and it consists of generate the robot's trajectory in a computer, it allows realize the task without interferences in the job of the industrial robot. According to the *International Federation of Robotics*, as of 2007 over 950,000 robots have been installed for industrial applications in the world and an additional 100,000 are being sold every year. Currently less than 1% of these robots are programmed using off-line graphical robot programming (CAD/CAM). In comparison over 70% of CNC machines are programmed using computer aided design and manufacturing software.

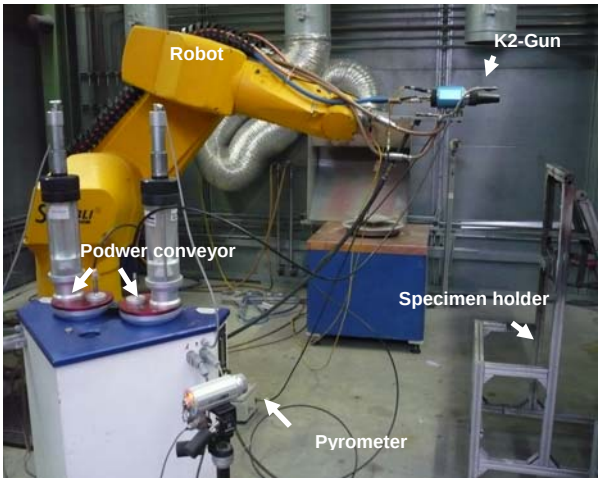


Figure 1.1 Robot RX 170 for the coating in IFKB

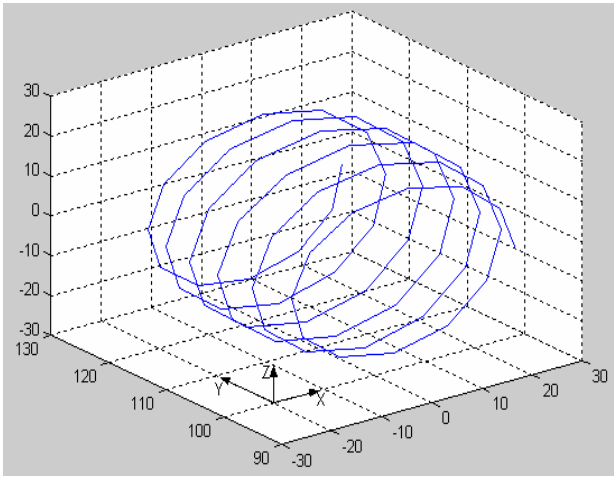


Figure 1.2 Simulation of the trajectory

By means of appropriate coupling of the automatically generated trajectories with close to reality FEM-models of workpiece and coating process, a bidirectional information flow between robot off-line programming and heat/mass transfer simulation can be achieved.

The work will focus on the optimization of this information flow so that the coating process with the generated trajectories could be simulated with the FEM programs ANSYS and ABAQUS. Resulting thermal distributions should contribute to the optimization of the robot trajectories.

For the acquisition of the geometrical data of the surface to be coated 3D CAD models as well as tactile and optical measurement devices are used. Algorithms and software tools for the generation of 3D trajectories as well as 3D FEM-Models of the considered workpieces should be developed. By integrating compatible interfaces, the geometrical data can be simulated, translated to the programming language of the robot controller and finally implemented in a real manufacturing cell.

In this study a software toolkit for the off-line generation, simulation and implementation of trajectories (position and orientation of the HVOF torch) is applied. In this way, a reduction in the programming time as well as high adaptability to perform modifications in the trajectory or related parameters (spraying distance, meander offset, special trajectories, speed profiles, etc.) can be obtained. An integral solution and the corresponding software architecture for the automation of the production of HVOF coatings on complex geometries are proposed.

1.1 Aims of the project

The objective of Diploma Thesis is to develop the different possibilities to program an industrial robot such as on-line or off-line programming, especially in the generation of the trajectories of the robot by means off-line programming strategies supported by CAD. The generated trajectories will be implemented in the cabin of the industrial robot.
Due to high thermal load generates over the surface of the piece in thermal spraying process, the process will be simulated in the software of finite element “ABAQUS”. This simulation shows the distribution of temperature and results stress reached.

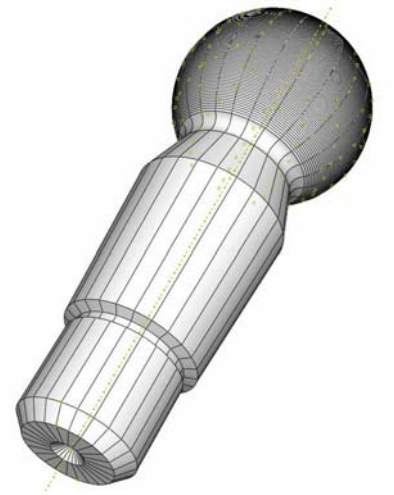


Figure 1.3 Model of piece in “ABAQUS 6.7”

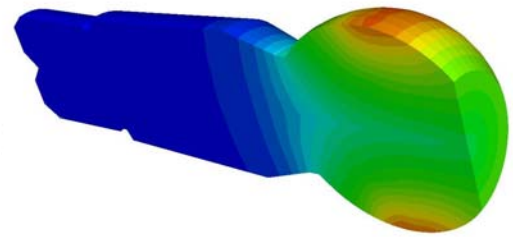


Figure 1.4 Field of temperature in a section

2 State of the art

2.1 Thermal spraying

The production process known as “thermal spraying” has emerged as an important tool in area of surface technologies for industrial applications. Thermal spraying is included in the group of processes which are used to apply metallic and non-metallic coatings. These technologies are mainly applied to protect the surface from the environment that may produce corrosion and wear of the substrate, thermal degradation and other surface phenomena. Usually they are also used to improve the appearance of the surface. The functionality of the layer composites is often characterized by the bonding between substrate and coating material, microhardness, porosity and residual stress distribution. Generally acceptable coatings are characterized by good adhesion, substrate compatibility and low porosity, although there are applications that can require high porosity to infiltrations of material.

Thermal spraying processes have been using since 1900’s, when Dr. Max Schoop introduced this technology using the combustion flame as heat source. Thermal spraying can be defined as a family of processes that uses a concentrated heat source to melt feedstock materials while imparting kinetic energy, using process jets to propel the molten particles toward a prepared surface (see fig. 2.1) [2].

It is normally necessary adequately to prepare the surface to be coated; different operations can be used such as degreasing, shaping, roughening (grit-blasting) or masking.

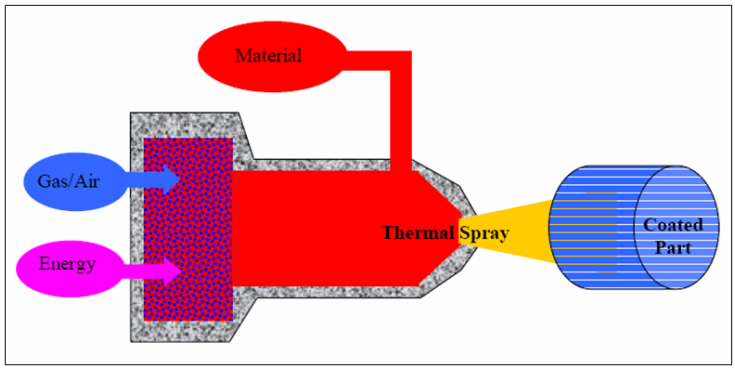


Figure 2.1 Schema of a thermal spray process [3]

It is possible to classify the different processes of thermal spraying according to the heat production processes:

- Combustion
  - Flame Spraying
    - Powder Flame Spraying
    - Wire Flame Spraying
  - High Velocity Oxigen-Fuel Spraying (HVOF)
    - O_2 + gas fuel (HVOF-G)
    - O_2 + liquid fuel (HVOF-K)

- Electrical discharge
  - Plasma Arc Spraying
    - Conventional Plasma or Atmospheric Plasma Spray (APS)
    - Vacuum Plasma Spraying (VPS) or Low-Pressure Plasma Spraying (LPPS)
  - Wire Arc Spraying
- Laser
- Cool spray process

The major advantage of thermal spraying processes is the wide variety of coating materials which can be used, such as metals, alloys, ceramics, polymers, composite materials (cermets such as WC/Co), graded materials and others [2].

Others benefits of thermal spraying processes can be:

- Relatively low costs of equipment and operation.
- Simple manage of installation.
- Relatively low environmental impact.
- Thick coatings (up to 500 μm) can be applied at high deposition rates.
- Components can be sprayed with a pre- or post-heat treatment, and component distortion is minimal.
- Possibility to recoat worn or damaged coatings without changing part properties or dimensions.

Figure 2.2 shows components for thermal spraying equipment:

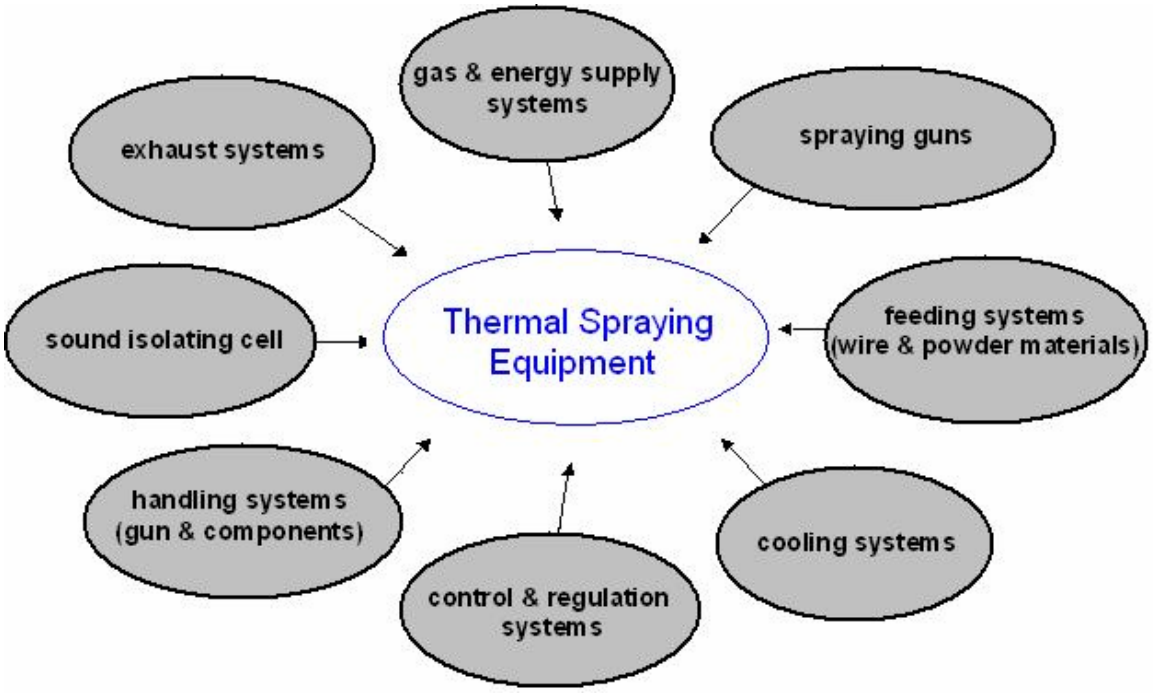


Figure 2.2 Components for thermal spraying

For industrial applications the thermal spray technologies offer the possibility to improve:

- Wear resistance.
- Heat resistance.
- Clearance and dimensional control.
- Corrosion and oxidation resistance.
- Electrical properties (resistance and conductivity)

2.1.1 Atmospheric Plasma Spraying (APS)

Atmospheric plasma spraying is one of the most flexible of the thermal spray processes regarding the materials that can be sprayed, because of the high gas velocity and extremely high temperatures.

In the atmospheric plasma spraying, the thermal energy of an electric arc and a plasma forming gas are used all together in the melting and projecting of the deposit material at high velocity onto a substrate [3].

The spraying gun is composed by two electrodes, a copper circular anode around a cathode of thoriated tungsten. By means of an electric arc discharge supported by a generator through the electrodes a mixture of gases is heated. The interaction of the electric arc with the gas mixture makes the gas atoms ionize and dissociate; as a result a plasma jet is formed.
The powder is injected in the plasma by a carrier gas, the powder particles are melted and accelerated in the plasma, the impact onto the substrate surface forms the coating (see fig. 2.3 and 2.4) [6].

Due to the high temperature achieved in the plasma, the spraying gun needs a cooling water system during the process.

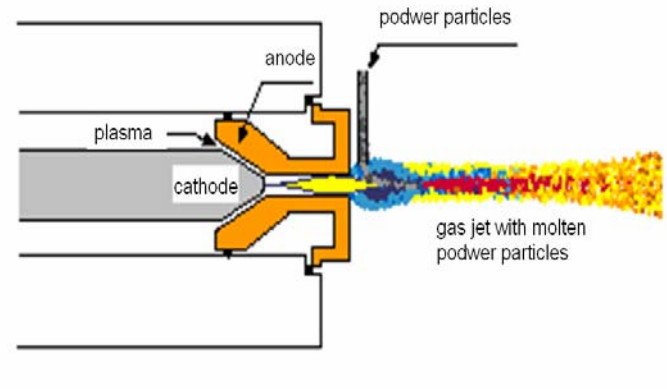


Figure 2.3 Schema of APS torch

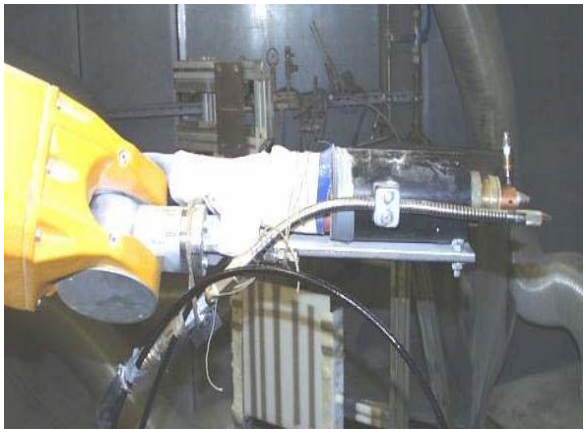


Figure 2.4 APS gun at IFKB

Table 2.1 shows the characteristics of APS process, such as plasma gas, maximum plasma temperature, gas velocity and others.

Plasma gas	Argon (Ar) Helium (He) Nitrogen (N_2) Hydrogen (H_2)
Gas rate (l/min)	40 – 50
Coating material	Metals, alloys, oxides, cermets, carbides, plastics and biocompatible materials in powder form
Material feed rate (g/min)	50 – 150
Electric power (kW)	30 – 80
Electrodes geometry	Cathode's geometry depends on the plasma gas used. Anode's shape and diameter are influenced by the temperature and jet velocity plasma
Substrate material	Metals, alloys, any steel, cast iron, Co and Ni alloys, light metals and its alloys, Cu and its alloys, glass, plastic materials and any ceramics
Spray gun temperature (ºC)	16000
Exit plasma temperature (ºC)	5500
Particle temperature (max,ºC)	> 3800
Gas velocity in nozzle (m/s)	800
Particle velocity (m/s)	120 – 600
Linear velocity of the torch (mm/s)	50 – 2000
Powder particle size (µm)	5 – 100
Spraying distance (mm)	60 – 130
Bond strength (MPa)	30 – 70
Porosity (%)	1 – 7
Thickness of the coatings (µm)	50 – 500

Table 2.1 Characteristics of APS process [2, 3, 6, 8]

Plasma sprayed coatings are generally much denser than other thermal spray processes with the exception of HVOF and detonation processes. The high quality of plasma spray coating microstructure results from the high particle kinetic energy and high degree of melting, which contribute to the deformation of the impacting particles and splat formation. The high droplet/substrate adhesion is achieved because of the high particle velocity (see table 2.1) and deformation that occur on impact. The inert gas gets a lower oxide content than other thermal spray processes. Disadvantages of the plasma spray processes are relative high cost and complexity of process.

Atmospheric plasma spray systems consist of similar components to other thermal spray systems [2]. The main components are:

- Gas supply.
- Gas hoses.
- Gas regulators for plasma and powder carrier gases.
- Gas flow controls, such as rotameters or mass flowmeters/controllers.
- Spray gun comprising a torch body, anode, cathode, gas and powder injector.
- Electrical controls for arc current and voltage.
- Direct-current power supply.
- Water-cooling circuit.
- Water-cooled power cables.
- Feedstock delivery.
- Hoses/cables
- Safety interlocks and console purging.

The APS process has a lot of applications in the modern industries, such as:

- Automotive industry
    - Alternator
    - Levies
    - Adiabatic diesel motors
- Energy industry
    - Gas turbines
    - Burners of powdered coal
    - Generates MHD (Magneto-Hidrodinamics)

For high performance applications, in order to avoid oxidation, approach theoretical bulk density and extremely high adhesion strength, plasma spray is carried out in a reduced pressure inert gas chamber and this process is called vacuum plasma spray (VPS).

2.1.2 High Velocity Oxigen-Fuel (HVOF)

The HVOF spray process was invented by Union Carbide in 1958, but did not become commercial until the early 1980s when the Jet-Kote system was introduced by James Browning [2].

The HVOF thermal spray process is another form of the flame spray process, but using powder as the coating material rather than wire or rod.

In this process the fuel is burnt with oxygen at high pressure to generate a high velocity exhaust exit. The powder is injected axially into the jet as suspension in the carrier gas. The gases are burnt in the combustion chamber and flow through the nozzle out of the torch [6].

There are two categories of this process, the Detonation Gun HVOF system and the Continuous Combustion HVOF system. The different of these systems resides in the use of different fuel gas, cooling systems and in the detonation gun system combustion is maintained by a timed spark, used to detonate the particle and gas mixture.

- Detonation-Gun spraying

As examples of this gun, the TJ-4000 (TAFA Incorporated), HFPD (High Frequency Pulse Detonation, AEROSTAR Coatings) can be mentioned. The characteristic of this gun is its long barrel on the front of the gun.
The process consists of igniting the gas mixture by a spark plug, a controlled detonation wave heats, which heats the particles to temperatures of 4500 ºC and accelerates these at subsonic speeds of 800 m/s.
The uniform, closely packed and laminar structure of a detonation coating is due to the high temperature and kinetic energy of the particles, resulting higher density, hardness and bond strength than other thermal spray processes [3].

- Continuous Combustion HVOF system

Several torches can be mentioned: the Diamond Jet gun (DJ HVOF gun) and DJ Hybrid (both were developed by Sulzer Metco Coatings Company), the JP-5000 (PRAXAIR Surface Technologies and TAFA Incorporated), HV-2000 (PRAXAIR Surface

Technologies), HP/HVOF (High Pressure HVOF, TAFA Incorporated) and Top Gun G torch (GTV GmbH). The continuous combustion Jet-Kote HVOF thermal spray system was developed as an alternative to the Detonation Gun system.

The HVOF thermal spraying is a flame deposition technique, the powder material is melted by the combustion of fuel gas and oxygen, and it is propelled at a high velocity by compressed air towards a surface.
The powder material becomes molten or semi-molten depending on the melting temperature and feed rate of the material. The flame temperature for this process is between 2300-3000 ºC. The particles are propelled of the gun nozzle at supersonic velocities of over 1350 m/s towards the substrate surface.

The HVOF system exhibits the highest bond strengths and lowest porosity of all the other thermal spraying processes, due to the high kinetic energy experienced by the impinging particles [3].

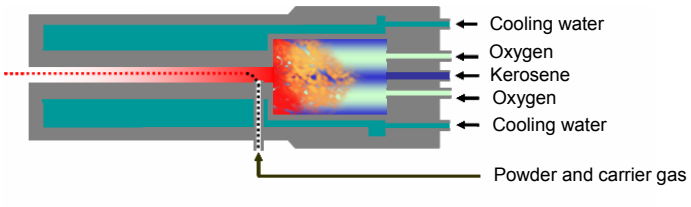


Figure 2.5 Schema of a HVOF torch (fuel gas)

Figure 2.6 HVOF gun at IFKB

The design of the HVOF gun is based on the requirement to burn oxygen and a fuel gas in an enclosed chamber, building up a high enough pressure to accelerate particles through the nozzle. There are two designs of burners such as chamber burner and throat combustion burner.

The HVOF gun presents basically three entrances (oxygen, coating material and fuel) and one exit. These products go through a convergent-divergent conduct with a bigger diameter at the exit to disperse the gas jet. These guns also need a cooling system (air or water) to support the high chamber combustion temperature. The cooling flow generally has its entrance near the nozzle and exits at the gun base (see fig. 2.5).

Table 2.2 shows the characteristics of HVOF process, such as type of fuel, coating material, gas velocity, chamber combustion temperature and others.

Type of fuel	Kerosene Propylene Propane Ethane Acetylene Hydrogen Methylacetylene-propadiene
Fuel rate (l/min)	40 – 60
Coating material	Cermets (carbides like WC/Co, Cr_3C_2Ni with a metallic binder based on Ni, Cr or Co), pure metallic and low melting ceramic (TiO_2).
Material feed rate (g/min)	15 – 50
Electric power (kW)	100 – 270
Spray gun temperature (ºC)	4500
Flame temperature (ºC)	3100
Particle temperature (max,ºC)	3300
Jet velocity (m/s)	500 – 1200
Particle impact velocity (m/s)	610 – 1060
Powder particle size (µm)	5 – 45
Spraying distance (mm)	150 – 300
Bond strength (Mpa)	> 70
Porosity (%)	0.5 – 2
Deposition rate (kg/h)	1.0 – 9.0
Thickness of the coating (µm)	100 – 300

Table 2.2 Characteristics of HVOF process [2, 3, 6, 8]

The HVOF process is designed around producing high velocities, because of the high kinetic energy of the particles, improved bond strengths as well as low porosity are obtained. The particles impact onto substrate in a semi-molten state. The advantages of HVOF process over other thermal spray process are:

- More uniform and efficient particle heating with fewer un-melted particle content, due to the high turbulence experienced by the particles within the combustion chamber.
- Much shorter exposure time in flight due to the high particle velocities.
- Lower surface oxidation due to short particle exposure time in flight to air.
- Higher density, lower porosity and smoother as-sprayed surface due to the higher impact energy.
- Lower ultimate particle temperature (3000 ºC) compared to other processes such as plasma (16000 ºC) or arc guns (6000 ºC).
- Thicker coatings, improved wear resistance and corrosion barrier, higher hardness ratings, higher bond and cohesive strengths.

The mechanical properties of the coating produced by HVOF depend on the microstructure of the deposits, which are determined by the physical and chemical state of the particles at the impact with the substrate: speed, temperature, melted grade and content in oxides. Finally these variables depend on the process parameters that include: the relationship fuel/oxygen, gas flow, spraying distance and distribution of the particles among others.

The flame temperatures reached with this process are lower than with plasma spraying process, the powder materials which can be used are limited. It is generally not possible to fully melt refractory ceramics particle by using this technique. It is important to take

into account that HVOF sprayed WC/Co coatings have hardness and wear resistance values superior to such coatings sprayed by plasma spraying process.

HVOF process equipment is similar to the one used for conventional flame spray [2], the main components are:

- Spray gun comprising a torch body, combustion chamber, powder injector and nozzle.
- Water- or air-cooling circuits.
- Fuel gas, oxygen and powder carrier gas circuits consisting of a high-volume gas supply, high-pressure gas hoses, high flow gas regulators for oxygen and fuel, high flow gas controls, flashback arrestors at the gun and regulators.
- Feedstock delivery: high-pressure powder feeder.
- Safety interlocks and console purging.

The applications of HVOF process to industry can be:

- Vapor turbines, the shovels are subjected to the erosion by condensation of water drops, producing a quick wear. The surface is protected with Stellite 6 (Co, Cr, W and C).

2.2 Industrial robots for thermal spraying

The second focus of this work is to generate the robot path, like a tool of work process. The structural element to thermal spraying processes is the industrial robot. Due to properties such as high precision, highly repetitive, without fatigue in addition, industrial robots can develop tasks in dangerous environments, so this is the adequate element to develop the thermal spraying processes. Industrial robots are also used to applications of spray coating, spray painting, assembly, welding, transport or manipulations, inspections, cleaning, etc.

An official definition of industrial robot is given by the norm ISO 8373-94, which defines an industrial robot like “an industrial robot is an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes”.

Robots present different systems of classification such as kinematic structure, degrees of freedom, drive technology and workspace geometry, they are also composed by a lot of different components such as sensor, controller, links and joints, actuators and motors, transmission mechanisms and a wide variety of end-effectors.

An important part in industrial robots is the definition of coordinate systems, such as representation of position and orientation in space. All this parts will be developed in the following chapter.

In the presented work, the aim consists of obtaining the path robot by different methods. The data acquisition defines the used method, which can be based on the on-line programming (teach-in, play-back, master-slave, programming supported with sensors) or off-line programming (reverse engineering, CAD model).

Figure 2.7 shows the robot RX 170 CS7, which is used for the torch handling at the IFKB (University of Stuttgart).

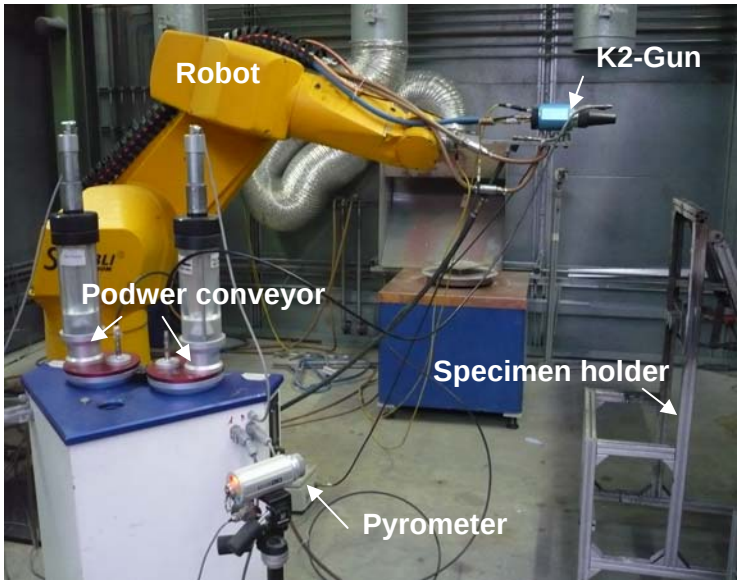


Figure 2.7 Robot RX 170 (Co. Stäubli) for the coating in IFKB

2.3 Simulation of heat transfer in thermal spraying applications

The third focus of this work is the simulation, which consists of the simulation of thermal spraying process for a piece, with the purpose of studying the temperatures distribution at the substrate's surface and the stresses to which the piece will be subjected during the coating process.

Simulation softwares are nowadays widely used to simulate complex processes, where the main purpose is to model the production process and foresee part properties. For example in the area of thermal spraying, ABAQUS (Hibbit, Karlsson & Sorensen, Inc), ANSYS (Swanson Analysis Systems, Inc, now ANSYS, Inc), FLUENT (Fluent, Inc, Joins ANSYS, Inc), among others software are used.
There are a lot of simulation softwares for industrial use, which can develop the processes of identification, execution and optimization of tasks in an assembly chain, design of models for CNC machine (TekSoft CAD_CAM), simulation of the manufacturing cell of the robot and of the trajectory programmed for the task (RobCad (UGS Corporation), Delmia IGRIP® (DELMIA Corporation), RobotStudio developed by Asea Brown Bowrie Ltd (ABB), Robotmaster (Jabez Technologies Inc)).

Simulation work developed in the last year in field of thermal spraying include the study of the splat formation, combustion process for HVOF or heat transfer and stress analysis of a propeller for HVOF.
In this case the main aim is focused on simulating the distribution of temperatures on the surface of the substrate. For that it is necessary to know the thermophysical laws that govern the heat transfer process.

- *Heat transfer equation*

The basic energy balance is (Green and Naghdi):

$$\int_V \rho \cdot \dot{U} \cdot dV = \int_S q_S \cdot dS + \int_V q_V \cdot dV \qquad [2.1]$$

Where V is a volume of solid material, with surface area S, ρ is the density of the material, $\dot{U}$ is the material time rate of the internal energy, q_S is the heat flux per unit area of the body and q_V is the heat supplied externally into the body per unit volume [5].

The equation of heat flow balance inside a body in Cartesian coordinates is:

$$-\frac{\partial}{\partial x}\left(k_x \cdot \frac{\partial T}{\partial x}\right)-\frac{\partial}{\partial y}\left(k_y \cdot \frac{\partial T}{\partial y}\right)-\frac{\partial}{\partial z}\left(k_z \cdot \frac{\partial T}{\partial z}\right)+q_V=\rho \cdot C \cdot \frac{\partial T}{\partial t} \qquad [2.2]$$

Where k is the thermal conductivity, T is the temperature and C is the specific heat of material.

Except in the phase changes, the internal energy can be related with the specific heat, by means of the following equation:

$$C(T)=\frac{dU}{dT} \qquad [2.3]$$

The phase change involves the appearance of a latent heat, added to the effect of the specific heat (see fig, 3.9 and 3.10). In many cases it is reasonable to suppose that the phase change occurs within a known temperature range and the process is modelled with just a generation of heat for unit of mass. However, in some cases, it may be necessary to include a kinetic theory for the phase change to model the effect accurately [5].

The equation of flow balance is:

$$\rho \cdot \frac{\partial U}{\partial t}+\frac{\partial}{\partial x}\left(k_x \cdot \frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k_y \cdot \frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(k_z \cdot \frac{\partial T}{\partial z}\right)-q_V=0 \qquad [2.4]$$

The conduction, convection and radiation processes take place at the surface of the solid and they can be defined by different laws.

- Heat conduction process

This process is assumed to be governed by the Fourier law:

$$k \cdot \frac{\partial T}{\partial x}-q_S=0 \qquad [2.5]$$

Where q_S is the heat flow per unit area, k is the conductivity matrix, k=k(T) and x is position.

- Heat convection process

$$q_{SC}=h\,(T-T^0) \qquad [2.6]$$

Where $h = h(x,t)$ is the film coefficient, $T^0 = T^0(x,t)$ is the sink temperature and $q_{SC} = q_{SC}(x,t)$ is the surface heat flux per area.

- Heat radiation process

$$q_{SR} = A[(T - T^Z)^4 - (T^0 - T^Z)^4] \qquad [2.7]$$

where A is the radiation constant (emissivity times the Stefan-Boltzmann constant) and T^Z is the absolute zero on the temperature scale used [5].

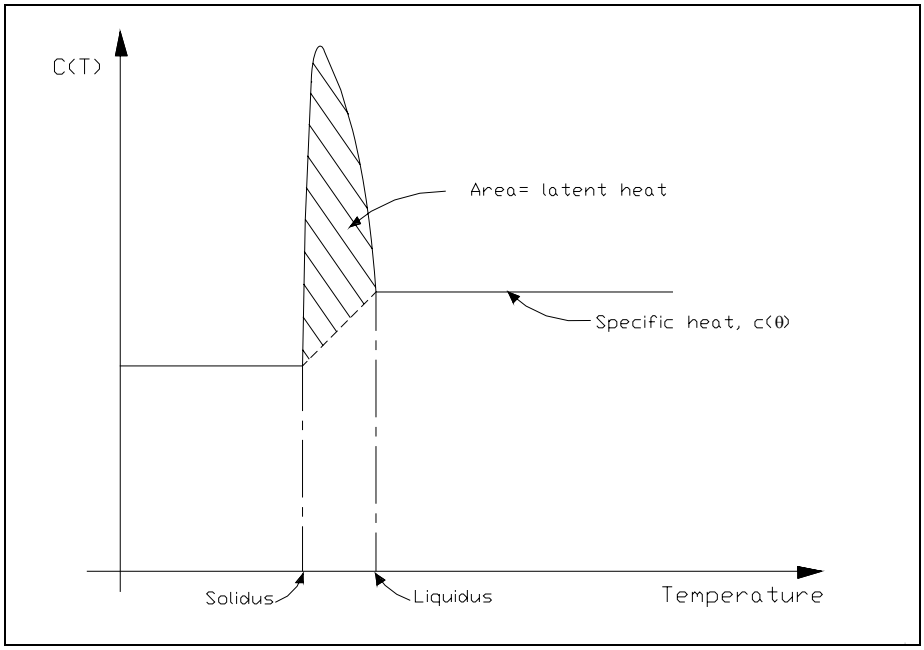


Figure 2.8 Relationship between specific heat and temperature [5]

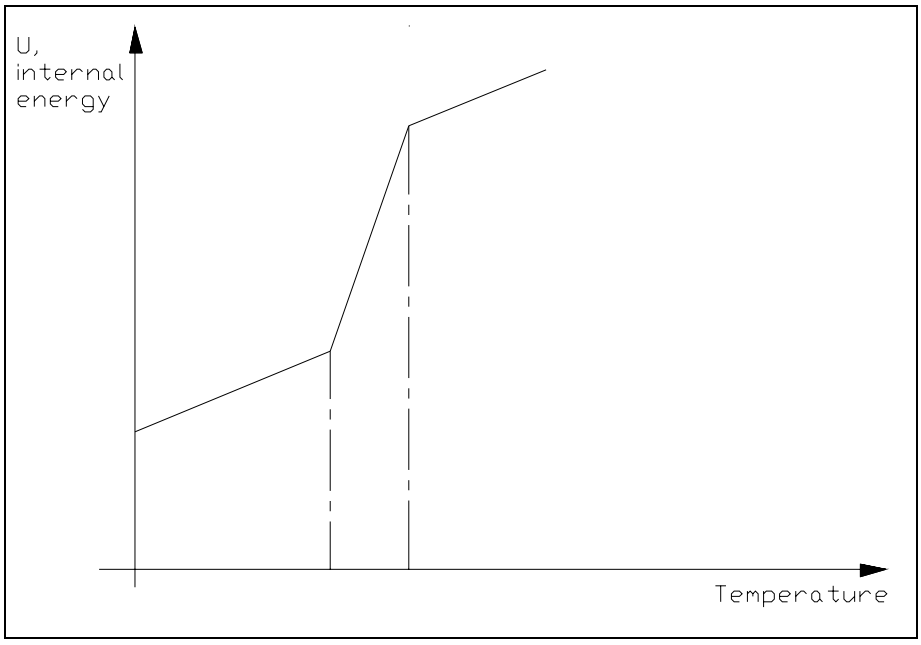


Figure 2.9 Relationship between internal energy and temperature [5]

In a more compact version:

$$\int_V \rho \cdot \dot{U} \cdot dV \cdot \delta T + \int_V \frac{\partial(\delta T)}{\partial r} \left[k_x \cdot \frac{\partial T}{\partial x} \hat{i} + k_y \cdot \frac{\partial T}{\partial y} \hat{j} + k_z \cdot \frac{\partial T}{\partial z} \hat{k} \right] dV - \int_V \delta T \cdot q_V \cdot dV = \int_S \delta T \cdot q_S \cdot dS \qquad [2.8]$$

To simplify the thermal and mechanic analysis of the piece is assumed that the thermal and mechanics problem are uncoupled. Therefore:

- U is only function of the temperature, T
- q_S and q_V do not depend on the efforts either on the displacements of the body

Because the heat contribution to the piece during the thermal spraying process some stress is generated, the equations that dominate the stress calculates due to thermal loads are developed.

- *Stress calculation*

Starting from the field of temperatures, the coefficient of thermal expansion,α, and a reference temperature,T_0, we can obtain the unitary deformation according to the following equation:

$$\varepsilon = \alpha(T, f_\beta) \cdot (T - T_0) - \alpha(T_I, f_{\beta,I}) \cdot (T_I - T_0) \qquad [2.9]$$

Where T is the actual temperature, T_I is the initial temperature, T_0 is the reference temperature, f_β are actual values of predefined field variables and $f_{\beta,I}$ are initial values of predefined field variables.

Figure 2.10 shows the relationship between the unitary deformation and temperature with α function of temperature.

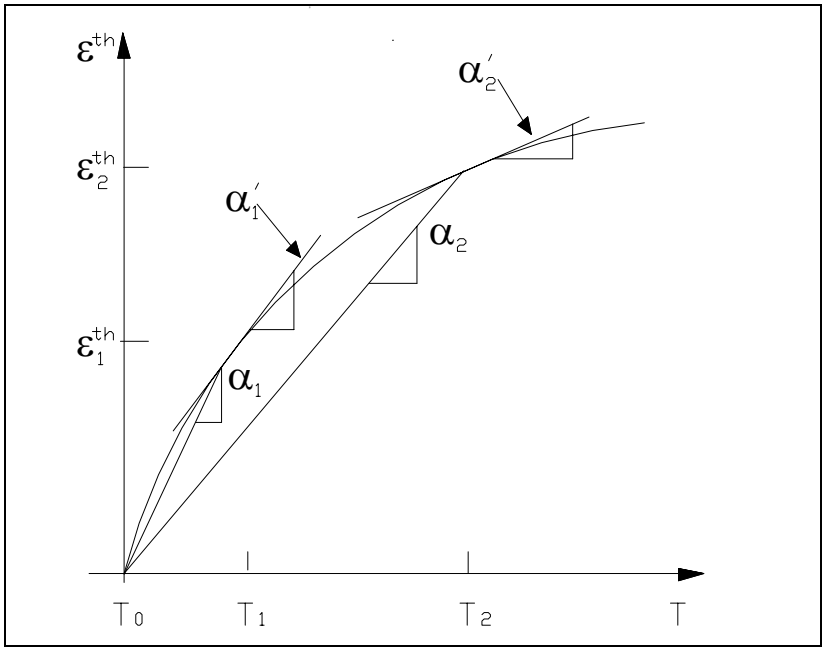


Figure 2.10 Relationship between unitary deform and temperature [5]

The second term of the equation 2.9 represents the deformation due to the difference among the reference temperature and initial conditions. If we presume that there is not deformation of thermal origin for the initial conditions, the second adding will disappear.

Using the laws that define the mechanical behaviour of the material, it is possible to translate these deformations in tensions. Exactly, in the case of elastic deformations, that is the simplest, the relationship between tensions and deformations is given by

$$\sigma = \varepsilon^{th} \cdot E \qquad [2.10]$$

3 Industrial robots

3.1 Introduction

In every range of industrial applications, cost reduction and quality increasing are essential objectives which lead nowadays to automation and computer controlled machining. Usual tasks for robot systems are “pick and place”, material loading and tool change operations. Due to the high precision levels required, complex component design and necessity of series production, robot technology has also become a mandatory tool in modern surface and coating technology. From the point of view of the robot programming, the development of accurate continuous path for the coating of complex geometries is a high demanding task. The aim of the present work is to develop methods of robot programming, based on the geometrical description of the component to be coated, for the optimization of surface coating processes. Reverse engineering and CAD-models are used for the acquisition of the geometrical data. The calculated robot trajectory is used for the numerical simulation of the temperature distribution during the coating process. In the following an introduction to industrial robot technology, basic components and commonly used coordinate systems is presented.

3.2 Definitions of industrial robots

To specify what kind of robot will be considered in the present work, official definitions of robot and industrial robot respectively are exposed:

“An automatic device that performs functions normally ascribed to humans or a machine in the form of a human” [13].

“An industrial robot is an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes” [14].

An industrial robot will be used in highly repetitive tasks and in case of dangerous or toxic environment for the human worker. In addition, modern robots can perform complex tasks, resulting in increased production over extended periods.

Typical applications of industrial robot include:

- Spray Coating
- Spray Painting
- Cutting Operations
- Assembling Operations
- Welding
- Packaging / Palletizing
- Material Removal
- Machine Loading
- Part Inspection / Testing
- Part Sorting
- Part Polishing
- Part Cleaning

These activities can be classified in following categories:

- Manipulation or transport.
- Processing: parts are altered by tools such as machining, welding, painting, soldering, coating…
- Inspection: parts are transferred from one place to another for examination, the robot awaits decision from inspector or machine for instructions for its next action.

3.3 Classification of robots

Robots can be classified according to various criteria, such as their degrees of freedom, kinematic structure, drive technology or workspace geometry.

3.3.1 Classification by degrees of freedom

A manipulator should possess six degrees of freedom in order to manipulate an object freely in three-dimensional space. We call a robot a *general-purpose robot* if it possesses six degrees of freedom, a *redundant robot* if it possesses more than six degrees of freedom, and a *deficient robot* if it possesses less than six degrees of freedom [16].
For special applications, such as assembling components on a plane, a robot with just four degrees of freedom would be sufficient. On the other hand, a redundant robot provides more freedom to move around obstacles and operate in a tightly confined workspace.

3.3.2 Classification by kinematic structure

Another scheme is to classify robots according to their structural topologies. A robot is said to be a *serial robot* if its kinematic structure take the form of an open-loop chain, a *parallel manipulator* if it is made up of a closed-loop chain, and a *hybrid manipulator* if it consist of both open-loop chains and closed-loop chains.

3.3.3 Classification by drive technology

Manipulators can also be classified by their drive technology. The drive technologies are *electric*, *hydraulic* and *pneumatic*. Most manipulators use either electric DC servomotors, because they are clean and relative easy to control. Hydraulic or pneumatic drives are usually used when high-speed or high-load-carrying capabilities are need. The disadvantage associated with the use of hydraulic drives is the possibility of leaking oils. Additionally, a hydraulic drive is flexible, due to the bulk modulus of oil. Although a pneumatic drive is clean and fast, it is difficult to control because air is a compressible fluid [16].

It has to be considered, that depending on the specific application, the wrist and gripper may use drive systems which can be different from the rest of the drives in the robot arm.

3.3.4 Classification by workspace geometry

The workspace of a robot is defined as the volume of space the end-effector can reach. This is directly related with the design of the mechanical system. Generally, there are five configurations:

- Cartesian coordinates
- Cylindrical coordinates
- Spherical coordinates
- Jointed Arm
- SCARA

3.3.4.1 Cartesian coordinates

To completely define the position of a point in 3D space, we need to specify particular values that fix the three degrees of freedom of position. By using Cartesian coordinates, the position of a point *a* with respect to the coordinate system OXYZ (see fig. 3.1), is defined by three values x, y, z or the vector P(x,y,z).
An example of robot arm controlled by using Cartesian coordinates is shown in figure 3.1b. The regional workspace of a Cartesian robot is a rectangular box.

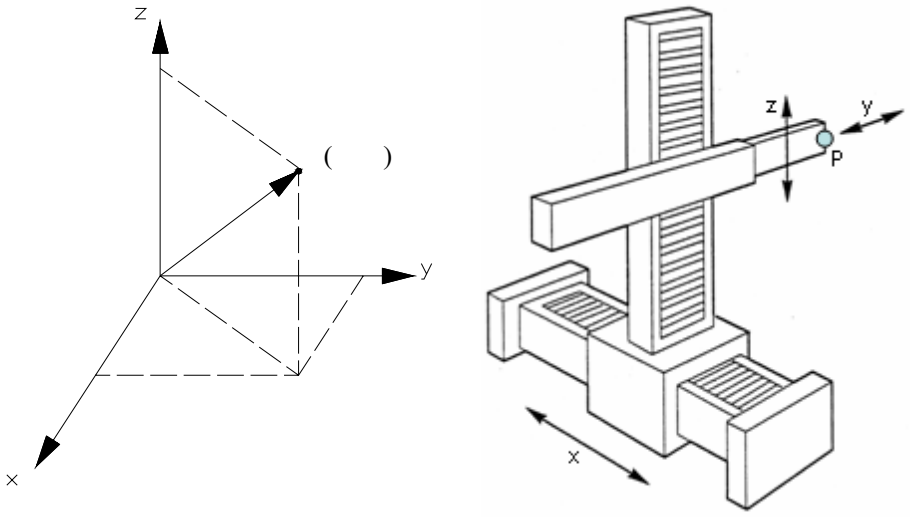


Figure 3.1 Definition of a position in space by Cartesian coordinates (left). Representation on a robot arm (right) [23]

3.3.4.2 Cylindrical coordinates

A position in space can be defined by cylindrical coordinates using the following values: magnitude of the projection of vector P onto the plane OXY, angle between the axis OX and the mentioned projection of P and the magnitude of the projection of vector P onto axis OZ (see fig. 3.2). The workspace of a cylindrical robot is confined by two concentric cylinders of finite length.

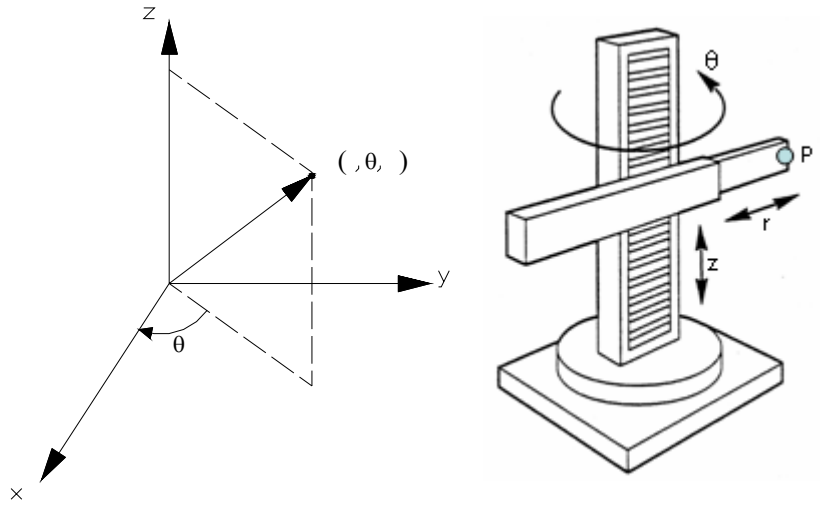


Figure 3.2 Definition of a position in space by cylindrical coordinates (left). Representation on a robot arm (right) [23]

3.3.4.3 Spherical coordinates

As shown in figure 3.3, the values for the spatial description of the position with spherical coordinates are the angle θ, the rotational axis, r, the reach axis, and ϕ, the bend-up-and-down axis. Programming a polar (spherical) coordinates robot involves translating x, y and z value for a point desired into two angles of rotation and an in/out value. Hence the workspace of a spherical robot is confined by two concentric spheres.

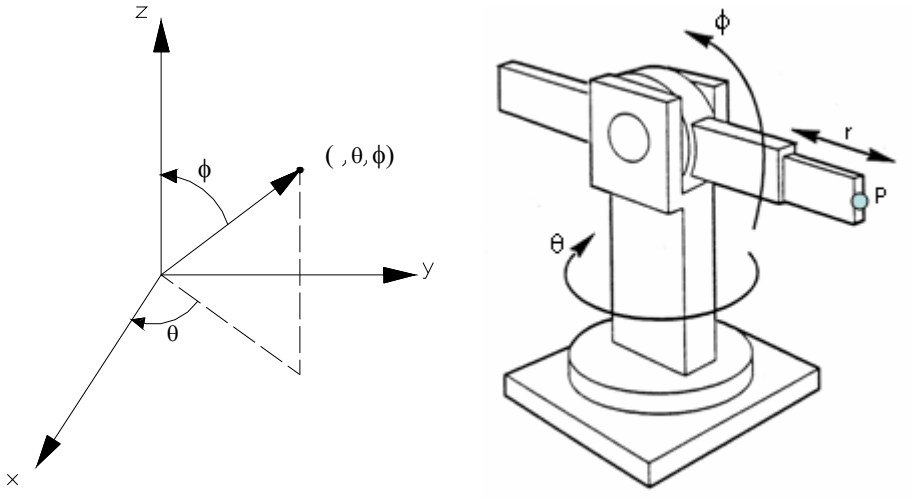
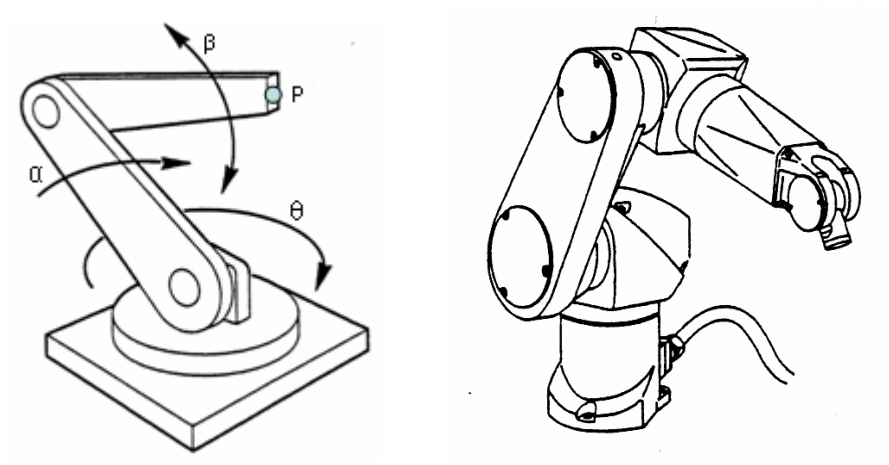


Figure 3.3 Definition of a position in space by spherical coordinates (left). Representation on a robot arm (right) [23]

3.3.4.4 Jointed Arm

The position in space can be also defined by the angles of the individual joints. As example, the robot arm shown in figure 3.4 can rotate about all three represented axes.

The axes for the revolute coordinates are θ, the base rotational axis, β, the upper arm rotational axis, and α, the lower arm rotational axis (see fig. 3.4).



(a) (b)

Figure 3.4 Jointed Arm [23]

3.3.4.5 SCARA

The SCARA (Selective Compliance Assembly Robot Arm) robot has a special configuration of the joints (see fig. 3.5). A SCARA robot presents the same work area as a cylindrical coordinates. However, the reach axis includes a rotational joint in a plane parallel to the floor. The SCARA arm is also more compact than the cylindrical coordinates arm.

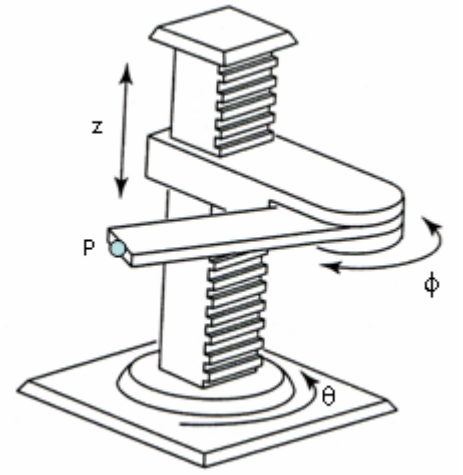


Figure 3.5 SCARA [23]

3.4 Basic components of industrial robots

Following main subsystems can be found in an industrial robot: the manipulator or mechanical structure, the actuators or motors drive, the end-effector and the controller. Figure 3.6 illustrates clearly the relationship of these four components at a typical industrial robot installation.

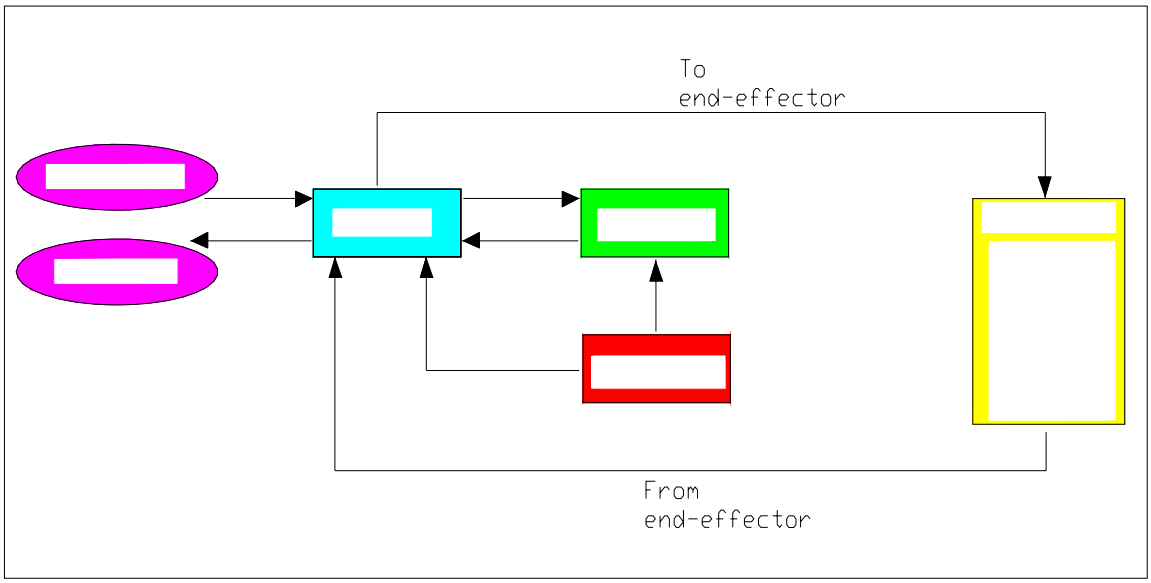


Figure 3.6 Basic components of an industrial robot [15]

3.4.1 Mechanical structure or manipulator linkage

The mechanical structure of an industrial robot consists of links and joints that allow the relative movement between two serial links, with axes capable of motion in various directions allowing the robot to perform work (see fig. 3.7).

The term lower pair is used to describe the connection between a pair of bodies when the relative motion is characterized by two surfaces sliding over one another.

The links of a robot arm should:
- have low mass;
- be very stiff;

these two requirements are not easily satisfied together. If an element is heavy and not stiff enough, it will deflect under its own weight. If the robot is also holding something in its gripper, for example, this deflection will be even larger.
To increase the stiffness of the links, they are usually made bigger, but this also makes them heavier. In this case, more powerful motors (hence, heavier) are needed to drive them. If the motors are placed at the joints, this extra weight also tends to deflect the link. This deflection results in positions errors at the end of the robot arm. It is therefore very important to keep these deflections as small as possible. The development of composites materials is allowing the substitution of steel in the links of the robots, carbon fibers are nowadays used for special applications.

The joints are typically rotary or sliding (see fig. 3.7). The last links is called the end-effector because it is this link to which a gripper or a tool is attached.

The manipulator can generally be divided into a regional structure and an orientation structure. The regional structure consists of the joints whose main function is to provide the specific motions that will enable the tooling at the end to do the required work. These are generally the proximal joints. The remaining distal joints are mainly responsible for the orientation the end-effector.

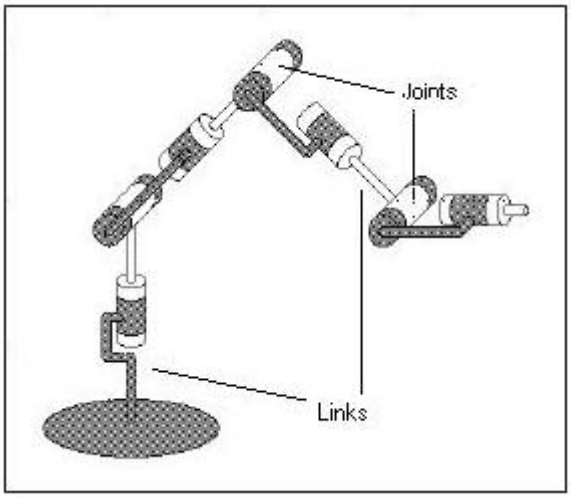


Figure 3.7 Mechanical structure of an industrial robot with 6 degrees of freedom

3.4.2 Actuators and motors drive

There are three primary sources of actuation power drive for industrial robots; those are *electric*, *hydraulic* and *pneumatic*. Some robot systems can require a combination of the three sources. The majority of the industrial robots are powered by electric drives, hydraulic drives and robots powered by pneumatic drives are relatively rare.

The function of the actuator power drive system is to provide and regulate the energy that is required for a robot to operate. In case of electrically driven robots, the power supply functions basically to regulate the incoming electrical energy and to provide the A.C or D.C voltages required by the electronic circuits internal to the robot controller and also by the drive motors. Hydraulically actuated robots include a hydraulic power supply either as an integral part of the manipulator or as a separate unit. The hydraulic system generally consists of an electric motor-driven pump, filter, reservoir and a heat exchanger. Pneumatically actuated robots are usually supplied by a remote compressor.

- Electrically actuated robots are almost all driven either by stepping motors or DC motors. They have less powerful than other types but have excellent repeatability and accuracy. This type is the most used as power drive actuator.

- Hydraulically driven robots use servo valves, analog resolvers for control and feedback, digital encoders and modern resolvers can provide a very high repeatability and accuracy.

- Pneumatically driven robots are used for limited activities, such as pick and place, where speed and precision are not critical. They use compressed air to drive the mechanical arm and tend to be lightweight, have limited capability.

3.4.3 Controller

The robot is connected to a computer controller which regulates the response of the motors and control motion parameters like velocity and acceleration. It processes the sensory information and computes the control commands that must be sent to the actuators to carry out the specific task.
It generally includes:

- A central processing unit (CPU) is the brains of the robot.
- Memory to store the control program and the state of the robot system obtained from the sensors.
- The appropriate hardware to interface with the external world (sensors and actuators).
- The hardware for a user interface.

The block diagram in figure 3.8 illustrates the many different parts of a robot controller.

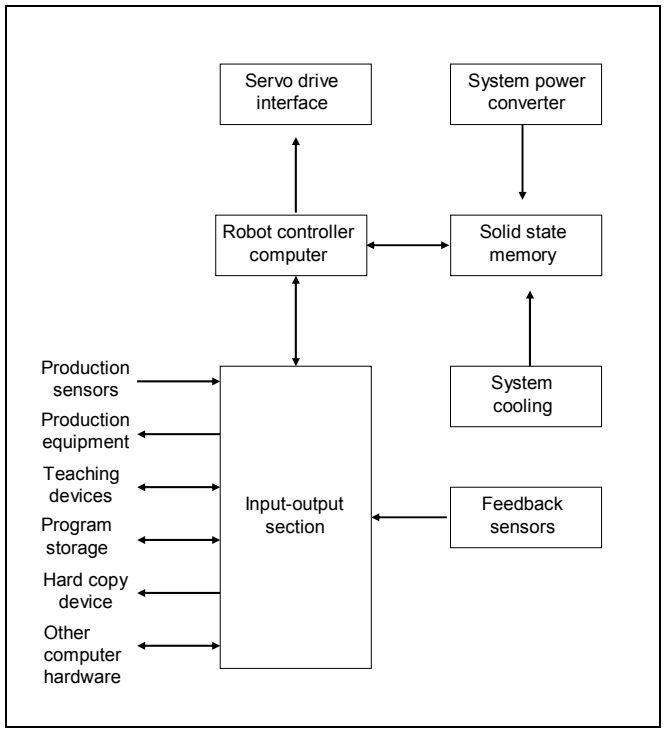


Figure 3.8 Robot controller block diagram [15]

3.4.4 Transmission mechanisms

The output of the reduction mechanism also have to be connected to the axis of the joint to be moved, so that the torque (or force) of the motor and its reduction mechanisms are transmitted to the joint. Transmissions are elements in charge of transmitting the movement from the actuators until the joints of the mechanical linkage. They are included together with the transmissions to the reducers, they are in charge of adapting the couple and the speed from the actuator to the appropriate values for the movement of the elements the robot. Since the robots move their end-effector with high accelerations, it is very important to reduce to the maximum their moment of inertia.

3.4.5 Internal sensors

Adding sensor to industrial robots could greatly expand their versatility while decreasing the mechanical tolerances.
Each joint that is motorised and controlled usually needs some kind of internal sensor of position or speed (only in the case of joints that use stepper motors are internal sensors not required). Internal sensors for robot arms must be easy to calibrate and adjust and produce clean (noise free) signals.

Internal sensors in industrial robots are mainly used to determine the position and orientation of the end-effector and individual axes. For this purpose mechanical, electrical, electronic and hydraulic devices to obtain can be applied. An example of mechanical device is the *governor*, which uses adjustable spinning weights to control the rotational speed of generator or motors [22].

Some hydraulic cylinders are equipped with *lead screw* that can detect the position of the cylinder by counting the revolutions of the screw as the shaft is extended or retracted.
For rotational joints, a shaft enconder can be used to detect precisely rotational displacements. Modern a*bsolute-readout enconders* can measure the degrees of rotation from 0.000º to 359.999º. I*ncremental encoders* emit a pulse for each increment of shaft rotation without an absolute reference.

Another device for measuring small shaft rotations is an electronic *strain gauge*, which in turn may use a piezoelectric device. Piezoelectric devices are crystalline materials that produce electricity when they are distorted.

Another device for angular positioning is the *synchro* system, which can transform an electrical input into an angular output or an angular position into an electrical output. Synchros are types of resolvers [22].

3.4.6 End-effectors and terminal devices

The end-effector could be thought of as the “hand” on the end of the robotic arm. An end-effector is a device mounted on the tool plate of the robot’s wrist to perform a particular task. The various types can be divided into two major categories: grippers and process tooling.

Grippers are end-effectors used to grasp, hold and transfers objects. They are multipurpose and have the following characteristic:

- Used to pick up and put down objects.
- Hold the workpiece. So that it cannot move within the gripper.

There are a variety of ways to grip a tool or workpiece, most of which are motivated by the nature of the item to be gripped. Some of these include:

- Mechanical gripping devices
- Vacuum cups
- Electromagnetics to pick up metal objects
- Adhesives
- Bayonet sockets that can be used to attach other tools
- Hooks

Process tooling is an end-effector designed to perform work on the part rather than to pick and place a work part. Tools serve a variety of roles in the manufacturing process. Some of the more significant robot end-effector tool applications include:

- Metal working: grinding, cutting, drilling, chipping
- Welding
- Surface treatments: coating, painting, finishing, cleaning, sealing
- Adhesive application
- Identification: marking, stamping

For the definition of the orientation of the end-effector, different approaches can be considered:

- Euler Angle Representations Z-Y-X or Z-Y-Z
- Euler Parameters
- Calculation of the rotation matrix
- Measuring of the angles of the wrist

3.5 Coordinate systems for industrial robots

In an industrial robot can be usually used three different coordinate systems: the world-system, tool-system and joint-system. The world-system offers a fixed system on which to identify the position of the points that compose the workpiece (see fig. 3.9). This is the main reference system. By using the tool-system the position of the hand effector is defined as well as its orientation. As shown in figure 3.10, by considering the joint-system, the angle of each individual joint of the robot arm is given.

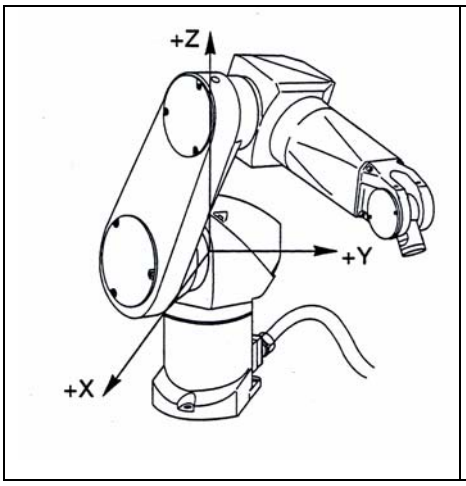


Figure 3.9 World-system

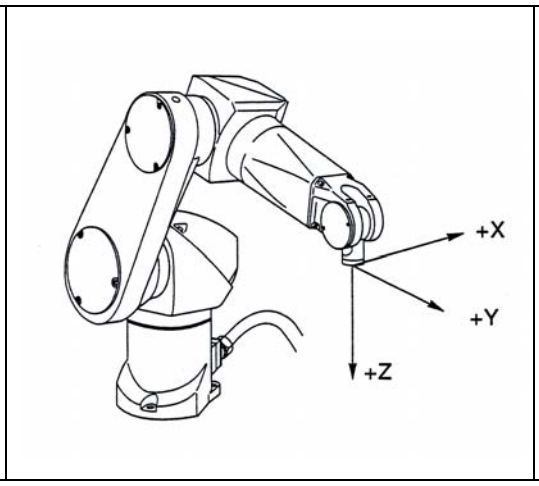


Figure 3.10 Tool-system

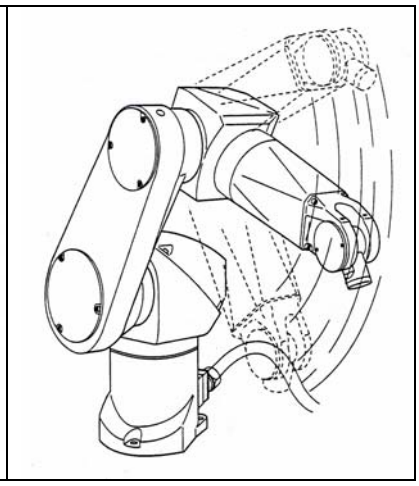


Figure 3.11 Joint-system

To identity the location of a body a reference coordinate system is established. The location of a body with respect to a reference coordinate system is known if the position of all the points of the body is known. If the body of interest is rigid, six independent parameters would be sufficient to describe its location in three-dimensional space. The location can be considered as composed of the *position* of a point (x,y,z) and the *orientation* of the moving frame with respect to the fixed frame (y,p,r), the last there component specify the orientation of the end-of-arm tooling. These three components are yaw, pitch and roll, these elements are figured as ZYZ' Euler values.

3.5.1 The representation of position in space

In robotics is usually use the Cartesian coordinated to locate a piece in the space, the position of a point with respect to the reference frame can be described by a 1×3 position vector. This vector is: P = [Px, Py, Pz].
Where Px, Py and Pz represent the projections of the position vector onto the three coordinate axes of the reference fixed frame (see fig. 3.1a).
Too we can use the polar or spherical coordinate to determinate the position of a point in the space (see fig. 3.2a and 3.3a) [3.3.4 Classification by workspace geometry].

3.5.2 The representation of orientation in space

The *orientation* of a rigid solid with respect to the fixed frame can be described in several different ways. We can use the *direction cosine representation*, the *screw axis representation* and the *Euler angle representation*.

Euler Angle Representations

The rotation is a motion with three degrees of freedom, a set of three independent parameters are sufficient to describe the orientation of a rigid body in space. In an Euler angle representation, three successive rotations about the coordinate axes of either a fixed coordinate system or a moving coordinate system are used.
The rotation de $\varnothing$ about the z-axis, generate the next matrix:

$$\varnothing \begin{pmatrix} \varnothing & \varnothing \\ \varnothing & \varnothing \end{pmatrix}$$

Similarly, when a rigid body performs a rotation of θ about the y-axis, the corresponding matrix can be represented as:

$$\theta \begin{pmatrix} \theta & \theta \\ \theta & \theta \end{pmatrix}$$

And finally the rotation is ψ about the x-axis. Hence the rotation matrix is:

$$\psi \begin{pmatrix} \psi & \psi \\ \psi & \psi \end{pmatrix}$$

We first consider three successive rotations of the moving frame about the coordinate axes of the fixed frame. Starting with the moving frame coinciding with the fixed frame, we rotate about the x-axis by an angle ψ, followed by a second rotation of θ about the y-axis and finally the third rotation of $\varnothing$ about the z-axis. The result is a matrix that is obtained by multiplying three basic rotation matrices:

$R(\psi, \theta, \varnothing)= M(x,\psi) \cdot M(y,\theta) \cdot M(z,\varnothing)$

The rotation about the x-axis is called a *roll*, the rotation about the y-axis is called a *pitch*, and the rotation about the z-axis is called a *yaw*. The convention to describing the orientation of a rigid body is roll-pitch-yaw angles representation, this angles representation is called *"X-Y-Z Euler Angles*" (see fig. 3.12).

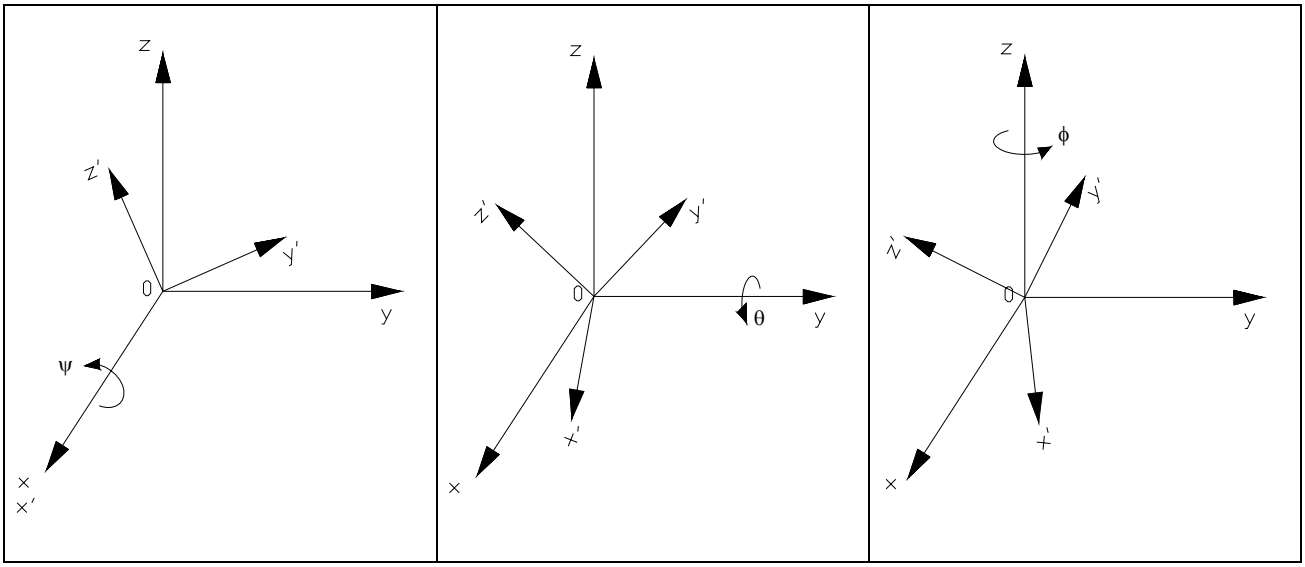


Figure 3.12 Successive rotations about the fixed coordinate axes [16]

A variation of this method consist of defining pitch, yaw and roll as follows: p*icht* is defined as a rotation about the moving reference frame Y axis, this is applied after *yaw*. The last rotation is *roll* and is defined about the Z axis of the moving reference frame, this is applied after *yaw* and *pitch*, (see fig. 3.13). This angles representation is called: "*Z-Y-Z Euler angles*".

Euler Parameters

Another representation of orientation is by means of four numbers called the "Euler Parameters". In terms of the equivalent angle θ and the equivalent axis $K=[kx,ky,kz]^T$, the Euler parameters are given by:

$\varepsilon_1= k_x \sin(\theta/2)$

$\varepsilon_2= k_y \sin(\theta/2)$

$\varepsilon_3= k_z \sin(\theta/2)$

$\varepsilon_4= \cos(\theta/2)$

It is then clear that these four quantities are not independent, but that:

$\varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 + \varepsilon_4^2 = 1$

must always hold. The Euler parameters are also known as a *unit quaternion* [17].

The rotation matrix, $R\varepsilon$, which is equivalent to a set of Euler Parameters is given as:

$$R\varepsilon = \begin{bmatrix} 1-2\varepsilon_2^2-2\varepsilon_3^2 & 2(\varepsilon_1\varepsilon_2-\varepsilon_3\varepsilon_4) & 2(\varepsilon_1\varepsilon_3+\varepsilon_2\varepsilon_4) \\ 2(\varepsilon_1\varepsilon_2+\varepsilon_3\varepsilon_4) & 1-2\varepsilon_1^2-2\varepsilon_3^2 & 2(\varepsilon_2\varepsilon_3-\varepsilon_1\varepsilon_4) \\ 2(\varepsilon_1\varepsilon_3-\varepsilon_2\varepsilon_4) & 2(\varepsilon_2\varepsilon_3+\varepsilon_1\varepsilon_4) & 1-2\varepsilon_1^2-2\varepsilon_2^2 \end{bmatrix}$$

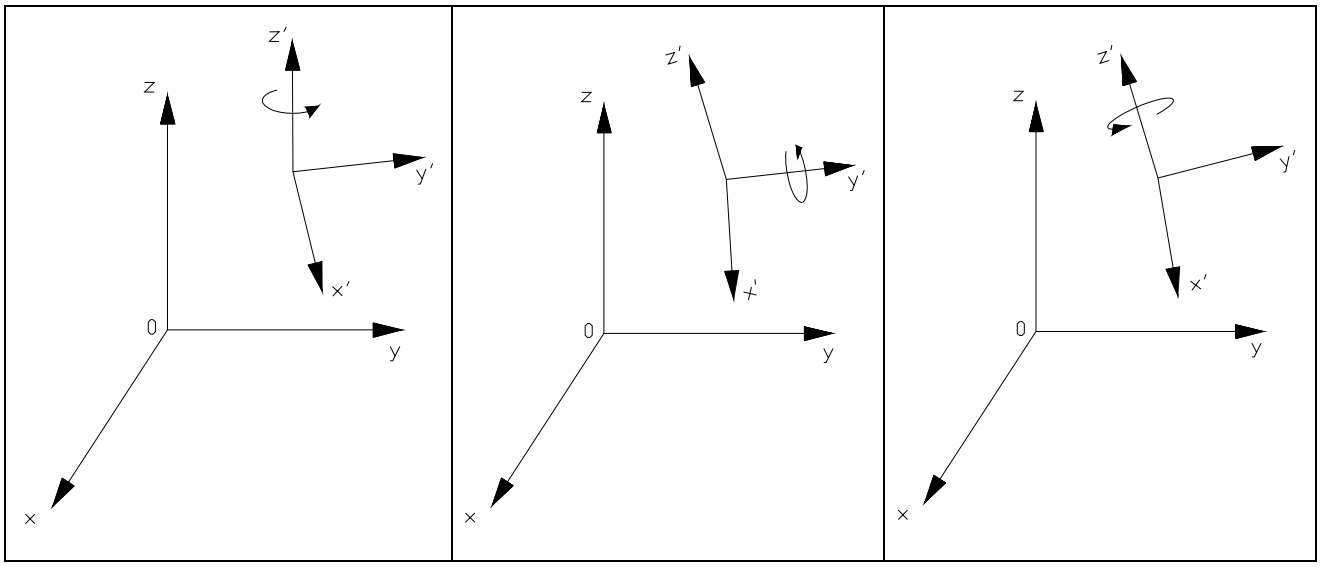


Figure 3.13 Successive rotations “Z-Y-Z”

An example of the representation of position and orientation of an arbitrary point in space by means of the world and joint systems is given by the table 3.1.

Position % Basis system in mm	**X 45,53**	**Y 35,28**	**Z -28,42**	**y 90,00**	**p -85,32**	**r 17,42**	
Angle-Position in grads	**Jt1 -3,47**	**Jt2 -85,98**	**Jt3 84,72**	**Jt4 -69,46**	**Jt5 6,43**	**Jt6 5,87**	**Hand 1,00**

Table 3.1 Example of representation of a point in space. World and joint system

3.6 Robot kinematics. Forward and inverse kinematics

Kinematics is the study of movement or motion without regard for the forces or torques that produce the movement. Kinematics thus includes the study of position, velocity, acceleration, and all higher derivatives of position. Dynamics is the study of movement or motion with regard to the forces or torques that produce it [18].
Kinematics of robot manipulator arms can be studied without needing to study their dynamics, because all the forces involved are generated and controlled at the Basic Level of control, and above this level, these forces and their consequences can be safely be ignored, as long as the speed and accelerations of the movements are not high. This is why we can talk of a Kinematic Level of Control, and not a Dynamic Level of Control. It is a situation special to industrial robot manipulator arms, and it is not true in general.

3.6.1 Forward kinematics

The problem of forward kinematics consists in determining what the position and orientation of the reference system attached at the end-point of the robot is, with respect to some fixed global reference system, given the values of the positions of each the joints, plus information about the types of joints and the geometry of the elements that connect the joints.

Given a reference system attached to each link of the robot arm, the position and orientation of the end-point (defined as the position of the origin of its reference system and its orientation with respect to the frame fixed to the base of the robot) can be calculated by the composition of the homogeneous transformation matrices which relate reference system on successive links.

In other words, the problem of forward kinematics can be reduced to the calculation of the relationship between two connected links in a robot arm, so a coordinate system will be fixed to each link, and then construct the homogeneous transformation matrix that represents the geometric relationship between them.

To do this in a convenient and accurate manner, all these link reference systems should be defined in the same way. The Denavit and Hartenberg method is the most used criterion to establish the link reference systems in robot arms [19].

3.6.2 Inverse kinematics

The problem of inverse kinematics consists in determining what values are needed for each of the joint positions, given a particular position and orientation of the reference system, plus information about the types of joints and the geometry of the elements that connect them [20].
To solve the inverse kinematics, two methods can be applied: solution of inverse kinematics by resolving the equation system given by transformation matrices and the solution based on geometrical relationships [21].

4 Robot programming

Programming is the identification and specification of a series of basic actions which, when executed in the specified order, achieve some specific task or realise some specific process. For the definition of these tasks different types of programming languages can be used.

Depending on the different types of basic actions specified in the programs developed, three types (or levels) of programming can be considered (see fig. 4.1).

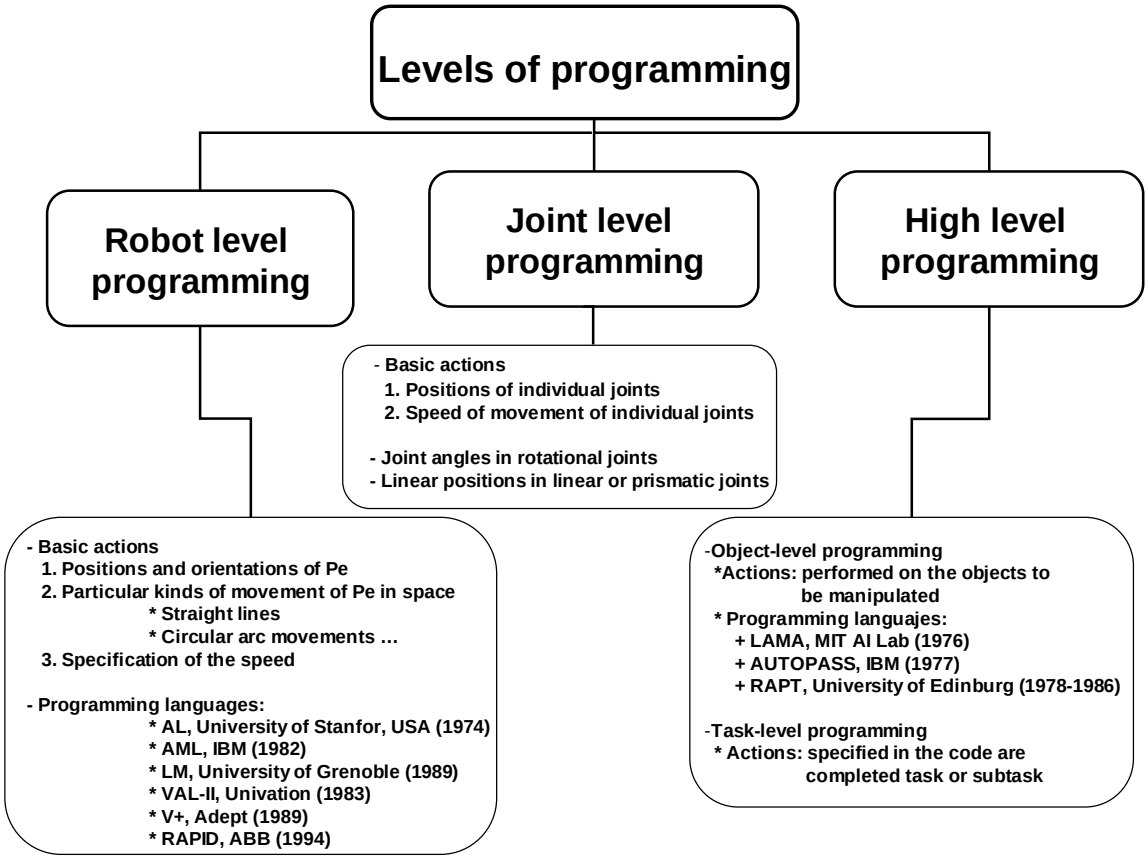


Figure 4.1 Flow chart for levels of programming

A common classification distinguishes between on-line and off-line programming. This work is focused on these strategies for the programming of industrial robots.

4.1 Robot programming methods for industrial robots

Robot programming methods are normally classified in:

- On-line programming methods, which use the robot to be programmed to generate the program.

- Off-line programming methods, which do not need access to the robot to develop the program. At least, not until the final testing of the program. It involves generally writing a program using a text-based robot programming language.

It is possible to use a hybrid programming, which has the advantages of both methods.

Another methods can be included, it is called “implicit programming”, which requires a sophisticated planning system, because no paths or positions have to be specified. The programmer only generates the goals of a task, which are then further detailed by the planner until the robot’s execution level is reached. Problems with this method are the high complexity of the industrial tasks, which require planning systems that are not available for industrial usage nowadays. This method is the most advanced of the two-mentioned methods for the robot programming [26].

Figure 4.2 shows the mainly methods used in the robot programming.

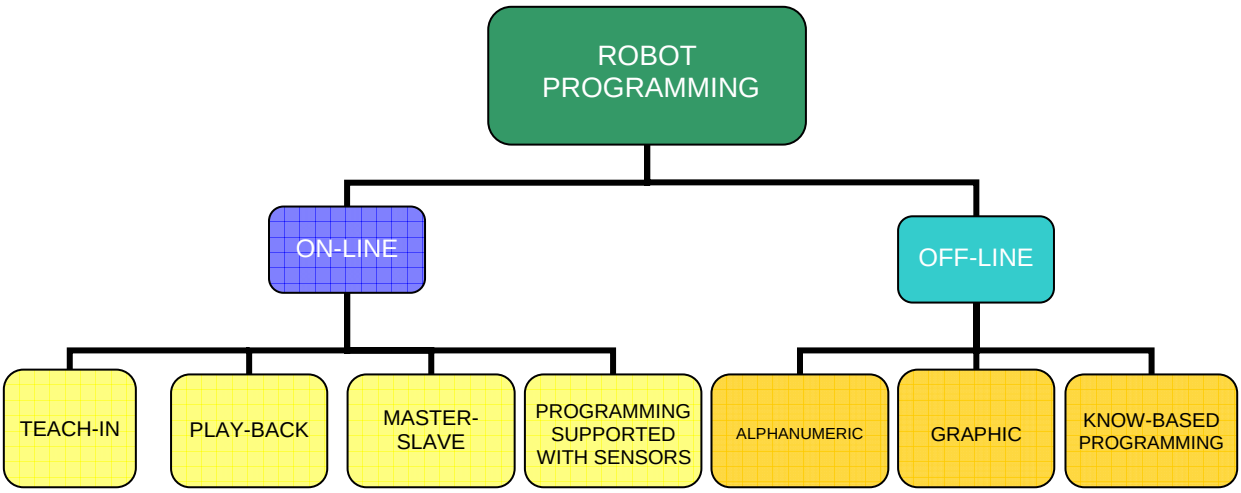


Figure 4.2 Robot programming methods

Table 4.1 shows any differences between on-line and off-line programming.

On-line programming	**Off-line programming**
Operational robots	Parallel mode of working, programs can be developed without the use of the robot
Sequential operation mode	Early examinations and code optimization
Limited possibility of simulation	Possibility of simulation
High accuracy in the movements is difficult to achieve	Reduction of programming time
It is difficult to incorporate external sensor data	External sensor and CAD data can be incorporated
Synchronisation or coordination with other robots is problematic	Programming not limited to the working cell

Table 4.1 Differences between on-line and off-line programming

4.1.1 On-line programming methods

Programming by guiding or programming by teach-in. This involves physically leading (moving) the robot arm through the movements and actions to be performed. Positions (and in some cases trajectories) are recorded by the robot system.

The different methods of on-line programming are teach-in, play-back, master-slave and finally programming supported with sensors (see fig.4.2).

⊗ *Teach-in*

The teach-in is the most popular method of programming point-to-point and continuous-path robots to lead the industrial robot through the task. This consists of moving the robot through the task, principally using the manual control pendant and when a desired point is reached, the operator stores coordinates into the robot's memory. The generated path to join two points of the trajectory can be defined as the movement of an axis or if two or more axes are moving.

⊗ *Play-back*

The play-back programming is used by low accuracy applications, it consist of taking the robot directly along the workpiece path by programmer manually. The robot controller is asked for the position of the axes, which are stored pressing a button.

⊗ *Master-Slave*

This programming method is mainly used in cases where the robot should move a big load. This method implies the knowledge of the kinematic model.

⊗ *Programming supported with sensors*

The very first thing this programming needs is to get some information of some support points of the robot's movement, these points will be measured by teach-in. After that, the robot path is calculated during the movement with the help of the sensors independent of the controller.

On-line programming has some advantages with respect to other types of programming.

- *Advantages of on-line programming*

  - It is only possible, positions are taught, can really be reached by the robot.
  - It is very easy to perform.
  - It is programmed in concordance with the actual position of equipment and pieces.

- *Disadvantages of on-line programming*

  - It is not practical for large or heavy robots.
  - It can not be used in hazardous situations.
  - The teach-in phase cannot be used during the production process.
  - The taught sequence cannot be altered.
  - The use of the robot and equipment is required.
  - Adaptation of the programs difficult.

4.1.2 Off-line programming methods

These methods involve the development of a text file, which contains the robot instructions and declarations that form the execution sequence. Indeed, in this text file other information, such as types of trajectories and maximum speed of movement are specified. Robot off-line programming is a method of combining computer simulations and graphics to produce a desired trajectory plan.

Off-line Graphical Robot Programming and Simulation are at the forefront of research in the field of industrial robotics. According to the *International Federation of Robotics*, as of 2007 over 950,000 robots have been installed for industrial applications in the world and an additional 100,000 are being sold every year. Currently less than 1% of these robots are programmed using off-line graphical robot programming (CAD/CAM). In comparison over 70% of CNC machines are programmed using computer aided design and manufacturing software [28].

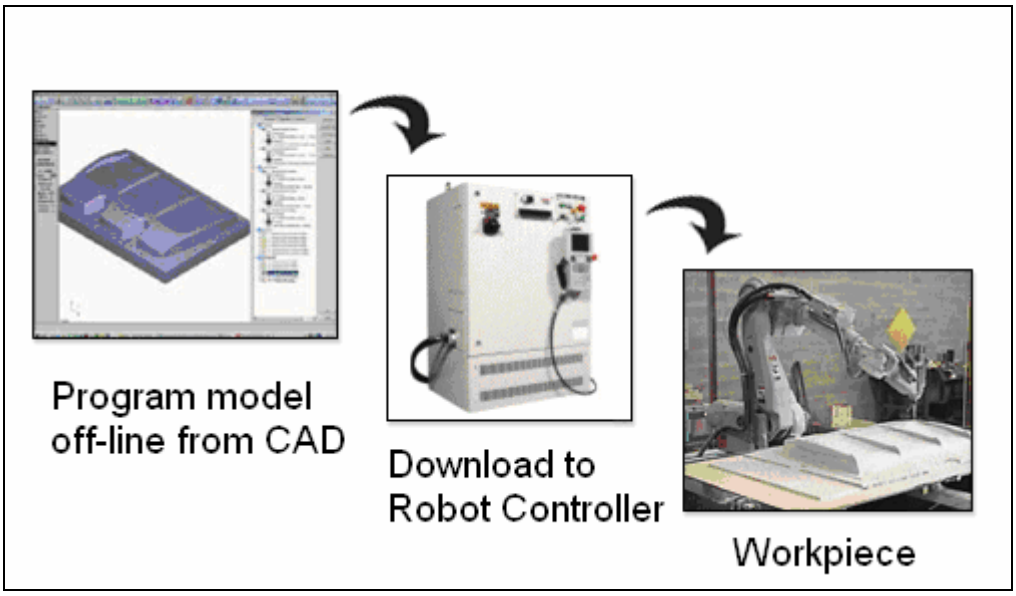


Figure 4.3 Steps in off-line programming

The different methods of off-line programming are alphanumeric, graphic and know-based programming (see fig.4.2).

⊗ *Alphanumeric*

This programming is based on introducing the robot program by the keyboard in a text editor. To work with this method it is very important to know the programming language of robot such as commands, movement instructions, position coordinates, tool orientation, etc.

⊗ *Graphic*

Graphic programming can simulate the working cell of the robot in 3D. With help of a graphic model the robot positions can be visualized in a screen. It is possible to find a lot of software that are working with this method, such as RobotStudio, Robotmaster, EasyRob, Delmia IGRID, Workspace, RodCad, Ropsim (Robot Off-line programming and SIMulation system). Some of these softwares will be described. Normally each manufacturer develops specific softwares for their robots.

1. *RobotStudio*

RobotStudio is an industrial software, which has been developed by the company ABB, being only suitable for robots of this company [27].
The main software tools are:

- CAD import

This program can easily import data in major CAD-formats, including IGES, STEP, ACIS, VRML, VDAFS and CATIA®.

- Autopath automatically

This is one of the most time-saving features. By using a CAD-model of the part to be processed, it is possible to automatically generate the robot positions needed to follow the curve.

- Path optimization

This software can automatically detect and warn about programs that include motions in close vicinity to singularities. It is possible to optimize speed, acceleration, singularity or axes to gain cycle time.

- Collision detection

Collision detection prevents costly damage to your equipment. By selecting concerned objects, RobotStudio will automatically monitor and indicate whether they will collide when a robot program is executed.

Also the company ABB presents softwares to specific applications such as arc or spot welding, assembly and packaging.

2. *Robotmaster*

Robotmaster is an add-on to the CAD/CAM software Mastercam, which integrates robot programming, simulation and code generation. This software has been developed by the company Jabez Technologies, which can be a solution in off-line programming for some robot manufactures such as Fanuc Robotics, Motoman, Stäubli, Kuka and ABB [28].
The main software tools are:

- CAD model generated or imported.

The CAD model can be generated into the program or imported from CAD software. Data translators are available for IGES, Parasolid®, SAT (ACIS solids), AutoCAD® (DXF, DWG, and Inventor TM files), SolidWorks®, Solid Edge®, STEP, EPS, CADL, STL, VDA, and ASCII, CATIA®, Pro/Engineer® and more.

- CAD/CAM based programming

The programming of the robot trajectory is done graphically using the same process and tools used for CNC machines, by selecting geometry (lines, arcs, part edges or 3D part). The software automatically generates the robot trajectory based on the above information.

- Convert CAD/CAM data to 6-axes robot output

It uses the graphical interface to fine-tune the parameters by which Robotmaster will translate the 2 to 5 axis CNC toolpath data into a 6-axis robot toolpath.

- Optimization of programs

It uses automated settings to quickly optimize robot motion and precise control of rotation around tool for:

  - Avoiding singularity and joint limits
  - Optimizing joint speeds and ensuring smooth robot playback

- Robot simulation

Validate the programs by using Robotmaster's robot simulator.

  - Automatic detection of collisions
  - View robot motion, by individual operation or complete toolgroup.

- Code generator

The post-processor is the module that converts the trajectories to robot specific off-line programming language. Each robot manufacturer has developed for their robots its own language.

⊗ *Know-based programming*

This programming is the higher step in off-line programming for industrial robots, which is based on the expert system and the knowledge of work. The expert system studies the task and generates a program for the robot depending on the specific task and the accumulated previous knowledge.

Off-line programming presents some benefits of using and some disadvantages with respect to other types of programming.

Advantages of off-line programming

- Increase of work safety, risk reduction.
- Increase of robot efficiency and productivity.
- Tasks optimization.
- Continuous production, it is not stopped.
- Programming of new parts without disturbing production.
- Integration of CAD-CAM system.
- Simulation of the real behaviour of the robot.
- Quicker start-up.
- Shorter change-over.
- The sequence of operations and robot movements can be optimised or easily improved, once the basic program has been developed.
- Previously developed and tested procedures and subroutines can be re-used.
- Programs can be tested and evaluated using simulation techniques.
- Programs can more easily be maintained, modified, documented and commented.

Disadvantages of off-line programming

- High initial expenses (software, worker formation, etc.).
- Interdependent engineering departments need to provide information at an earlier time.
- Calibration.
- Low precision, inaccuracy of positions.
- Possible information loss.

4.1.3 Programming tools for thermal spraying

Demands on robot programs for coating complex parts having 3D freeform surfaces have increased over the last years. Accurate programs are needed to ensure a constant coating thickness, a homogeneous structure, also a constant spray distance and transverse speed or a constant spray angle play important roles.
The quality of thermal spray coating is function of spray parameters but also of the gun trajectory. The trajectory generation represents an important part of work with industrial robots. The spraying trajectory is designed as a function of the working piece surface.

4.1.3.1 Thermal spraying applications for robot ABB

The manufacturer of robots ABB (Asea Brown Bowrie Ltd) has generated a thermal spray toolkit for thermal spray applications based on the software RobotStudio, which is devoted to use CAD and off-line programming in thermal spraying applications.
This toolkit is composed of three modules, which are PathKit, ProfileKit and MonitorKit.
The first module “PathKit” provides different methods to generate the trajectory of the torch on several surfaces, which can be round, square, curved or rectangular, also this module can generate the trajectory on rotating workpieces.
The second module “ProfileKit” allows coating surface analysis, with this module can be selected the right spray parameters, this coating parameters can be flow, coating distance, speed of the gun in the trajectory and more.
The last module is called “MonitorKit”, which can capture the robot displacement in real-time during the coating process, it is possible to have with this module all trajectory information, such as the speed at each point, the position of the points, this result can be exported into a test file.

Also, the software allows the work with the files CAD (Computer Aided Design) of workpieces, which can be imported to create trajectories. The different modules allow the generation of the trajectory on several surfaces. At the beginning the generation of trajectory is a laborious process, all the points have to be defined along the path needed on the piece, set orientation, speed parameters for each point and then joint all the points in a specified order to generate a trajectory to spray. If this process were manual it would be very laborious and inefficient, here the purpose was to generate a software toolkit to auto-generate the spraying trajectory [29].

The main characteristics of the different modules of this software are described in the following paragraphs:

PathKit

RobotStudio API is an Application Programming Interface for the users to call a procedure or read a value from a variable of RobotStudio in their extended program.
When an entity or a surface the program starts, it is also possible to introduce the input parameters to create the trajectory has been selected.

The first function of this module is to generate the gun trajectory on different surfaces.
The trajectory on a squared surface is called “Meander”, this program is frequently used when the robot works on plane surface or quadrilateral piece. The trajectory is composed by many well-regulated points. Whether the trajectory has been generated on a curved surface, the torch orientation at each point varies with the time in order to keep the torch normal to the surface. A trajectory curve and rotating workpiece are also frequently used in the thermal applications. The same trajectory on a square piece is sometimes used to spray on circular piece, but this procedure is not advisable.

To work with this software all the necessary to do is to import the CAD model of the workpiece into RobotStudio and select the right parameters of the trajectory and thus it will be automatically generated.

ProfileKit

This is the second module. “ProfileKit” allows an analysis of the coating surface, which is over roughness, fractal dimension, deposition of the coating, etc.

MonitorKit

This module can capture the robot displacement in real-time during the thermal spraying. For this, it is necessary to get the motion parameters of the robot, which are space position, tool orientation, robot posture, such as the tool orientation is calculated with the tool centre point coordinates P(x,y,z), these coordinates can be easily obtained from the controller and the tool orientation is written by a quaternion array.

In order to achieve this, the program was then exported into a real robot and the real trajectory was captured by MonitorKit.

4.1.3.2 Off-line programming for spraying of 3D surfaces

At the Fraunhofer Institute for Material and Beam Technology, equipment for atmospheric plasma spraying (APS) and vacuum plasma spraying (VPS) as well as high velocity oxifuel (HVOF) spraying is available. For these processes, the KUKA industrial robot with rotate/tilt positioners and linear unit is used as the handling system for the spray process [30].
At this institute a process by means of off-line programming for coating of 3D surfaces has been generated. This process presents three main steps.

First Step: CAD data for calculation

The CAD data generated during the design process are used for calculations, if these data are not available the part must be digitized by reverse engineering, these methods based on optical effects can be laser triangulation or coordinated measuring machine among others. These measurements end in a cloud of points, from which surfaces of the model can be created using algorithms.

Second Step: Robot Programming

The robot program is generated using the CAD data. In this step the path strategy for movement along the part surface is defined, different parametrized functions (3D spiral path, crossing path, etc.) can be choosing. It is possible to introduce the path parameters (distances, speed, etc.).
The robot motion simulation can be performed with a rendered image for visualization, can be detected possible collisions. If is not possible to keep a 90-degree spray angle by collision problems, spatial reorientation of the tool can be smoothed using special calculation algorithms or manually.
If the path definition is finished, a directly executable robot program can be generated.

Third Step: Simulation

Two software components are used for the simulation, the first one is KR C1 Office, which is used to an exact virtual representation of the Kuka robot controller and the second one is KR Sim for projecting and planning of the entire robot system.

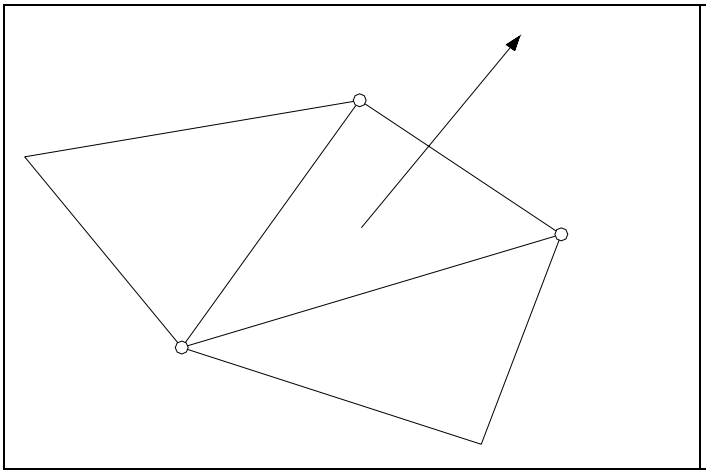
These softwares allow to run and to verify virtually the coating process in real time, it is possible in this step to solve collisions, end positions and other problems.

4.1.4 Robot program development process

4.1.4.1 Formats used in design programs

There are a great number of formats to keep the information of the generated designs, such as STL, IGES, STEP, ASCII, ISO-G-CODE, UDAS, DWG, ACIS and more, some of these formats will be described.

- STL-Format (Standard Transformation Language)
  - Too als StereoLithography.
  - The geometrical description by means of STL-Format is based on triangulation of surfaces. In this way, spatial coordinates of the vertexes and the normal vector for each triangle are given (see fig. 4.4). Since for the calculation of the orientation of the torch the corresponding normal vectors are needed, the STL-Format will be in following used.
  - This format keeps the piece in a mesh, which is composed by triangular elements. Each triangle is represented by three vertexes and a normal vector to its surface. This format represents the coordinates of the vertexes and the normal vector.



```
facet normal -9.956242e-001 -8.037510e-002 4.766797e-002
     outer loop
        vertex 9.662602e+000 2.691262e+001 8.735000e+001
        vertex 7.725023e+000 2.536364e+001 4.535000e+001
        vertex 9.885985e+000 2.414552e+001 8.735000e+001
     endloop
  endfacet
```

Figure 4.4 Graphic representation of the STL-format

- IGES-Format *(Initial Graphics Exchange Specification)*
  - It defines a neuter format of data that allows the digital exchange of information among design systems attended by computer (CAD).
  - Using IGES, the user can exchange information of the model in the form of outlines, surfaces or solid representations. Among the applications that support are included drawings, traditional models for functions and manufactures analysis.
  - The data are information of surfaces (nurbs, spline).

- STEP- Format *(STandard for the Exchange of Product model data)*
  - It is a Standard for the exchange of data of products.
  - It describes the information of a product through the cycle of life of the same one, independently of any systems in particular.

- It allows the exchange of data among CAD, CAM, FALLS, PDM/EDM, and other CAX systems, it supports models of mechanic design, electric, analysis and it manufactures, among others.

- ASCII- Format
  - This format includes information of points, which is made by means of the coordinates.
  - A representation of the structure of ASCII format is the following:

```
67.32     -68.99    -239.36
76.60     -72.05    -223.54
66.70     -72.05    -235.29
76.40     -76.05    -220.04
66.91     -76.06    -231.09
57.18     -76.06    -242.48
85.86     -79.17    -215.34
77.00     -79.17    -217.94
67.55     -79.18    -228.00
```

- ISO-G-CODE
  - This format is used in the numeric control programming, CNC (Computer Numerical Control).
  - (N0,X=X1,Y=Y1,Z=Z1) for installation of machines tool.
  - A representation of the structure of ISO-G-CODE is the following:

```
%0  G71*
N001  G99  T0  L0  R0*
N002  G17  T0  G0*
N003  G00  S40  G90  M03*
N010G0X-28.394Y-4.259Z50.003*
N011G1X-27.484Z50.539F500*
…
N032G1X-8.034Z57.302*
N033G1X-6.583Z57.490*
```

4.1.4.2 Geometrical data acquisition and program process

Tools for the acquisition of geometrical data can be used CAD-models or reverse engineering, it is also possible to take geometrical data by teach-in.
The figure 4.8 shows the process to data acquisition for the generation of robot trajectories for thermal spraying. The three main methods used for geometrical data acquisition will be developed.

Teach-in

This is a process of on-line programming. The measures of the needed data are carried with the Manual Control Pendant directly in the cabin of robot. This process is useful when the geometry of the piece is simple or whether exists some coating program already that needs data to be initialized.

The figure 4.5 shows the flow chart to the teach-in programming method, which is composed by the building of program, representation of the trajectory with the help of some subroutine generated in Matlab (simulation) and finally the implementation of program in robots.

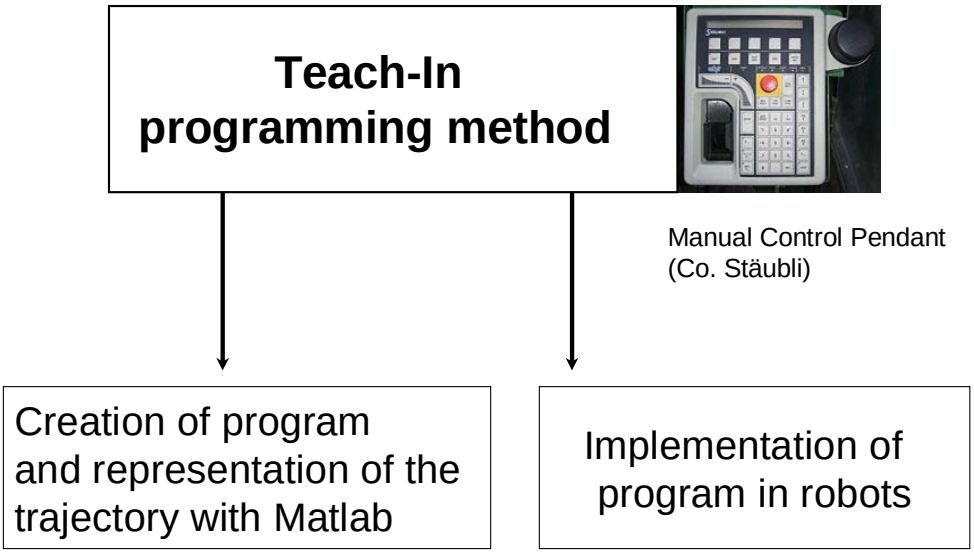


Figure 4.5 Flow chart for the generation of robot trajectories for thermal spraying applications by using Teach-in

If the pieces present complex geometries a design of model in CAD or the reverse engineering are used, these processes are used with off-line programming.

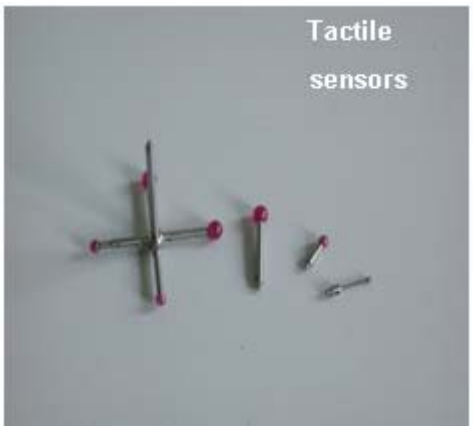
Reverse engineering

The reverse engineering process consists of measuring of the points coordinates that compose the piece by means of an optic or tactile system, so it is possible to generate a discreet model of the piece starting from a cloud of points.

One of the many definitions of reverse engineering is taking apart an object to see as it works in order to duplicate or enhance the object. In the practice, taken from older industries, is now frequently used on computer hardware and software.

Reverse engineering has become a viable method to create a 3D virtual model of an existing physical part to use in 3D CAD, CAM, CAE and other softwares. Reverse engineering involves measuring an object and reconstructing in a 3D model. The physical object can be measured using 3D scanning technologies such as CMM (coordinate measuring machine), laser scanners, structured light digitizers or computed tomography. The measured data alone, usually represented by a cloud of points, lacks topological information and it is therefore often processed and modelled into a more usable format such as a triangular faced mesh (STL), a set of NURBS surfaces or a CAD model.

According to the figure 4.12 when the process of date acquisition is carried out for the reverse engineering, the data are initially a cloud of points. With the help of software these points are drawn to generate a surface and to obtain the normal vectors of the same one and to be able finally to create the mesh.



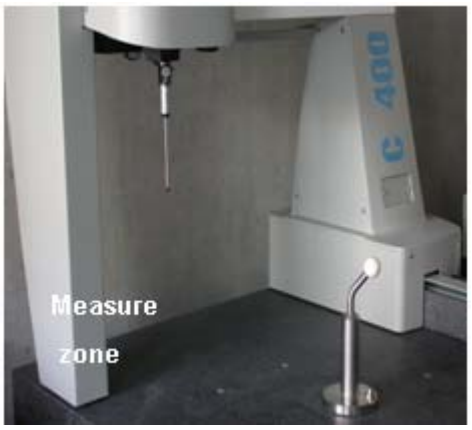


Figure 4.6 Sensors of coordinate measuring machine

Figure 4.7 Coordinate measuring machine of IFKB (University of Stuttgart)

When the surface is already generated, it is possible to simulate the trajectory of the robot's gun. Initially this trajectory is referred in the coordinate system of the machine's reverse engineering, it is necessary carry out a change of coordinates to the coordinate system of the robot. Finally the code of the program is generated in the robot's language that makes the coating.

The data of reverse engineering can be used for the simulation of the piece in a program of finite elements, such as ABAQUS or ANSYS with the purpose of obtaining the possible tensions, deformations and temperatures that are reached from the piece's coating.

CAD-models

The design of CAD model is another method to geometrical data acquisition of pieces. In this case a three-dimensional model is generally generated starting from the geometrical data of the piece. Some design softwares can be used to do this step such as SolidWorks® (SolidWorks Corporation now a subsidiary of Dassault Systèmes, S.A), CATIA® (Dassault Systèmes, S.A) or Pro/Engineer (Parametric Technology Corporation).

The work process begins with the generation of perpendicular sections over 3D-model of the piece. These sections are guided according to the trajectory that follows the robot's gun. The information of the cut surface is saved in the STL-format, obtaining a meshed surface with triangular elements. This format gives information of the vertexes coordinates and the normal vector for each triangle. Additionally, a filtration of double points has to be carried out over the information saved with the STL-format. Ordering the points and their normal vectors can carry out a representation of them with help of some subroutines of Matlab. Carrying out a change of the coordinate system over the resulting data, it is already possible to generate the robot trajectory, see figure 4.8.

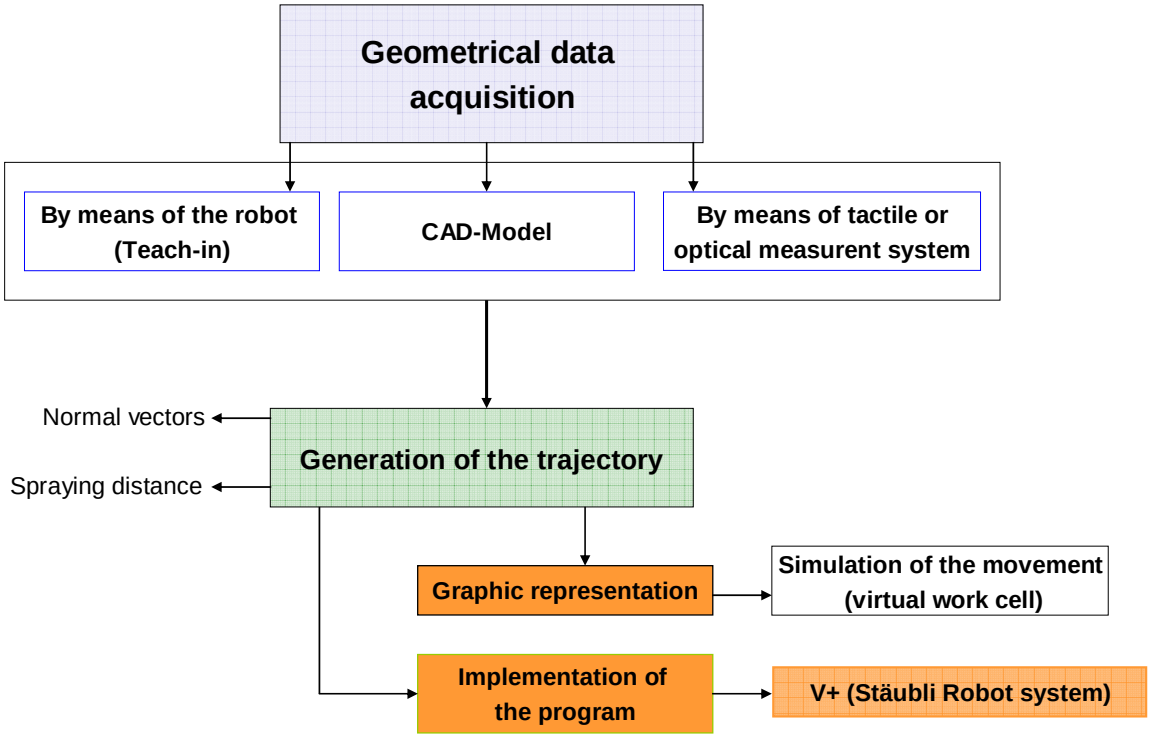


Figure 4.8 Flow chart for the generation of robot trajectories for thermal spraying

4.1.4.2.1 Teach-in

By means of the teach-in programming method, the Cartesian coordinates of the points which are included in the robot trajectory are measured with the Manual Control Pendant in the manufacturing cell. As case study, the KBA-blades will be considered.

KBA-Blade

This pieces are property of the company König und Bauer AG. The geometry of pieces is very simple, with any little points it is possible to define the form of the pieces. This method is very effective to this case.
Figure 4.9 shows some KBA-blades, already these pieces have been coated by APS process.

Figure 4.9 KBA-blades (Co. König und Bauer AG)

In this way, the interpolation points have been captured to define the curve shown in the figure 4.10. The calculation of the complete trajectory is carried out by using the Matlab-subroutine 4.1 (see Annex 4). A set of parallel curves (with the required off-set in Z-axis, see fig.4.11) are defined and the interpolation points arranged in the meander sequence. The disposition of the parts in the cabin of the robot was made with double pieces, each row contained two consecutive pieces along the "Y-axis" and were installed six rows in the direction of the "Z-axis" in the world coordinate system of the robot. KBA blades present a straight part, which has been aligned in the two parts to facilitate the coating process.

The measured coordinates of the interpolation points can be represented with the Matlab-subroutine 4.1 (see fig. 4.10). A coating program is designed directly in the language of the robot, which requires entering the coordinates of the first line of motion, for which was measured in the cabin of the robot the position and orientation of the torch in eight points. The base for defining a point in the robot space is by means of the coordinates "x", "y", "z" and the normal vector "nx", "ny", "nz". These eight points initialize the program for the generation of the movement in the gun of the robot.

A representation in the XY plane of the eight measured points has been presented in figure 4.10.

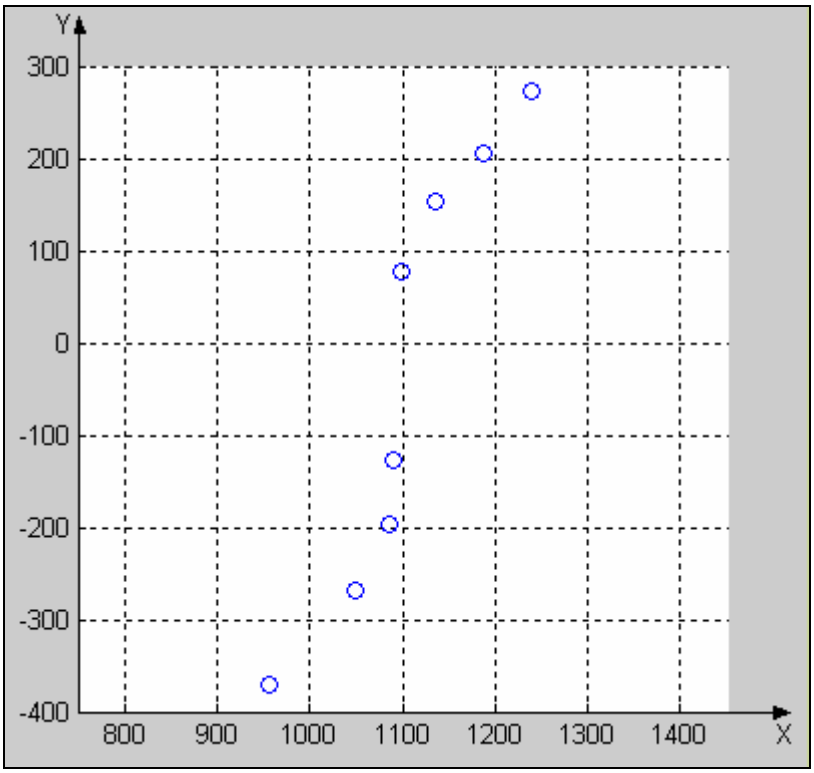


Figure 4.10 Representation of the measured points in the cabin of the robot (mm)

An off-set of 3 mm has been defined, which is a standard value commonly used in thermal spraying applications to ensure homogeneous coating thickness. The figure 4.11 shows the resulting robot trajectory with the corresponding points referred to the world coordinate system of the robot arm.

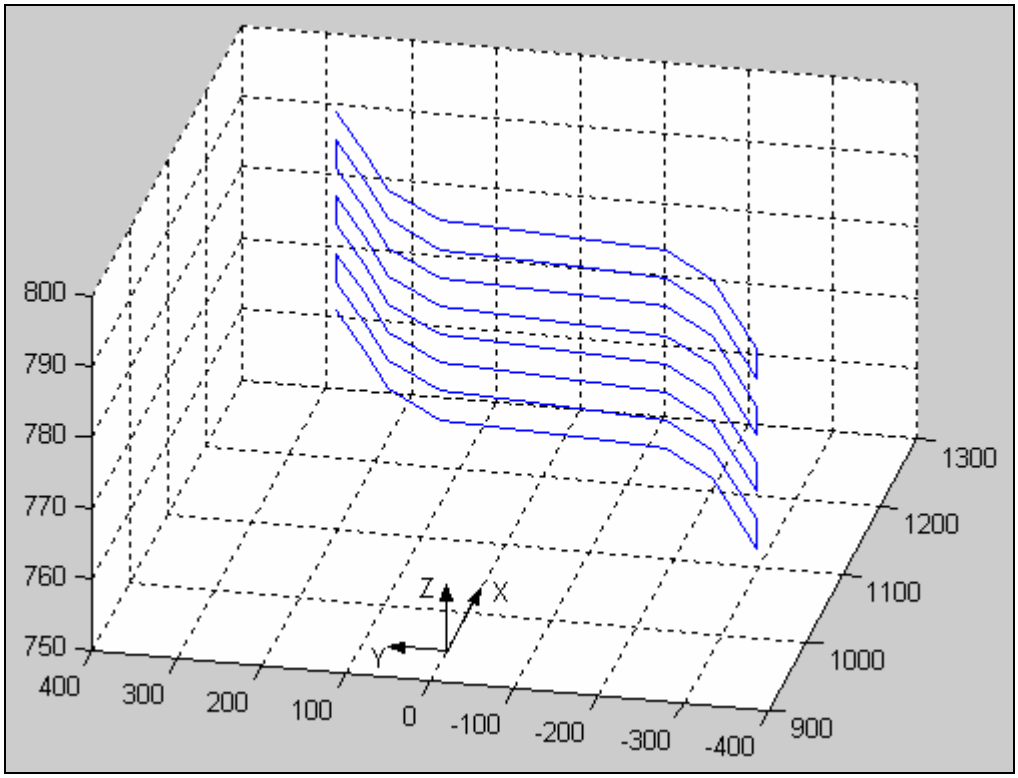


Figure 4.11 Simulation of the trajectory in KBA pieces

The generated path is included in the code of the subroutine Kbauer3.v2. This subroutine (see Annex 4) also contains the required commands for the robot speed, positions for ignition and tool replacement.

4.1.4.2.2 Reverse engineering

The robot trajectory can be also generated by using the geometrical information obtained by the coordinated measuring machine or laser scanner system.

The steps to develop a robot program in this case are:

1. Generation of the point cloud.
2. Definition of the surface to be coated.
3. Calculation of the normal vectors of the surface.
4. Generation of the gun trajectory by considering the points of the mesh and the obtained normal vectors.
5. Generation of program code in the required programming language, in this case V+ (Stäubli robots).
6. Implementation in the robot cell and coating.

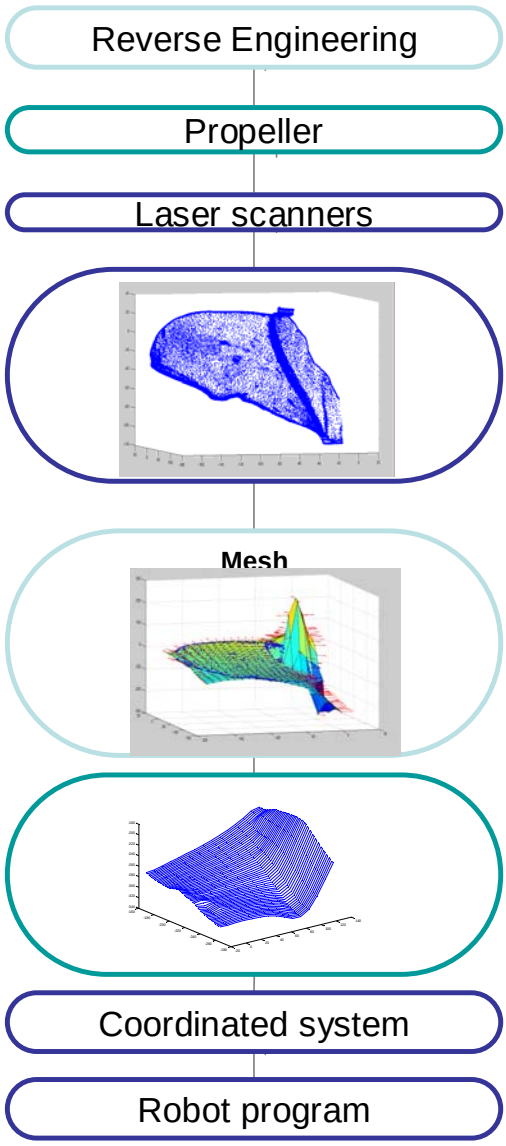


Figure 4.12 Reverse engineering applied to propeller

This case of reverse engineering will be studied using a propeller, the coating will be carried out by this method.
To try to generate a robot program of this piece by teach-in can be an inefficient and inadequate process because a lot of points are necessary to define the robot trajectory. Also to draw a model of this piece in a CAD software design can be very laborious and complex, due to the specific shape of this piece and that it does not present any symmetry type in the part of the cross.

Using the reverse engineering can be solved the problem, in this case it is possible to generate the surface mesh by means of the information of the points coordinates of the propeller and finally the curve of the coating trajectory.

PROPELLER

In this case the measures were taken by the coordinated measuring machine and laser scanners, the measures take with the laser scanners have been used to the propeller's coating.
The generated file with the laser scanners consist of 27458 points, which is composed of the coordinate "x", the coordinate "y" and the coordinate "z" (see Annex 4).

The measured points can be represented with the Matlab-subroutine 4.2 (see Annex 4). As shows the figure 4.13, it is possible to observe the quantity of points that were measured in the laser scanners, which represent only a third part of the piece's total.

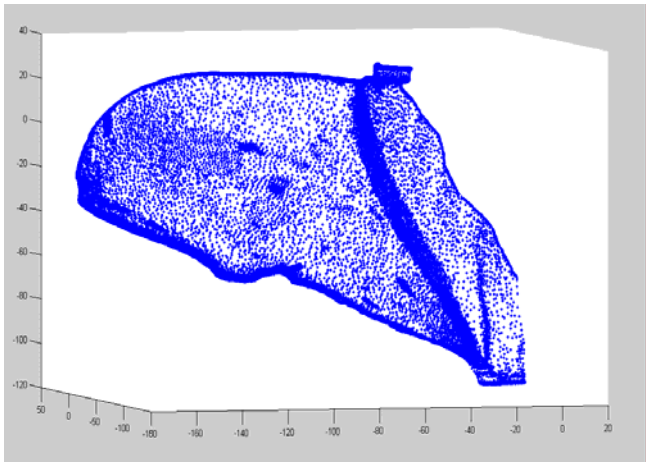


Figure 4.13 Representation of measured points by laser scanners

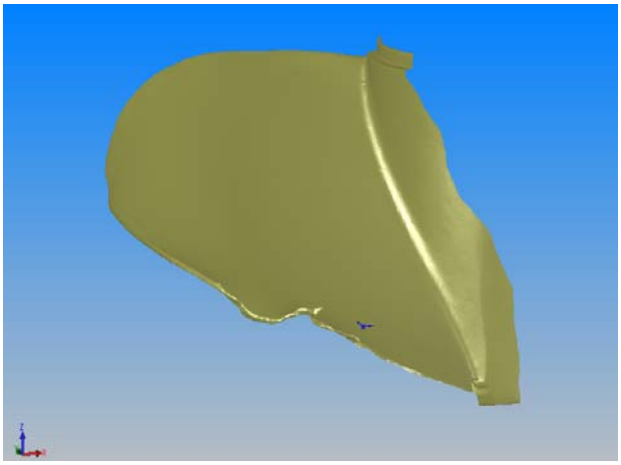


Figure 4.14 Representation of measured points by laser scanners in a design program

A representation of the measured points with the laser scanners can be obtained with the design program SolidWorks, which represents a surface of the propeller, as shown in figure 4.14.

The main steps to create the robot trajectory by means of reverse engineering can be the following:

- First step

To make a filtrate of the points, a representation of these points is reflected in figure 4.15.

- Second step

To generate a mesh support in the measured points, the mesh is represented in the figure 4.15 with the normal vectors of every cell of the surface. This representation has carried out with the Matlab-subroutine 4.3 (see Annex 4).

This mesh shown in figure 4.15 has an error in the cylindrical part of the propeller, there are not points here, so the mesh fails its value. Something similar happens in the corners, the mesh has been defined as a rectangular surface, the propeller does not present points and has an erroneous value in these positions.

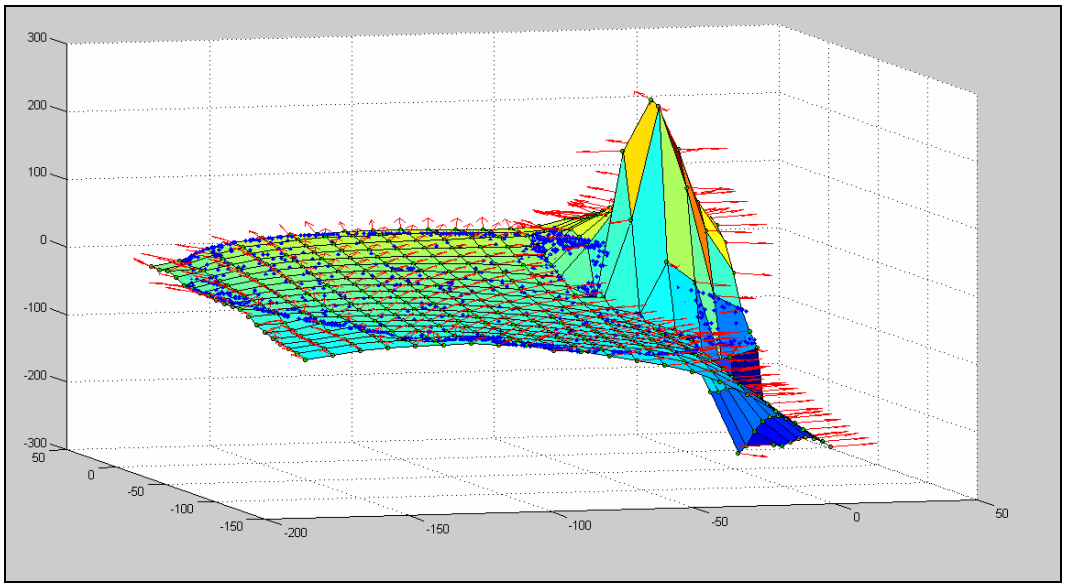


Figure 4.15 Representation of the mesh and normal vectors

In brief, a filtrate of the measured points by the laser scanners and an appropriate mesh allows the definition of the gun trajectory. In order to get this is advisable to have a rectangular mesh, where each cell had a thickness between 3-5 milimeter in the direction of Y-axis and a maximum thickness of 10 milimeter in the other main direction of the cell.
The main direction of movement is given by the direction with bigger thickness, while another direction is coincident with the off-set.

Figure 4.16 shows the representation of the trajectory of the gun in the coordinate system of laser scanners.

Additionally, it is necessary to make a change of coordinates to the coordinate system of the robot, which is basically to apply a translation movement and a rotation matrix.
To make this process of the coordinate transformation it is necessary to know at least the coordinates of three points onto the propeller in the coordinate system of laser scanners and these same points must be identified on the piece to obtain with the robot by teach-in their coordinates in the World-System robot.

The subroutine to the coordinate transformation generates a data file with the new coordinates of the points but in the coordinate system of the robot.

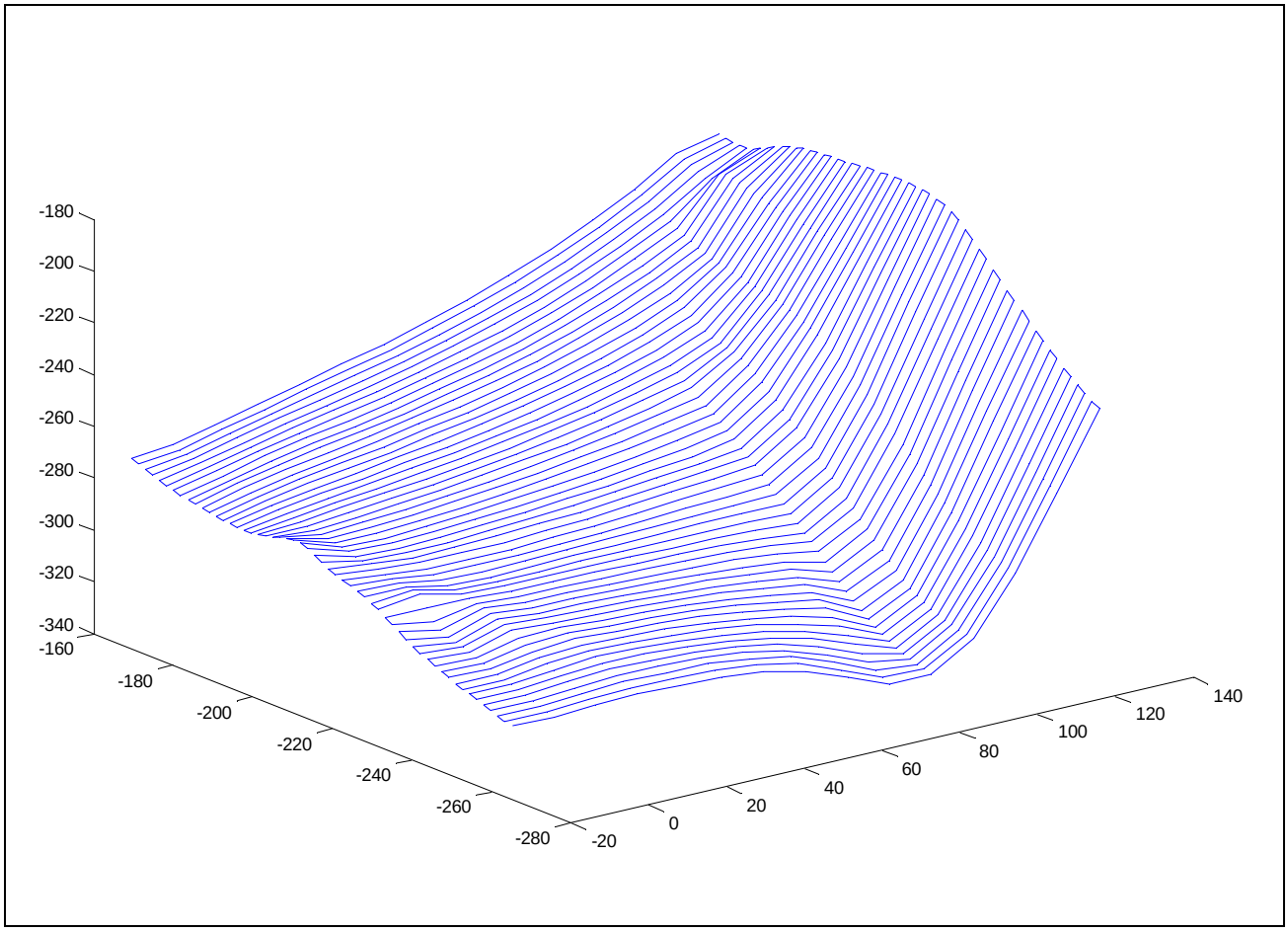


Figure 4.16 Simulation of the coating's trajectory

The figure 4.16 shows the simulation of the coating's trajectory, which presents the third part of the propeller.

The generated program in the robot's language to make the coating of the propeller reads the new coordinates of the points and creates the trajectory of the robot's torch through the movement among these points.

4.1.4.2.3 Design Aided Computer (CAD)

The generation of the robot trajectory by using the geometrical information of the STL-model includes the following main steps.

1. For the processing of STL-Files different commercial software tools can be found. Some examples are SolidWorks, CATIA and Pro/Engineer.

2. As shown in figure 4.17 (Data acquisition for the path definition by using a CAD-model), the geometrical points which will be used for the generation of the robot trajectory are calculated by intersection of the surface with orthogonal planes.

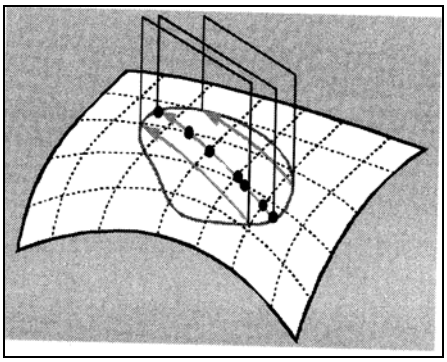


Figure 4.17 Data acquisition for the path definition by using a CAD-model

3. It is necessary the adaptation of the registered points, the leaking of points collection from non useful coordinates.

4. By means of the Matlab-subroutine 4.4 (see annex 4) it is possible to display the selected points and corresponding normal vectors.

5. To be used for the calculation of the robot path, the given Cartesian coordinates must be transformed into an absolute coordinate system in the working cell of the robot.

6. The generation of the trajectory requires the array/collocation of the spatial points in the desired sequence. Within the scope of thermal spraying meander paths (see fig. 4.26) are commonly applied. By the definition of the robot path as a meander trajectory homogeneous coating thickness and reduced material wastage/usage can be achieved.

This process will be applied for the programming of the robot trajectory to coat spherical joints (see fig. 4.19). Also this process will be used to calculate the robot path of the KBA piece.

CASE STUDY FOR THE GENERATION OF THE TRAJECTORY BY USING A CAD MODEL. BALL PIVOT

The ball pivot is a component of the company Alfred Heyd GmbH and co. KG. The function of this component of the joint rod consists of the stabilization and vibration absorption at high speed trains. This piece is used in the Spanish high speed trains called “AVE”, the figure 4.18 shows the ball joint, the position of this piece in the train (joint rod) and a picture of this train.

The properties required for this piece are high loading capacity, good solidity, without corrosion, wear resistance and a long live.

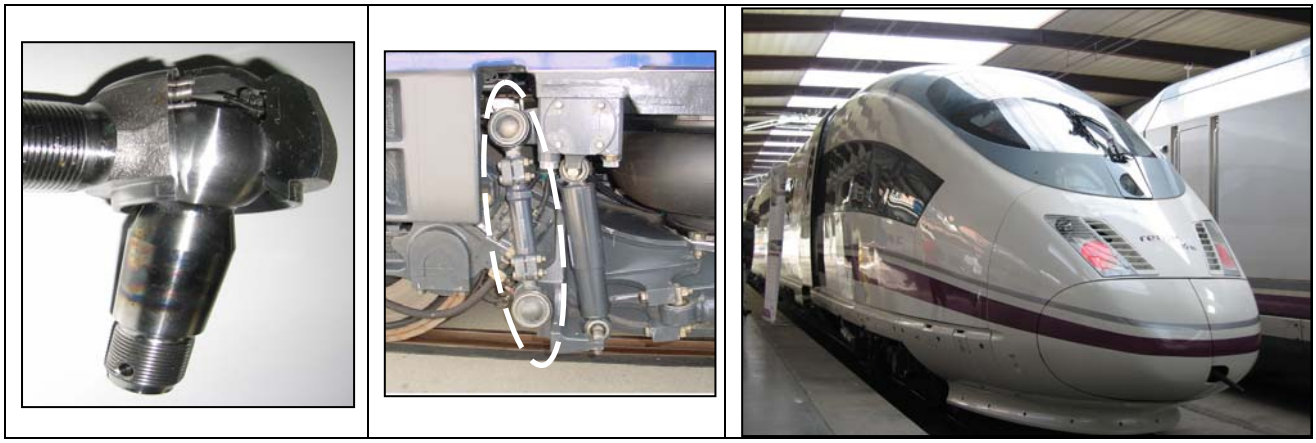


Figure 4.18 Ball joint, joint rod and high speed train (AVE)

Starting from geometry data is possible to created a CAD model, the figure 4.20 shows the piece model in software design SolidWorks, also is represented the trajectory that would follow the robot's torch.

The figure 4.19 shows the real image of "Kugelzapfen" piece.

Figure 4.19 Real image of workpiece (Co. Heyd)

Due to condition works it is absolutely necessary to coat a part of the sphere, this coating surface is shown in the figure 4.20.

A coating process under rotation of the component requires a manufacturing device which guarantees a reliable and technically safe rotation process.
The piece will be sustained by a plate of claws, by this way it is possible to have an axis of rotation or revolution.
Due to the rotation of the piece, the robot trajectory can be defined only with a section (see fig. 4.21).This section presents a thickness about 0,1 milimeter, it is therefore a very small thickness can be considered that the robot trajectory presents a constant value of the coordinate "z".

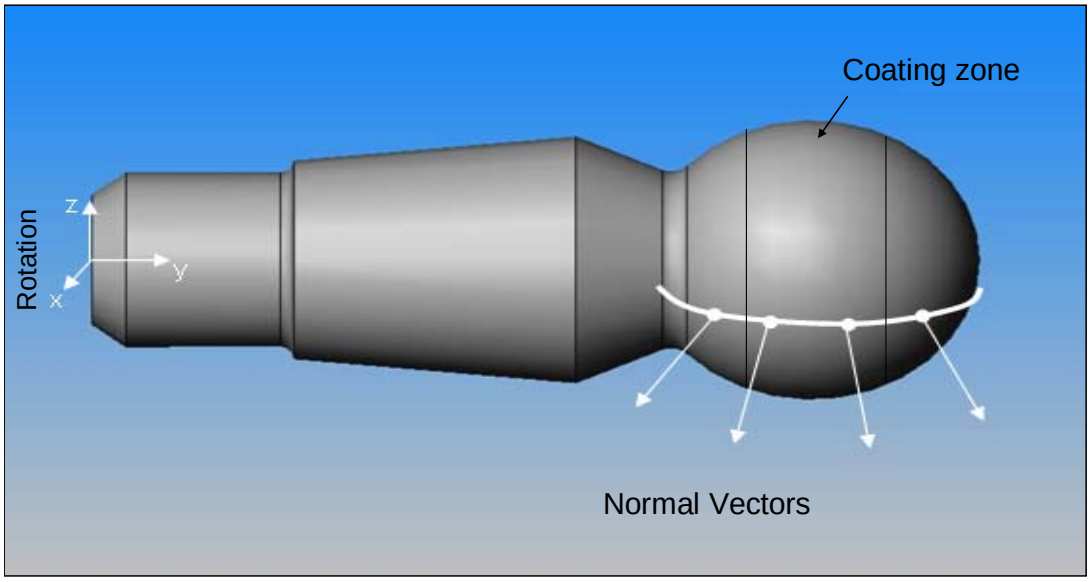


Figure 4.20 Description of the trajectory and the coating area

Already have been obtained the section for the movement of the robot, in this case is centred solely in the part spherical. It is necessary to store the coordinates of the surface in a file, it is possible to do it with help of STL-format, that transforms the surface in a mesh of triangular elements, which keeps the coordinates of the vertexes of each triangle and the normal vector to the surface of the same one.

In the figure 4.21 is shown the graphical representation of the component.

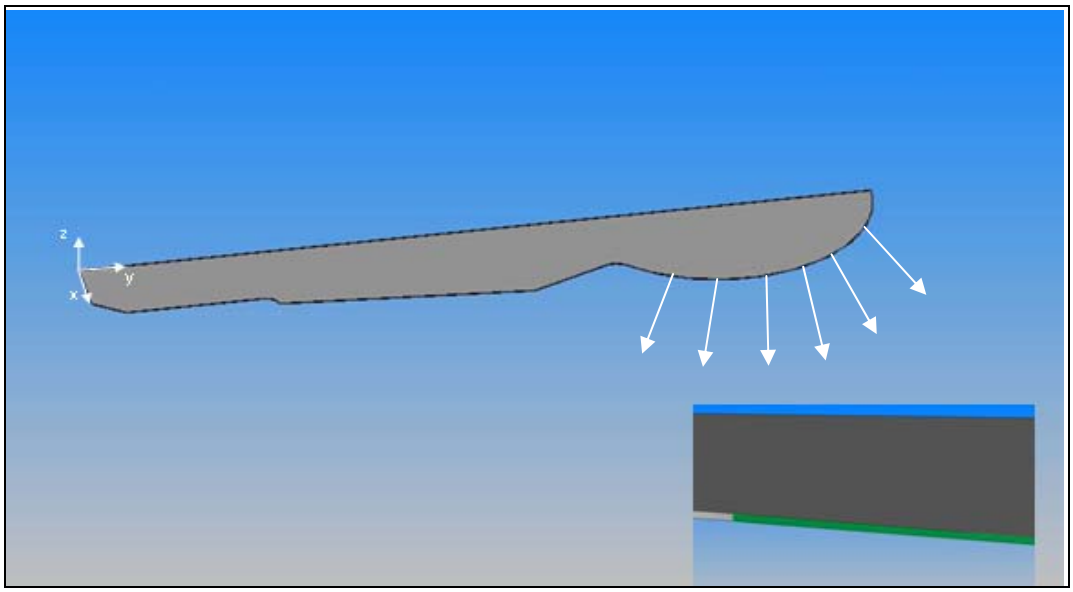


Figure 4.21 Section of the Kugelzapfen piece

The mesh of this superficial section is composed initially by eighty four triangles, it is necessary to extract of the file the information of points coordinates and the normal vectors associated.

The file is composed by a series of lines, where the columns of this lines represent the coordinate "x", the coordinate "y", the coordinate "z", the coordinate “normal vector x", the coordinate “normal vector y" and the coordinate “normal vector z".
A representation of the points and their normal vectors is shown in the figures 4.22 and 4.23, which have been generated with the Matlab-subroutine 4.4 (see annex 4).

The next step consists of the adaptation of the registered points, to obtain it, it is necessary the filtrate of points collection, this process is manual, the quantity of necessary points to generate the robot path is fundamentally function of specific geometry and the programmer’s experience.

In this case of study, it is only possible with six points to describe the gun trajectory as shown in the figure 4.24.

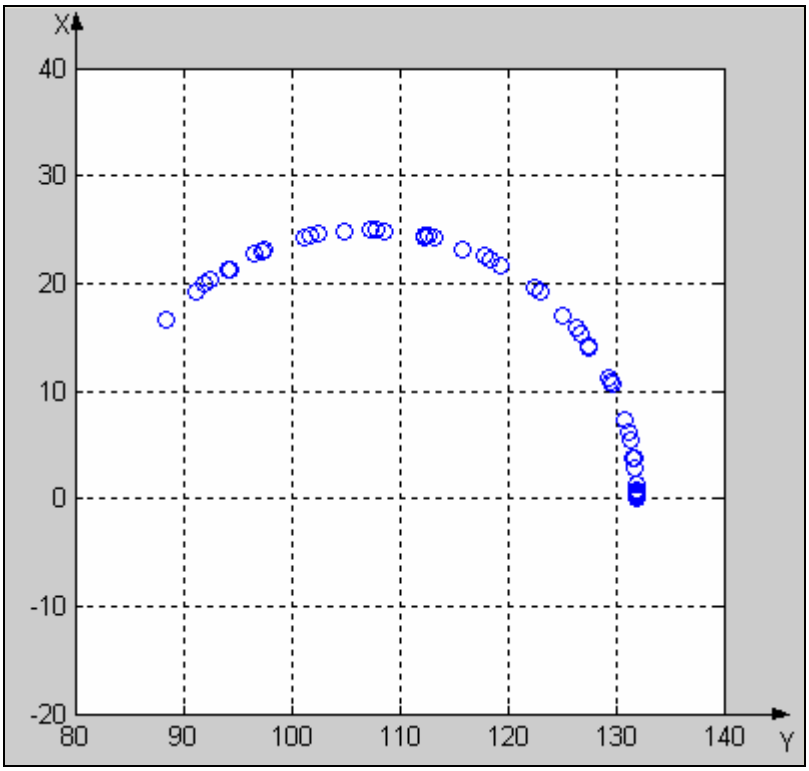


Figure 4.22 Representation of the points into the plane XY

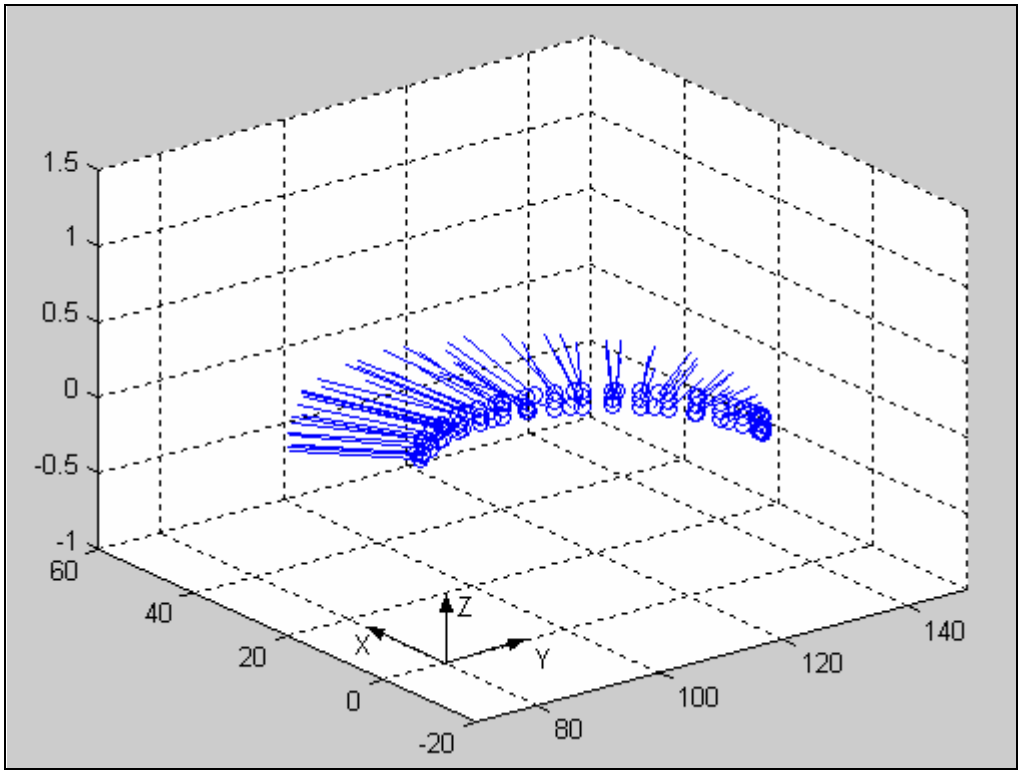


Figure 4.23 Representation of the points and their normal vectors

It is possible to generate the robot path as a function of the points that compose the superficial section of the sphere. Coordinates of the points are referred to the coordinate system of the design software, it is necessary to adapt them to the coordinate system of the robot, this can be made with a translation movement and a rotation matrix.

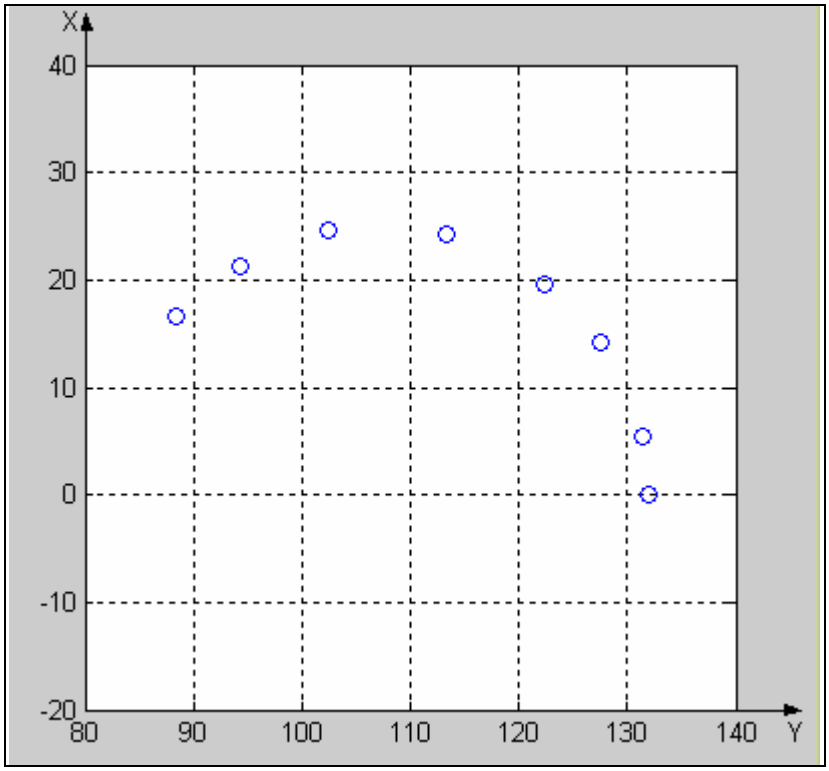


Figure 4.24 Representation of the eight final points into the plane XY

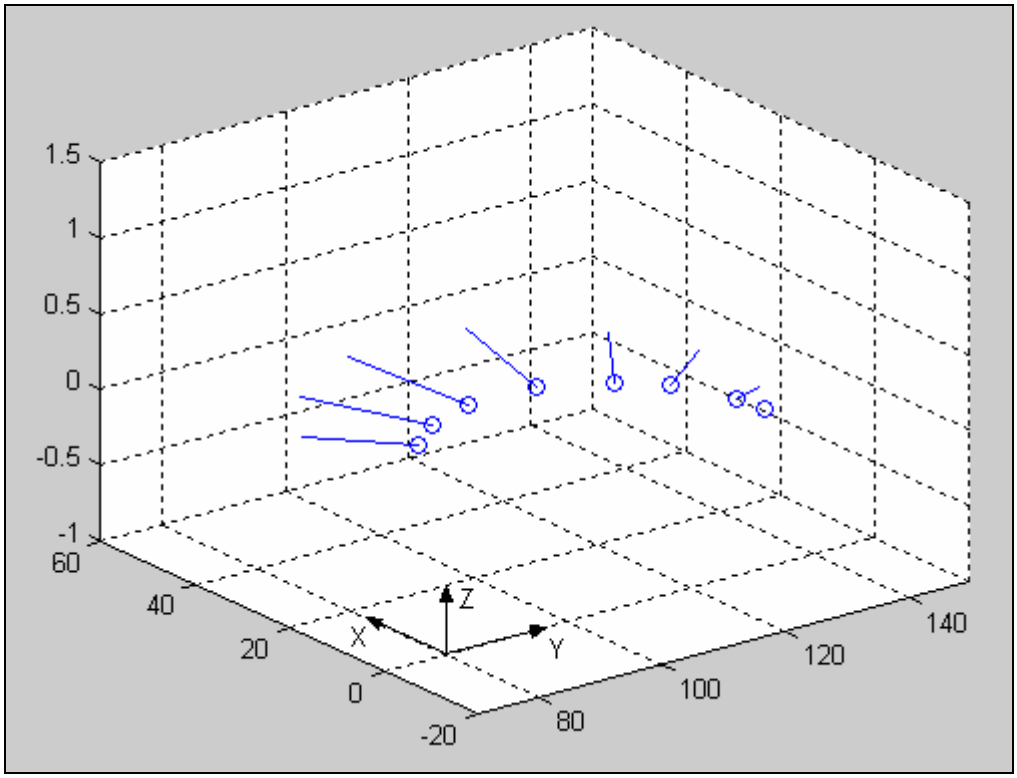


Figure 4.25 Representation of the eight final points and the normal vectors

In case of study, the robot path is a circumference (see fig. 4.26). This movement of the gun accompanies by the rotation piece allows to achieve the coating process of the piece.

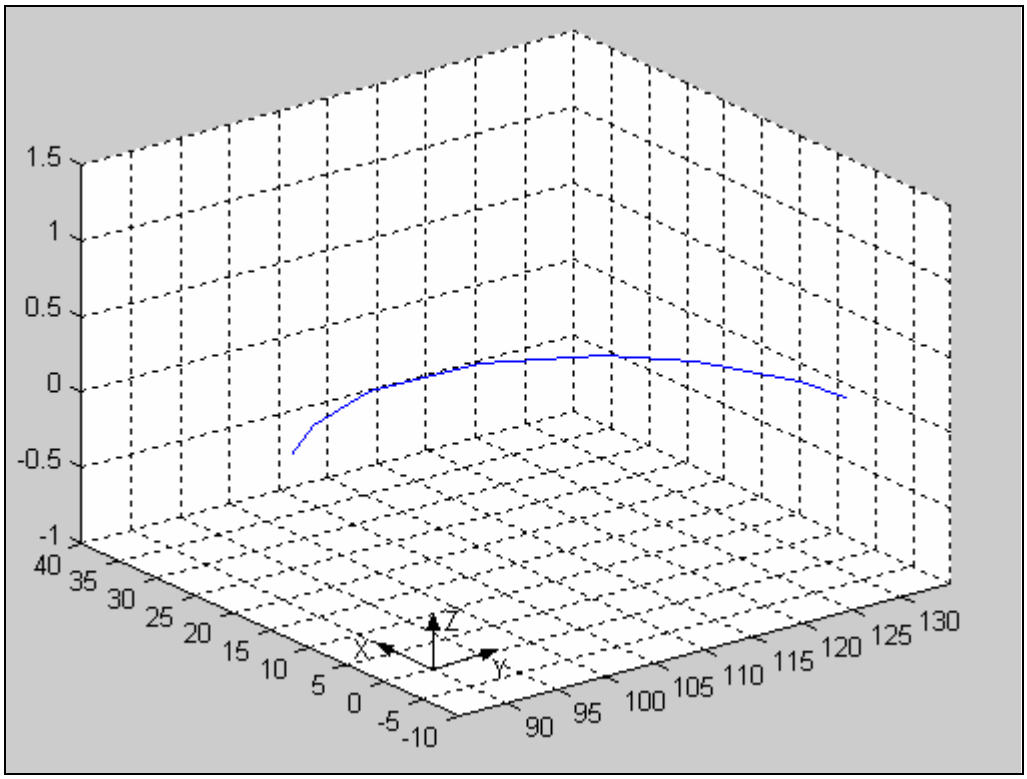


Figure 4.26 Simulation of the trajectory of the robot's torch

The generated path is developed in the code of the subroutine Kugelzapfen3. The robot speed, tool replacement and positions for ignitions are included in this subroutine (see annex 4).

CASE STUDY FOR THE GENERATION OF THE TRAJECTORY BY USING A CAD MODEL. KBA-BLADE

In case a CAD-model of a piece, which has to undergo a coating process is available, the model can be used to obtain interpolation points and normal vectors that are required for the definition of the robot path. In this case was used a 3D mechanical design software called “SolidWorks” as CAD-system. The image generates in the software design is presented in the figure 4.27.

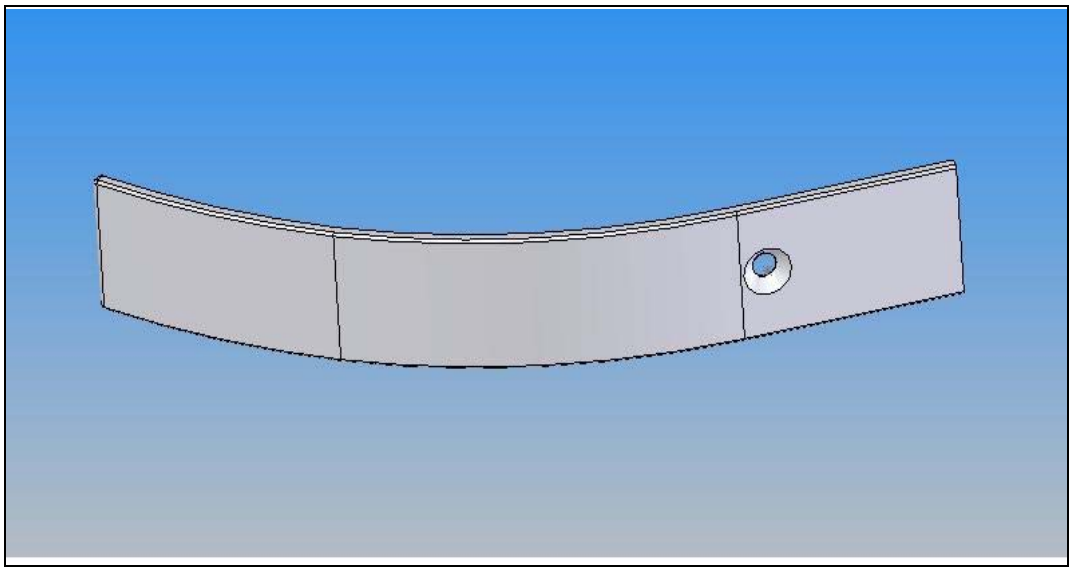


Figure 4.27 Design of the KBA piece in CAD program

The positioning of the pieces in the manufacturing cell is the starting point of the pieces disposition of the model. The disposition of the piece was already seen on the method by teach-in, two serial pieces, so the model reflects this disposition as well (see fig. 4.28).
This way the geometrical data are acquired starting from the real disposition of the KBA-blades for the coating process.

Starting from the assembly it is possible to generate the robot path. For this is necessary to make sections in the piece, the number of necessary sections is coincident with the lines that makes the robot during the coating process. In this work four sections have been defined with a typical value of off-set of 3 mm (see fig. 4.30).

The created surfaces are stored in the STL-format, the information of this surfaces is a mesh of triangular elements (see fig. 4.4). The STL-file contains the Cartesian coordinates of the corresponding vertex and the components of the normal vector for each triangle.

Bellow the adaptation of the registered points consists of the filtrate of double points and the collocation of the spatial points in the appropriated sequence. The robot path presents a unique way of coating, in the case of study exists symmetry in the trajectory every time the robot’s torch realizes the round trip movement, with the different of the off-set in the points coordinates (see fig. 4.34).

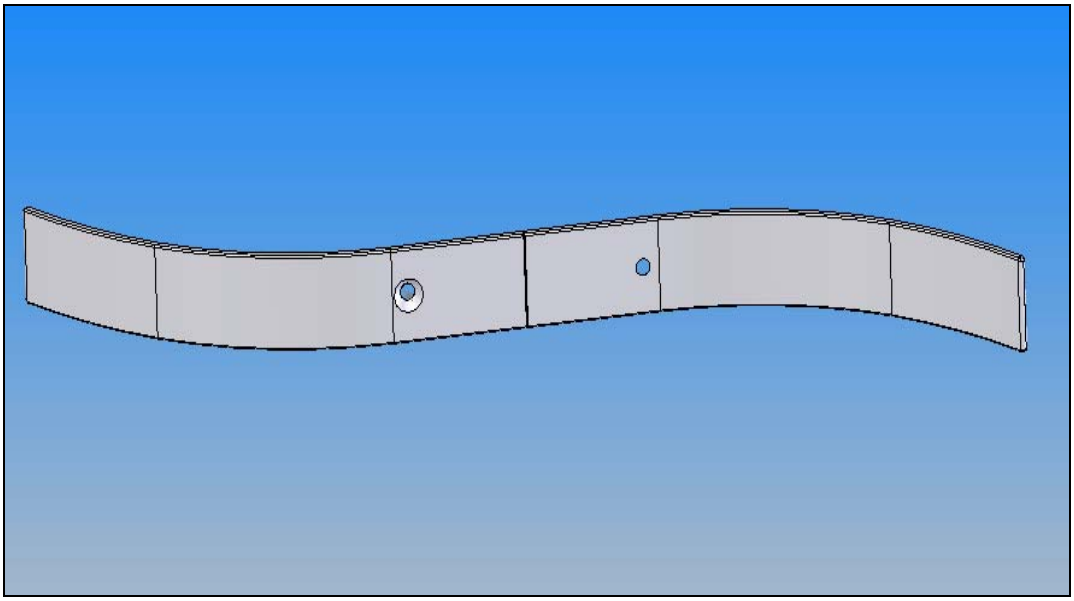


Figure 4.28 Simulation of assembling the KBA pieces in the manufacturing cell

The figure 4.29 shows some sections that have been made in the KBA-blades in order to find the robot trajectory. Initially have been made four sections on the pieces, the figure 4.30 shows an expansion of the areas sectioned, for better visualization.

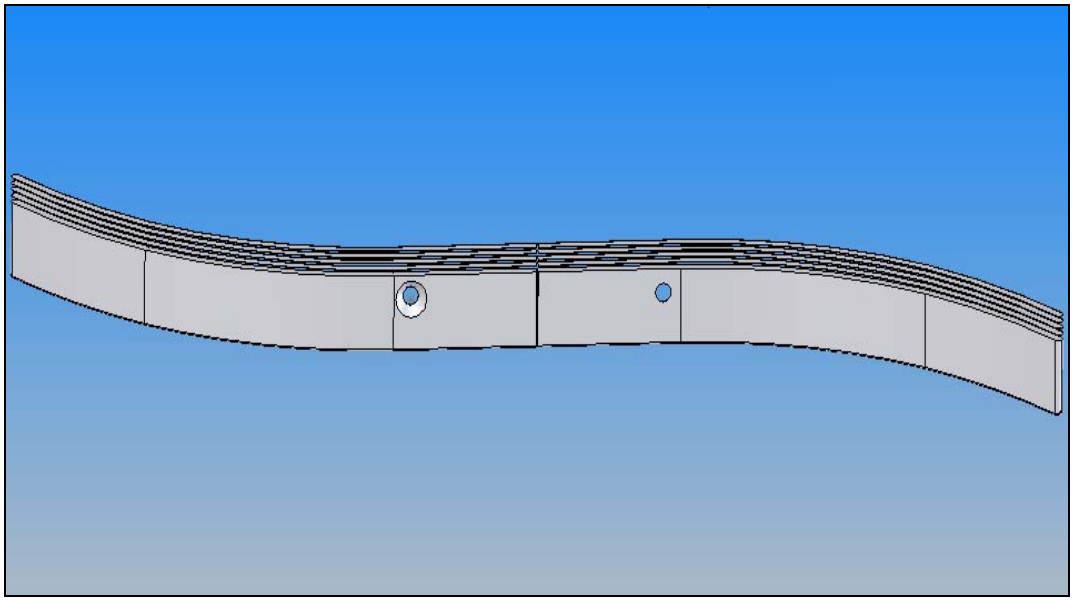


Figure 4.29 Different sections on the KBA pieces

In this case a first section is enough to define the robot trajectory, because the difference between sections is the off-set, since all the points present the same coordinate “x” and coordinate “y ".

CAD-system has been used taking into account a simple piece. If we were interested in working with a more complex piece we would focus on the propeller as example, which has already been studied by teach-in. The main problem it is found when working with the CAD-system is to generate the 3D model of the piece.

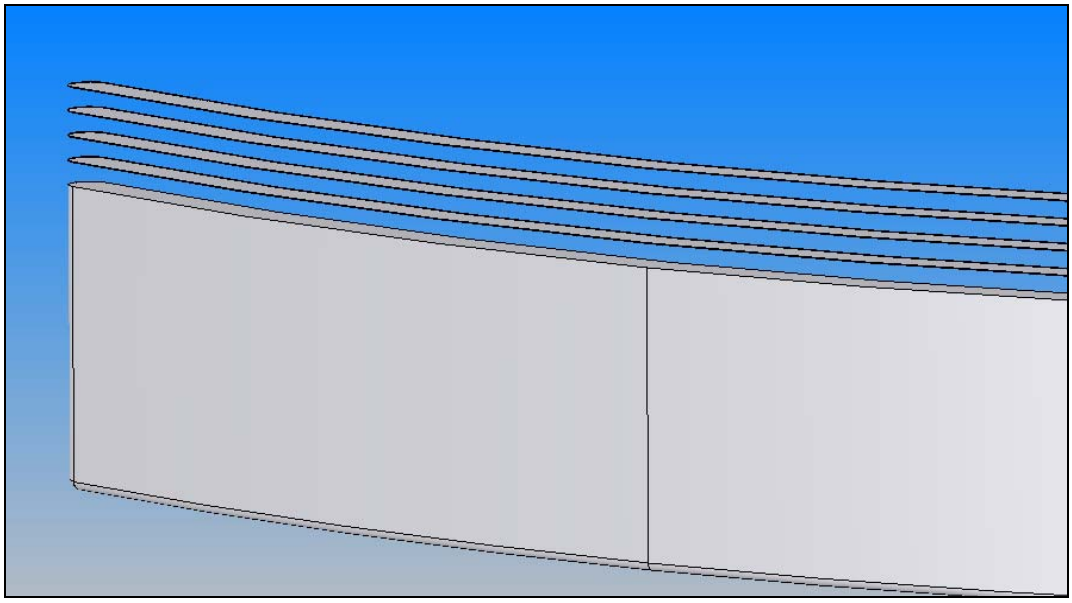


Figure 4.30 Different sections in the KBA piece

The surface highlighted in green presents a mesh by triangles, this surface has been saved in the STL-format, which has been previously mentioned.

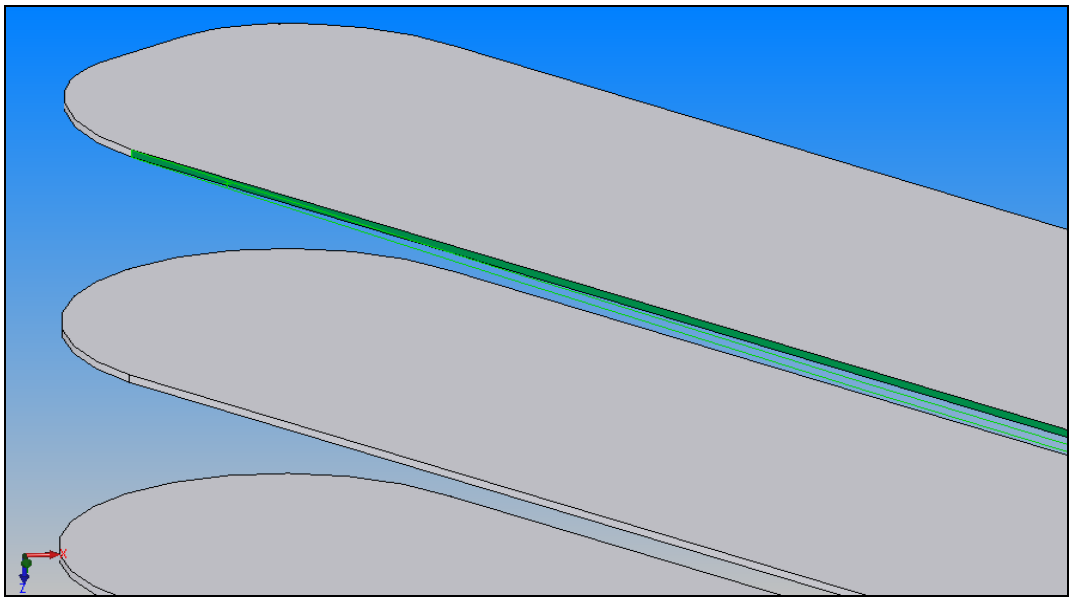


Figure 4.31 Surface with mesh, information stored in the STL-format

The figures 4.32 and 4.33 show a representation of the points obtained from sections performed in the pieces. So the figure 4.32 only shows points, which reflects an ideal view of the robot path. While in the figure 4.33 is shown the points represented in the previous figure and the vectors normal coupled, these vectors would be the direction that defines the orientation of the robot's torch in order to optimize the coating process.
The points coordinates are referred to the coordinate system of the design software, it is possible makes a change to the coordinate system of the robot with a translation movement and a rotation matrix.

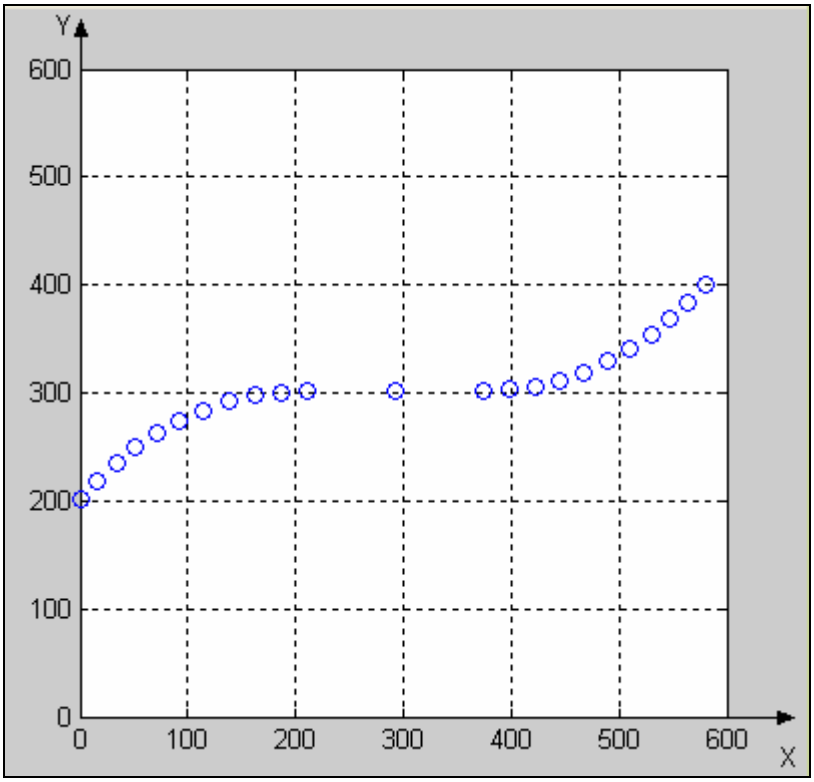


Figure 4.32 Representation of the points of a section in the XY plane

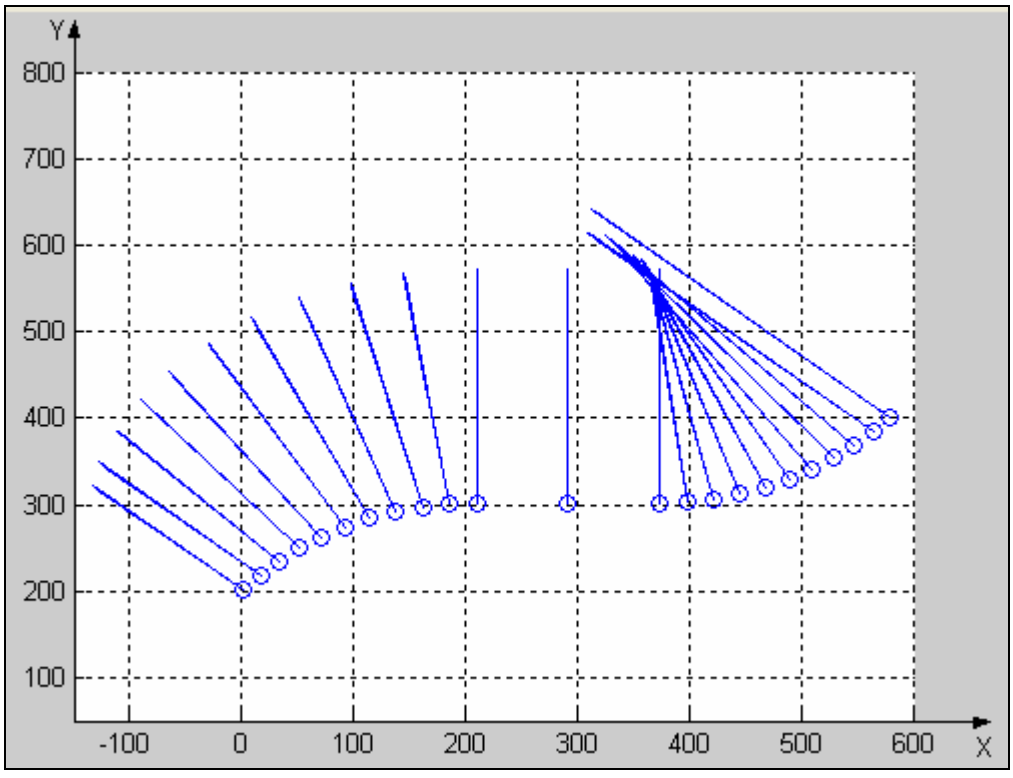


Figure 4.33 Representation of the points and the normal vectors in the XY plane

Finally in the figure 4.34 has been reflected the generated trajectory with the Matlab-subroutine 4.5 for the coating of KBA pieces (see annex 4).

In this case, the generated path is included in the code of the subroutine Kbauer3.v3 (see Annex 4). This subroutine presents light modifications to the original subroutine Kbauer3.v2, which was used to create the robot path of the KBA piece on the method of teach-in.

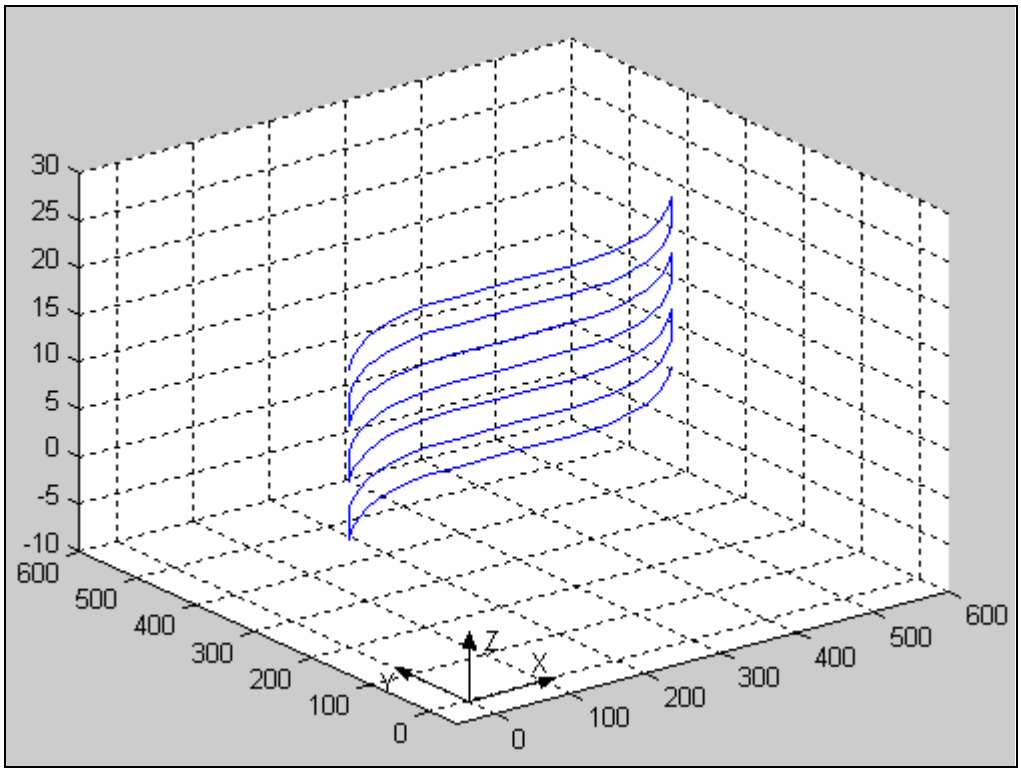


Figure 4.34 Simulation of the trajectory of the robot's torch in KBA pieces

5 Numerical simulation

5.1 Introduction

The numerical analysis of the heat transfer from the coating flame and the hot particles (hot spot) of the surface and inside a mechanical piece using as a ball pivot ("Kugelzapfen") during the coating process is the main point in this chapter. The finite element method software "ABAQUS 6.7" is used in this study to modelled, analysed, controlled and optimised this heat transfer phenomena.

The coating process is important to improve the mechanical and chemical properties of the. In order to get the aim ceramic or cermets additions are used in the coating process such as WC-Co, TiO_2, Mo,..

5.2 Numerical simulation of the temperature distribution during the coating process

Due to the working condition of the ball pivot (Kugelzapfen), the surface with contact suffers important wear. To reduce this problem, it is desirable to apply a coating process over its surface to improve its properties, such as wear, erosion or corrosion resistance, without changes in its mechanical properties [4.1.4.2.3 Design Aided Computer (CAD)].

Due to the impact of high flame temperature and hot particles, the piece suffers a significant thermal load during the coating process [2.1.2 High Velocity Oxigen-Fuel (HVOF)]. This can lead to deformation and structural changes inside the component and considerably affects the residual stress distribution on the coating system and the substrate. Therefore, it is necessary to optimize and control the thermal transfer during the coating process.

By means of numerical simulation with the method of the finite elements it is possible to calculate the temperature distribution on the workpiece during the coating process and thus to generate an input for a further calculation of deformation and residual stress distribution. For the production of the coatings the High Velocity Oxygen Fuel (HVOF) technology over the workpiece is used [Chapter 2. State of the art]. The parameters of the process can be found in the table A.5.1 of the annex.

As shown in the figure 5.1, the surface of contact is a small area onto the sphere of the piece. The coating is applied over this area (see fig. 4.20).

By means of the finite element method it is possible to simulate the effects of thermal loads caused by impinging gases (coating flame) and particles on the temperature distribution in the workpiece. Commercial softwares based on the FEM for the calculation of thermal and mechanical analysis are ANSYS, ABAQUS, FLUX3D®, NX NASTRAN and more. In this work the "ABAQUS 6.7" software was used.

Figure 5.1 Ball pivot under assembly conditions in the articulated joint

In this chapter the process to generate this model and the different steps to develop the simulation in the software, such as the definition of substrate material, heat flow, initials and boundary conditions or meshing of the model are described. These steps are represented as a flow chart in the figure 5.2.

The main steps of a numeric simulation are preprocessing, processing/solving and postprocessing.

- Preprocessing: this process consists of the design of the piece, it is possible to define two different ways in order to get it, by CAD or INPUT.
  By CAD, this first way is easier to work, it is as using a CAD-software. It is only necessary to draw the singular piece and to follow the different steps represented in the figure 5.2.
  By INPUT, this second way is more complex, in this process it is necessary to know the different commands that uses the software, such as (element, node, material, step, mesh, boundary conditions,…). This process is developed over a text editor.

- Processing / Solving: in this step the software calculates or solves the problem that has been defined in the previous step. Different problems can be solved such as heat transfer analysis, sequentially coupled thermal-stress analysis, fully coupled thermal-stress analysis, static stress analysis, coupled thermal-electrical analysis,...
  In this work the used procedure has been a coupled thermal-stress analysis.

- Postprocessing: it is possible to visualize the results of the problem, such as the temperature distribution, the stress field (Von Mises, Tresca,…), displacement in nodes, thermal gradient and more.

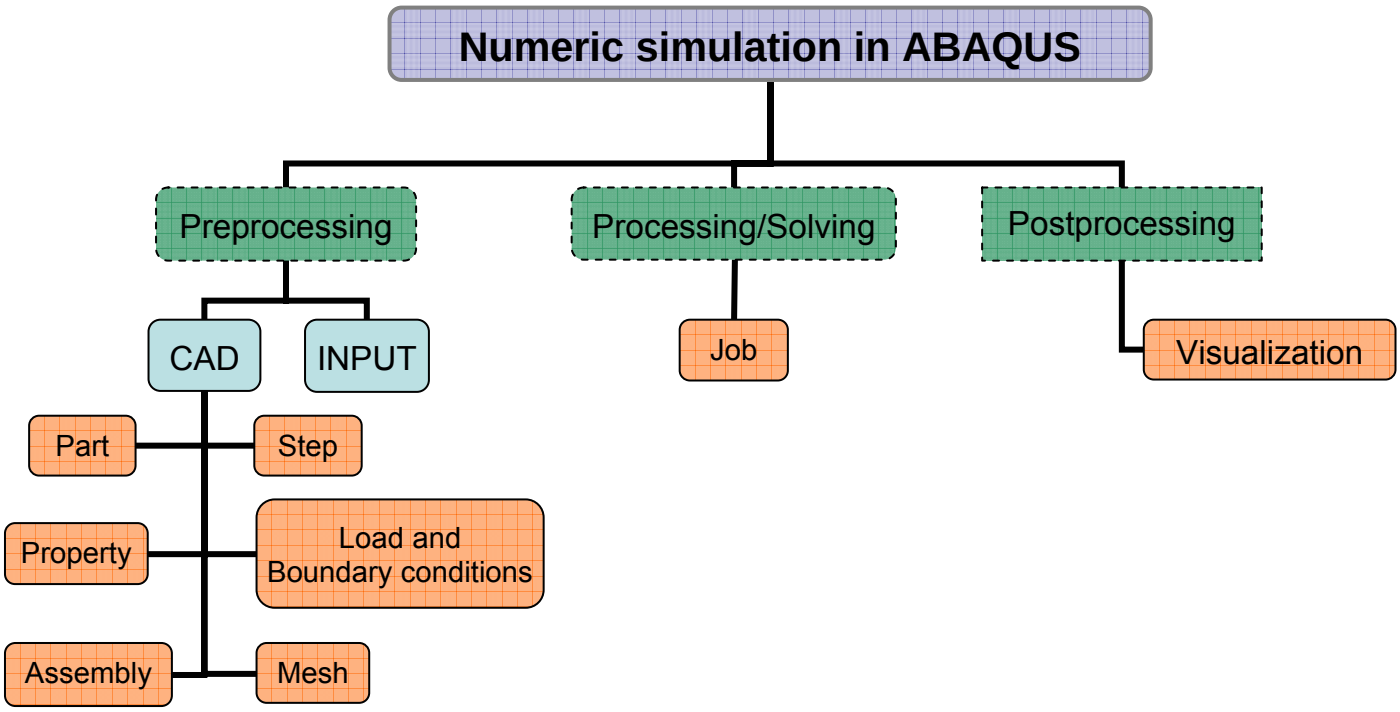


Figure 5.2 Flow chart for numeric simulation in the “ABAQUS 6.7” software

5.2.1 Preprocessing

The ABAQUS 6.7 software offers directly the possibility of designing the geometry of the workpiece and allows importing the geometrical data of a CAD-model in the following formats: ACIS, IGES, VDA, STEP, CATIA®, PARASOLID®, Pro/Engineer®, IDEAS®. The ball pivot was designed in the ABAQUS software.

The simulation process has a series of steps; it begins with the model generation, characteristics of the material, boundary conditions and loads, steps selection, meshing the model, solving and results (see fig 5.2). The last step will be presented in the following chapter (Results of simulation). This series of steps are developed.

5.2.1.1 CAD model

The geometrical data were obtained by measuring the ball pivot (Kugelzapfen). On basis of these measurements, models of the workpiece were built by using ABAQUS and SolidWorks softwares. These models were respectively used for the numerical simulation and for the calculation of the robot trajectory for the coating process [4.1.4.2.3 Design Aided Computer (CAD)].

As shown in the figure 5.3, in order to obtain a model of the workpiece for the numerical simulation, a section of this piece was drawn on basis of the geometrical data in the ABAQUS software. This designed section was rotate 360º over the symmetry “Y-axis” and therefore the model was created.

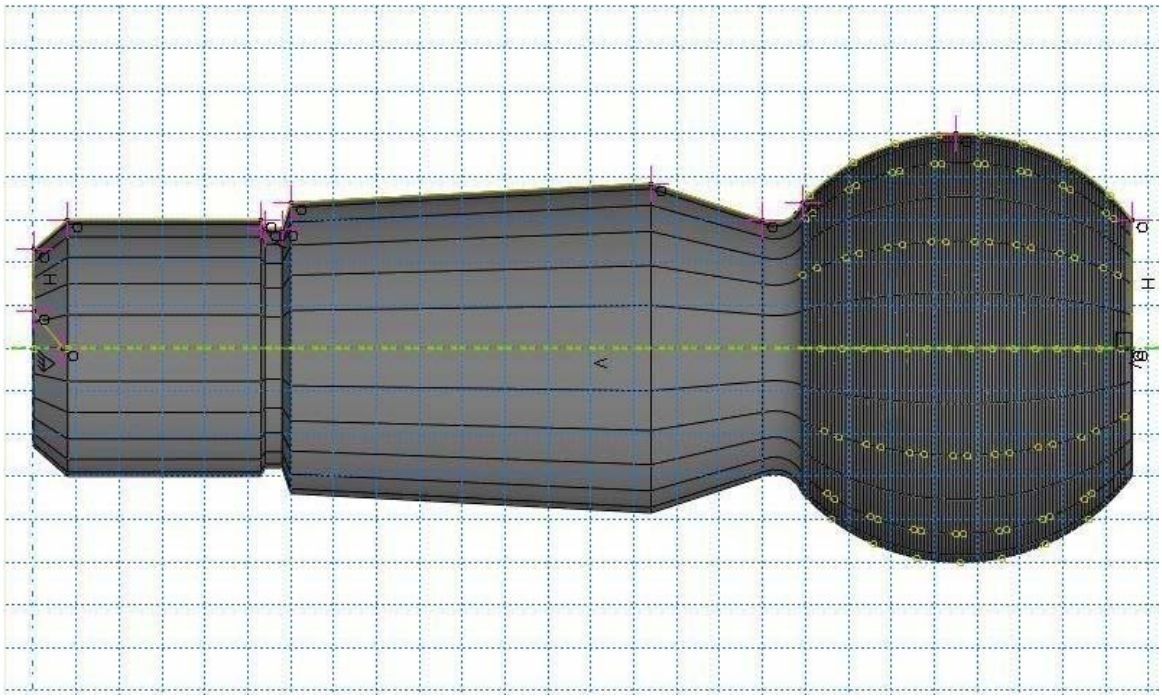


Figure 5.3 Model generation in the “ABAQUS 6.7” software

The figure 5.4 shows the model created in the ABAQUS software, this model is the result of the process of rotation. The different sections are necessary to optimize the mesh and also to obtain the referent nodes, which can be used by the calculation of the areas where the thermal loads are applied.

By importing directly the model of the piece from the design software to the numeric simulation program, it would have not been necessary to model the piece in the ABAQUS software and the process would have been simplified.

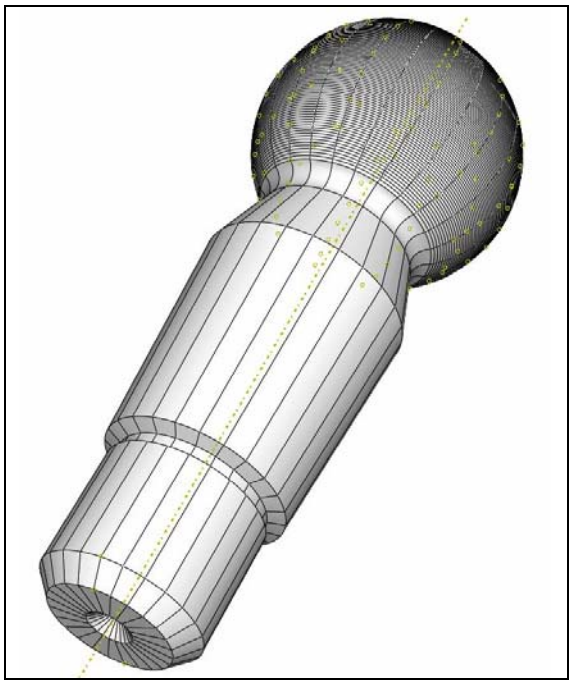


Figure 5.4 Model of the workpiece in the “ABAQUS 6.7” software

5.2.1.2 Characterization of the material for the substrate and coating material

The substrate is made of the material 42CrMoS4, which is a steel (DIN EN 10083) and the coating material is WC-Co in powder form. To implement the model it is necessary to specify the properties of the materials that compose it, such as density, Young's module, thermal expansion coefficient, thermal conductivity and specific heat. These properties are function of the temperature, so tables of properties for different temperatures are introduced in the ABAQUS software and the program obtains through linear interpolation the range of values.

Substrate: Steel 42CrMoS4

Density

Density (kg/m3)	Temperature (ºC)
7780	0
7750	100
7720	200
7700	300
7680	400
7630	500
7580	600
7540	700

Table 5.1 Density of the material, steel 42CrMoS4 [33]

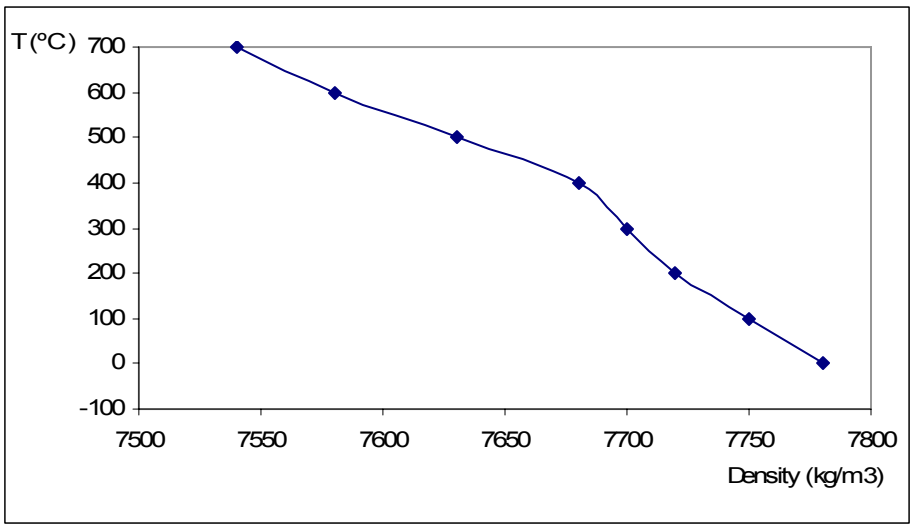


Figure 5.5 Evolution of the density with the temperature in the steel 42CrMoS4

Module's Young

Module´s Young (Pa)	Temperature (ºC)
220 E9	0
210 E9	100
200 E9	200
190 E9	300
180 E9	400
175 E9	500
165 E9	600

Table 5.2 Module's Young of the material, steel 42CrMoS4 [33]

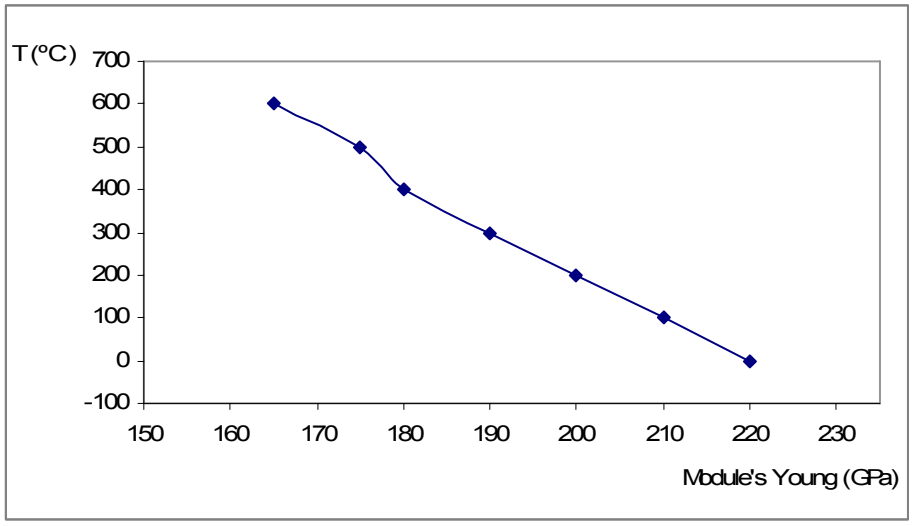


Figure 5.6 Evolution of the module's Young with the temperature in the steel 42CrMoS4

Thermal conductivity

Thermal conductivity (W/m K)	Temperature (ºC)
55	0
55	100
54	200
41	300
39	400
36	500
34	600
29	700
29	800
29	900
29	1000

Table 5.3 Thermal conductivity of the material, steel 42CrMoS4 [33]

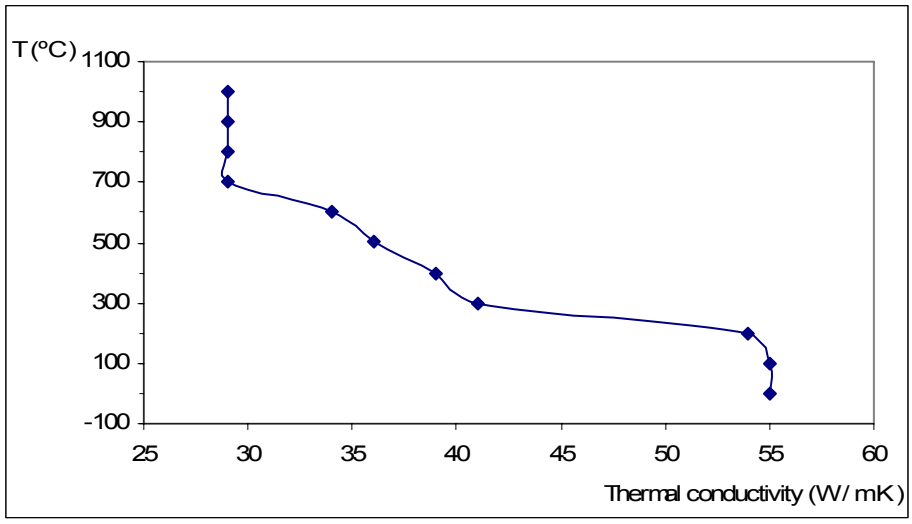


Figure 5.7 Evolution of thermal conductivity with the temperature in the steel 42CrMoS4

The thermal conductivity diminishes with the increase of the temperature.

Thermal expansion and specific heat

Thermal expansion ($ºC^{-1}$)	Temperature (ºC)
10,50 E-6	-100
11,50 E-6	0
12,00 E-6	100
12,50 E-6	200
13,00 E-6	300
13,30 E-6	400
13,75 E-6	500
14,25 E-6	600
14,80 E-6	700

Specific Heat (J/kg K)	Temperature (ºC)
420	0
460	100
490	200
520	300
540	400
560	500
580	600
620	700

Table 5.4 Thermal expansion and specific heat of the material, steel 42CrMoS4 [33]

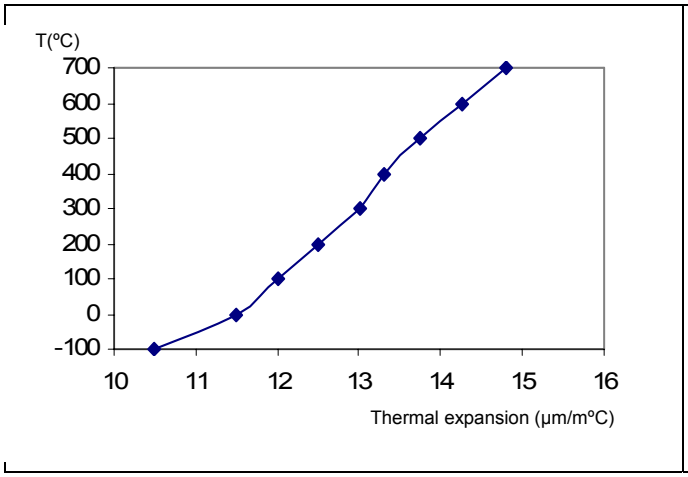


Figure 5.8 Evolution of the thermal expansion with the temperature in the steel

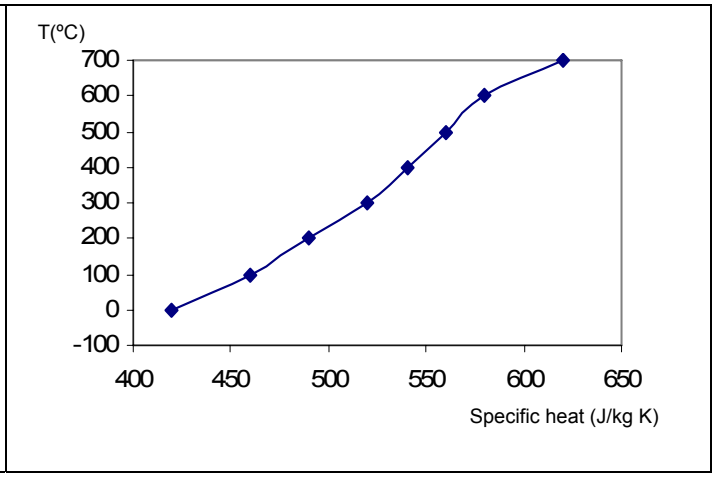


Figure 5.9 Evolution of the specific heat with the temperature in the steel

Finally in the following lines appears the part of the code that reproduces this information and that will be used by the ABAQUS software.

```
*SOLID SECTION, ELSET=_PICKEDSET2, MATERIAL=STEEL-42CRMOS4
1.,
*MATERIAL, NAME=STEEL-42CRMOS4
*CONDUCTIVITY
55.,   0.
55., 100.
54., 200.
41., 300.
39., 400.
36., 500.
34., 600.
29., 700.
29., 800.
29., 900.
29.,1000.
*DENSITY
7780.,   0.
7750., 100.
7720., 200.
7700., 300.
7680., 400.
7630., 500.
7580., 600.
7540., 700.
*ELASTIC
2.20e+11, 0.3,   0.
2.10e+11, 0.3, 100.
2.00e+11, 0.3, 200.
1.90e+11, 0.3, 300.
1.80e+11, 0.3, 400.
1.75e+11, 0.3, 500.
1.65e+11, 0.3, 600.
*EXPANSION
1.050e-05,-100.
1.150e-05,   0.
1.200e-05, 100.
1.250e-05, 200.
1.300e-05, 300.
1.330e-05, 400.
1.375e-05, 500.
1.425e-05, 600.
1.480e-05, 700.
*SPECIFIC HEAT
420.,   0.
460., 100.
490., 200.
520., 300.
540., 400.
560., 500.
580., 600.
620., 700.
```

Coating material: Powder WC-Co

The properties of the Tungsten carbide and Cobalt are presented in the table 5.5, also a picture by microscope of this coating material is shown in the figure 5.10.

Powder	**WC-Co / 83-17 /**
Thermal expansion (m/m K)	5,4 E-6
Young's Module (Pa)	580 E9
Density (kg/m^3)	14500
Thermal conductivity(W/m K)	110
Specific heat (J/kg K)	480

Table 5.5 Properties of coating material, powder WC-Co [36, 37]

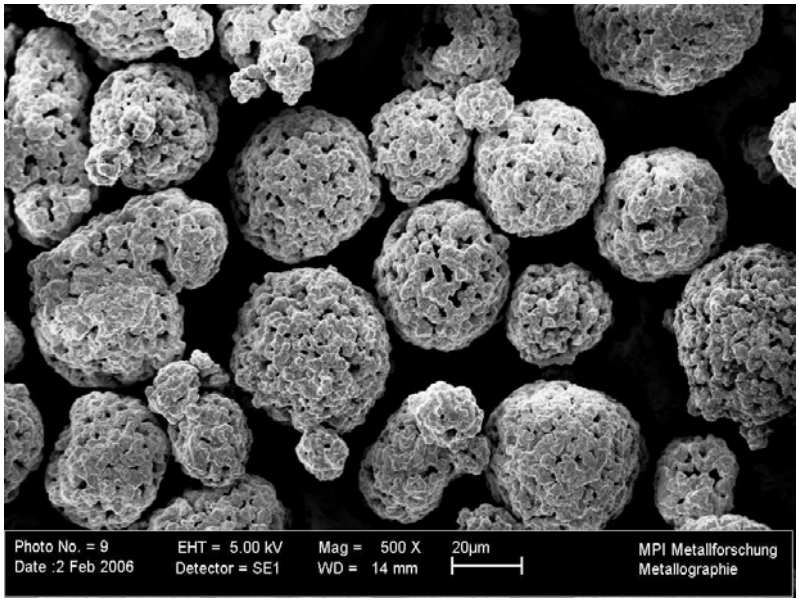


Figure 5.10 Coating material seen by microscope

Finally in the following lines appears the part of the code that reproduces this information and that will be used by the ABAQUS software.

```
*MATERIAL, NAME=WC-Co
*CONDUCTIVITY
110.,
*DENSITY
14500.,
*ELASTIC
5.8e+11, 0.3
*EXPANSION
5.4e-06,
*SPECIFIC HEAT
480.,
```

5.2.1.3 Mesh of the piece

The type of element uses in the mesh of the model is the C3D8T, an 8-node thermally coupled brick with trilinear displacement and temperature.

The mesh can be manual or automatic, in this model the mesh has been generated by the automatic way, the total number of nodes that compound the model is 19399 and the number of elements is 14304.
The figure 5.11 shows the mesh of the model of the workpiece.

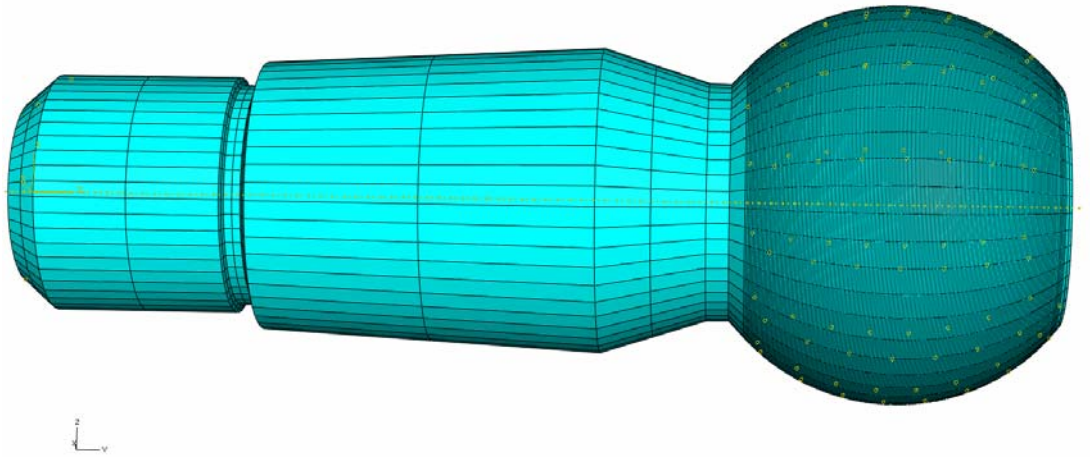


Figure 5.11 Mesh of the model

5.2.2 Model descriptions

The modelling of this study consist to describe the initials conditions, boundary conditions, the thermal loads or the flow model, field output and history output of the variables.

5.2.2.1 Initial conditions

The initial conditions at the nodes are the value of atmosphere temperature, 20 ºC, (see fig. 5.12).

```
** PREDEFINED FIELDS
**
** Name: Field-1   Type: Temperature
**
* INITIAL CONDITIONS, TYPE=TEMPERATURE
_PICKEDSET23, 20.0
```

During the coating process the piece has a movement of rotation (150 rev/min), while the torch of the robot has a translation movement. In the simulation process the movements are realized by the torch and the piece is fixed (see fig. 5.13), the fixation of the piece is introduced as a boundary conditions, which is represented by the following program code:

```
** BOUNDARY CONDITIONS
**
** Name: Disp-BC-1 Type: Symmetry/Antisymmetry/Encastre
**
*BOUNDARY
_PICKEDSET158, ENCASTRE
```

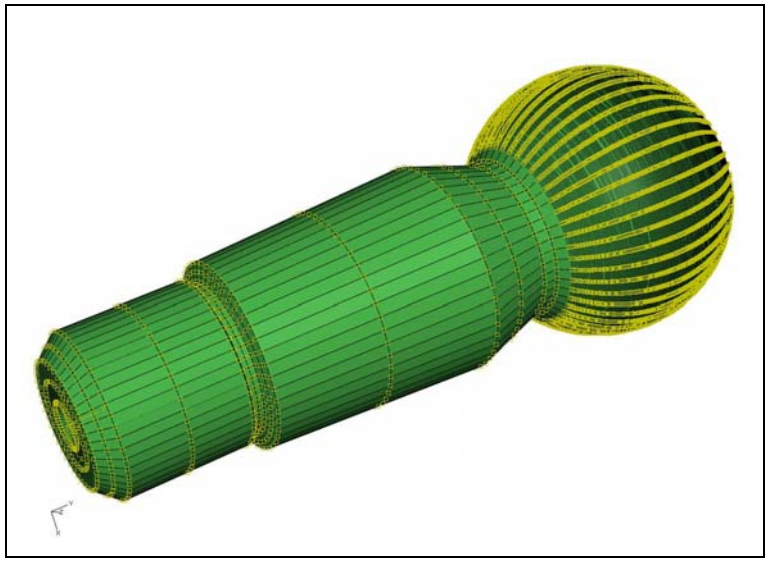


Figure 5.12 Initials conditions for the ball pivot in the ABAQUS software

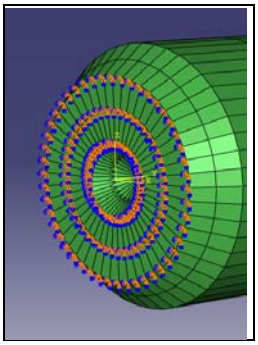


Figure 5.13 Boundary conditions for ball pivot, DOF=0

5.2.2.2 Heat transfer

During the coating process, the thermal energy is transferred by the flame and the impacting particles to the workpiece. Two main processes are present during the coating of the piece: heat transfer by convection (due to the impinging gases) and heat transfer by conduction (because of the totally or partially molted particles). Also a contribution of heat transfer by radiation could be considered, but the emissivity in the flame is very low in comparison with the emissivity in the substrate and the energy transferred through radiation can be compared as the reradiation from particle, and thus neglected.

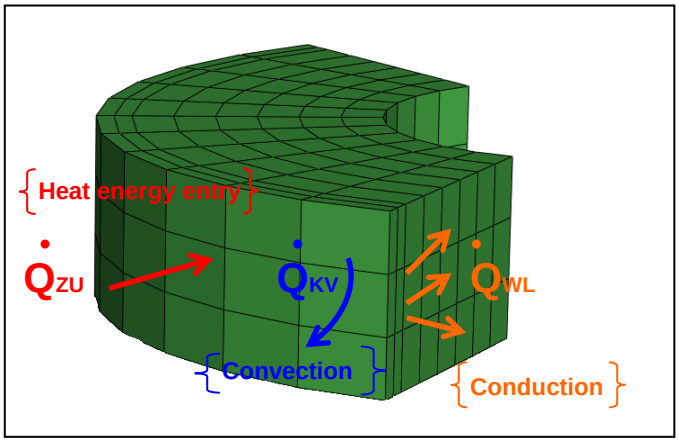


Figure 5.14 :
Energy flow scheme during the coating process.

Onto the surface because of conduction and convection.

Process of convection between the piece and the flame

The heat transfer by convection is due to the contribution of the impacting flame, which is composed by the combustion of gases and it is considered in the numerical simulation as a fluid. The flame has at the exit of the nozzle a temperature in the range 2500 – 3000 K depending on the process parameters. The temperature of the gases at the impact point is determinate by the spraying distance considered. [34].

The equation 5.1 defines the heat transfer by the convection process, which is function of the film coefficient "h", the area of incidence and finally the difference of temperature between the flame and the surface of the substrate.

$$\dot{Q}_{flame} = h \cdot \mathrm{A} \cdot (T_{flame} - T_{surface}) \qquad [5.1]$$

In the equation 5.1 is calculated the heat transfer in Watt, it is possible to calculate the heat transfer by other way, so shows in the equation 5.2, which offers the heat transfer in Watt pro quadric meter.

$$Q_{SC} = h \cdot (T_{flame} - T_{surface}) \qquad [5.2]$$

The calculation of the film coefficient is carried out in the equation 5.3, where the thermal properties of the combustion gases and the specific geometry of the problem are related.

$$h = (k_{gas})^{0.58} \cdot (Cp_{gas} \cdot \mu_{gas})^{0.42} \cdot (Nu/Pr^{0.42})/ L \qquad [5.3]$$

Where k_{gas} is thermal conductivity, Cp_{gas} is specific heat, μ_{gas} is dinamic viscosity and L is characteristic length [38, 39].
The relationship between the Nussel and Prant numbers is presented in the equation 5.4, which is function of the diameter of the gun nozzle of the HVOF deposition, the spraying distance between gun exit and substrate, Reynolds number and r describes the radial distance of interest from the gas impingement point on the substrate (see fig. 5.15).

$$Nu/Pr^{0.42} = (D/r) \cdot ((1-1.1 \cdot D/r) / (1+0.1 \cdot (H/D-6) \cdot D/r)) \cdot 2 \cdot ((1+0.005 \cdot Re^{0.55})^{0.5}) \cdot Re^{0.5} \qquad [5.4]$$

The rate r/D is the relative radial distance from the centre of the jet and H/D is the relative distance from the substrate to the nozzle exit.

Reynolds number is function of the gas exit velocity, density and dynamic viscosity of combustion gases and characteristic length, the calculation is presented in the equation 5.4.

$$Re = (d_{gas} \cdot v_{gas} \cdot L) / \mu_{gas} \qquad [5.5]$$

The range of validation for the equations is:

$$\begin{gathered} 2 < H/D \leq 12 \\ 2.5 < r/D \leq 7.5 \\ 2000 \leq Re \leq 400000 \end{gathered} \qquad [5.6]$$

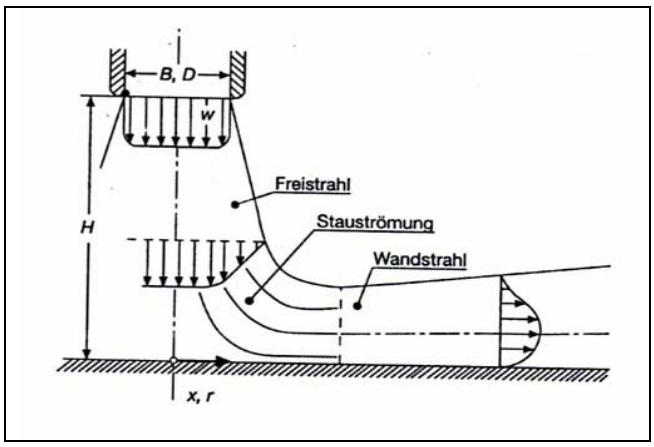


Figure 5.15 Flow through a circular nozzle [39]

With the data reflected in the table A.5.1 of annexes, it is possible to calculate the value of heat transfer by convection of the flame at the surface of the substrate, as shown the table 5.6.

Spraying distance (mm)	160	200
Gas velocity (m/s)	600	500
Particle temperature (K)	1600	1150
Reynolds number	2.5116e+005	2.0930e+005
$Nu/Pr^{0.42}$	335.4189	293.9188
Film coefficient (J/sKm^2)	629.6612	551.7555
T_{flame} (K)	1400	1150
$T_{surface}$ (K)	293	293
Q_{flame} (MW/m^2)	0.8230	0.4729

Table 5.6 Values range of parameters for convections [34]

The heat transfer by convection is between 5-10% of the total heat transfer at the surface of the substrate.

In the simulation the resulting values of the temperature of the flame and the film coefficient are 1150 K (850 ºC) [34] and 552 J/sKm2 respectively, these values correspond to the spraying distance of 200 mm.

In the following lines appears the part of the code that reproduces this information and that will be used by the ABAQUS software.

```
*SFILM
_PICKEDSURF10, F, 850., 552.
```

Process of conduction between the piece and particles

The heat transfer by the conduction process is represented in the equation 5.7, which is function of the quantity of injected particles or powder, the specific heat and the difference of temperatures between particles and the surface of the substrate.

$$\dot{Q}_{particle} = \dot{m}_{particle} \cdot Cp_{particle} (T_{particle} - T_{surface}) \qquad [5.7]$$

The maximum heat transfer is obtained for the value of deposition rate of 9 kg/h [35], considering an increment of temperature of 1000 K (particles ~ 5 µm and spraying distance of 200 mm) [34] and taking in account that the specific heat is 480 J/kgK, making use of the equation 5.7, the value obtained is 1208 W.

In the simulation program the loads are applied over a square cell, which present an area of 100 mm^2 approximately, so the maximum value of heat flux is 12 MW/m^2.

Spraying distance (mm)	160	200
Particle temperature (K)	1600	1300
Maximum deposition rate (kg/h)	9	9
Specific heat particle (J/kg K)	480	480
$\dot{Q}_{\text{particle}}$ (W)	1568,4	1208,4
Q_{particle} (MW/m^2)	15,7	12,0

Table 5.7 Values range of parameters for conductions [34, 35]

The table 5.7 shows for two spraying distances the values of the particle temperature, maximum deposition rate, specific heat and the resulting value of heat transfer by the conduction process.

For the industrial application of the coating of the ball pivot (Kugelzapfen), the used powder rate is of 100 g/min (6 kg/h) and with an increment of temperature of 1000 K, the resulting value of the heat transfer by the conduction process is 8 MW/m2.

In the following lines the code used in the ABAQUS software are reproduced:

```
** Name: SURFFLUX-1   Type: Surface heat flux
*DSFLUX
_PICKEDSURF10, S, 8e+06
```

Application of the loads in the surface of the piece

The thermal loads are applied in the surface of the ball pivot in the initial model, the application of the loads is previous to the mesh of the model.

During the coating process the piece presents a movement of rotation, while the robot's torch has a movement of translation. The simulation of both movements is represented by a heat source, and the resulting movement is a spiral (see fig 5.16).

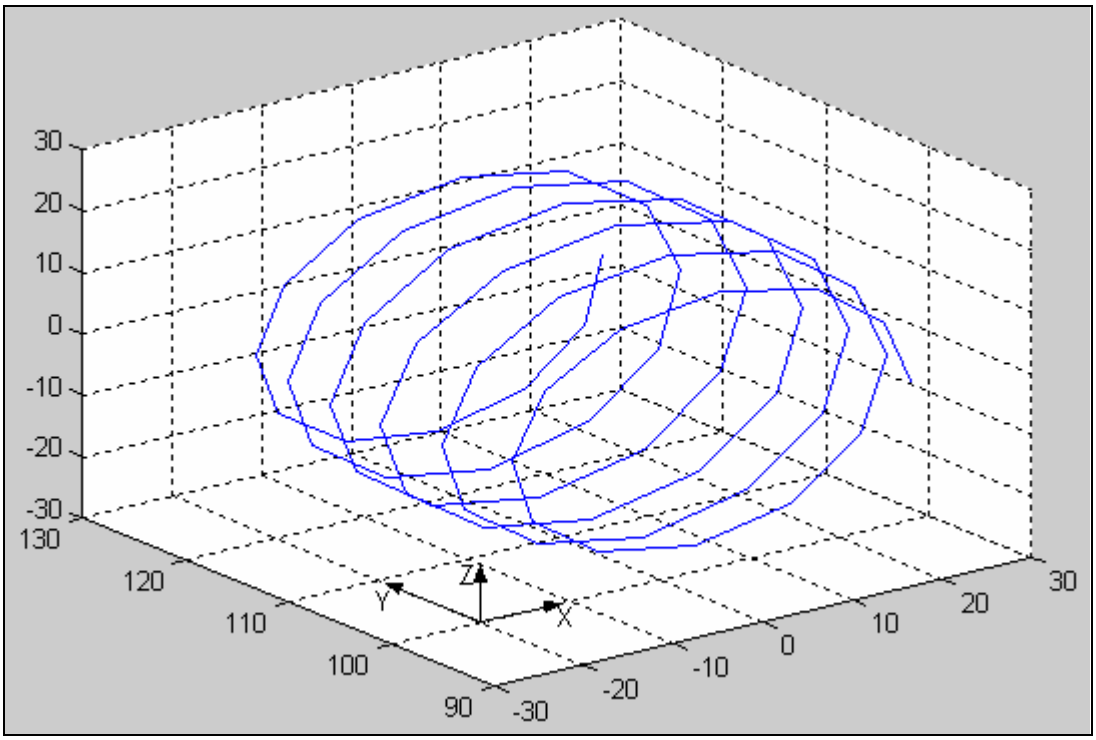


Figure 5.16 Representation of the movement of the heat source

The table 5.8 shows the parameters of the movement used in the calculation for the duration of the thermal loads in the steps, with the values of velocity and offset a period of 0,4 seconds is obtained by cycle. Each ring is compound by 12 nodes, each 30º exists a node (see fig. 5.17).

Velocity of torch (mm/s)	12,5
Rotation of the piece (rev/min)	150
Off-set (mm)	5
T_{cicle} (s)	0,4
T_{step} (s)	0,034

Table 5.8 Parameters of movement and time of load

The figure 5.18 shows the thermal loads applied in a ring, the sphere of piece has been divided in six rings and every ring presents thermal loads. The simulation process has the duration of 20 seconds.

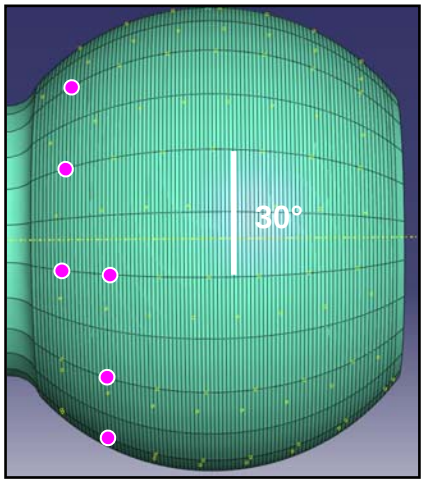


Figure 5.17 Representation of the nodes in the first ring

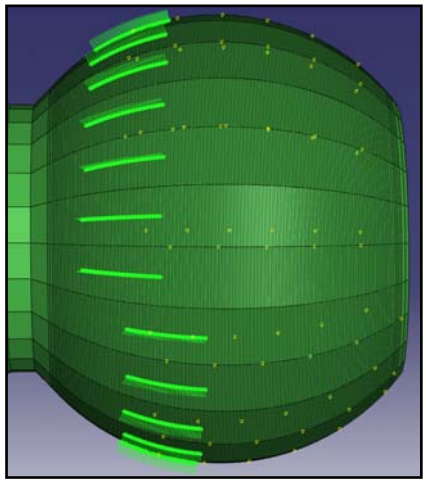


Figure 5.18 Ring with thermal loads

5.2.2.3 Steps selection

Before doing the processing, it is necessary to define the calculation procedures and the applied loads in the piece. After the processing the volume of data is very huge, for this reason it is necessary to select the data that must be stored.

For the data storage, it is possible to distinguish:

- Field output, the list with instantaneous values of the variables (temperature, stress, heat flux, displacement, thermal gradient, …)
- History output, the evolution of the variables

In this work the applied procedure is a couple thermal-stress analysis. In this method the displacement and heat transfer equations are simultaneously solved. Also the displacement values are considered in the thermal calculus.
The thermal loads had been previously presented in the paragraph 5.2.2.2, although are defined in the steps selection.

Finally in the following lines appears the part of the code that reproduces this information and that will be used by the ABAQUS software.

```
** STEP: Step-1
**
*STEP, NAME=STEP-1
*COUPLED TEMPERATURE-DISPLACEMENT, DELTMX=300.
0.0334, 0.0334, 3.34e-07, 0.0334
**
** LOADS
**
** Name: SURFFLUX-1   Type: Surface heat flux
*DSFLUX
_PICKEDSURF10, S, 8e+06
**
** INTERACTIONS
**
** Interaction: SURFFILM-1
*SFILM
_PICKEDSURF10, F, 850., 552.
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*OUTPUT, FIELD, VARIABLE=PRESELECT
**
** HISTORY OUTPUT: H-Output-1
**
*OUTPUT, HISTORY
*ELEMENT OUTPUT, ELSET=ELEMENTS
S, MISES
**
** HISTORY OUTPUT: H-Output-2
**
*NODE OUTPUT, NSET=NODES
NT,
**
*END STEP
```

6 Results of simulation

6.1 Thermal analysis

The part of this study was consisting to model and analysed the temperature and stress distribution in the surface and inside the substrate during the coating process. The aim of the coating process is to improve the mechanical and physical properties of the substrate after the coating process by taking care of the effects of heat transfer which can cause undesired effects such as microcracks inside the substrate. The results of this analyse is very important to interpreted and too understand the effect of the heat loads of the piece during the coating process. The evolution of nodal temperature can help us to understand the change in the piece associated to the coating process. This change has a strong relation with the generated stress and the coating mechanical properties. In this paragraph, the temperature evolution, the instantaneous temperature field and heat flux evolution are studied and presented.

Figure 6.1 shows the temperature evolution in three nodes on the surface of the piece. The three selected nodes are located at the beginning, half and final part of the surface to coat (see fig. 6.2).

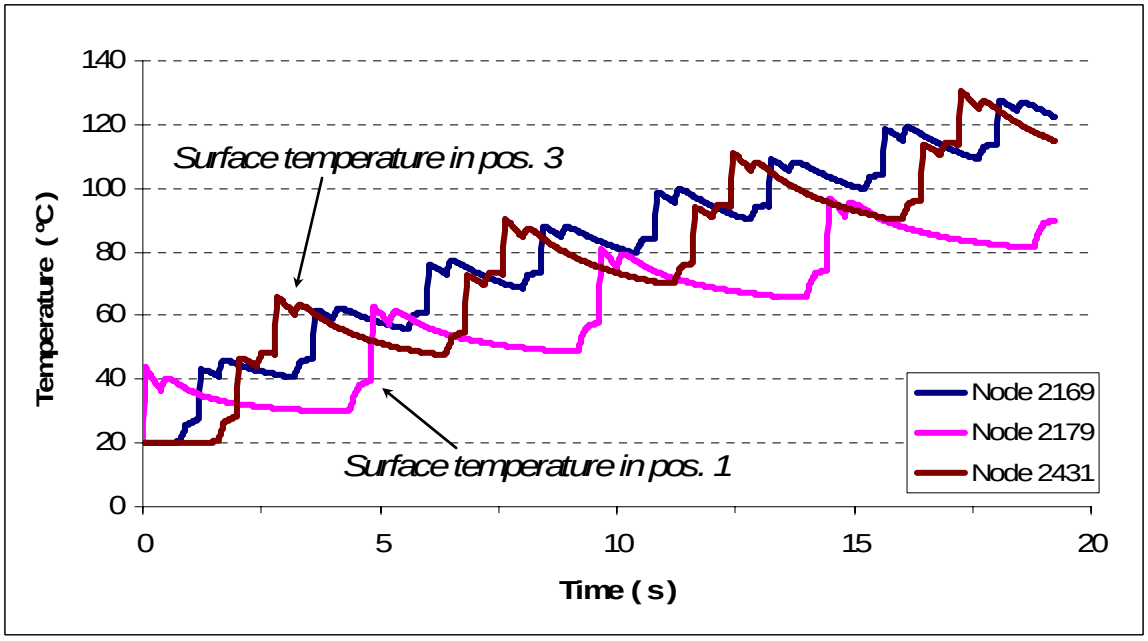


Figure 6.1 Evolution of nodal temperature during the simulation of the coating process

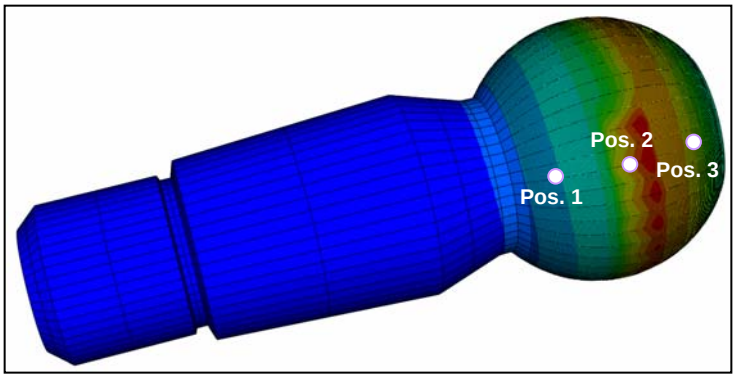


Figure 6.2 Positions of selected nodes

The time of the simulation in the ABAQUS software is 20 seconds, which are represented by four cycles. During a cycle the jet of the heat source is moving for all the points on the surface, drawing a spiral as trajectory (see fig. 5.16). In the figure 6.1, it is possible to distinguish each cycle for the first and third node. For the second node, eight peaks appear because of the jet past over this node two time per cycle.

Every cycle means an increment of temperature although the value of the increment is decreasing with the new cycle. For this reason, it is not possible to forecast the reached temperature during the application of several cycles without the help of the numerical simulation.

Next to the application of four cycles the maximum reached temperature is 145 ºC (at a different node of the selected on figure 6.1, see fig. 6.3).

The coating process of this workpiece has approximately the duration of three minutes and the measures of temperature by thermographic camera provide a maximum value of 220 ºC.

Figure 6.3 shows the instantaneous temperature field inside the substrate represented with a section of the piece. On this figure, it is possible to visualize the depth of maximum temperature inside of the material during a period of the simulation of the coating process. Other remark is the high temperature in the borders on the ball.

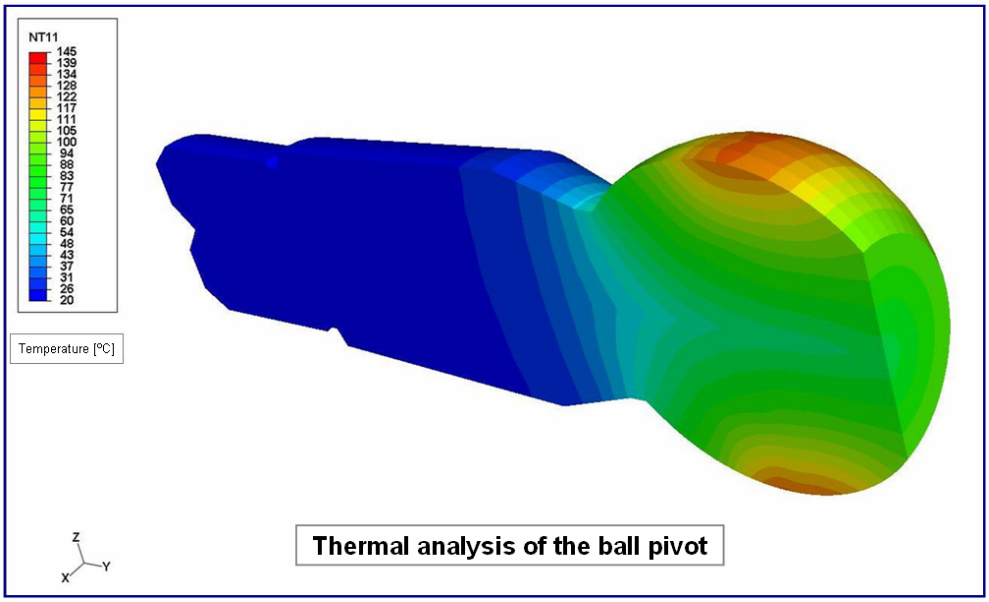


Figure 6.3 Field of temperature in a section of the piece

The heat flux is implemented in each step and come from to convection and conduction loads.

Figure 6.4 shows the evolution of heat flux in the nodes of the elements remarked on the figure 6.6. It is possible distinguish four cycles for the first and third node. It is important to note that for the same nodes all cycles have the same maximum heat flux.

For the second node, eight peaks appear because of the jet past over this node two times per cycle, as said above. The heat flux in the intermediate node has an approximate value to the half of the reached value in the other nodes.

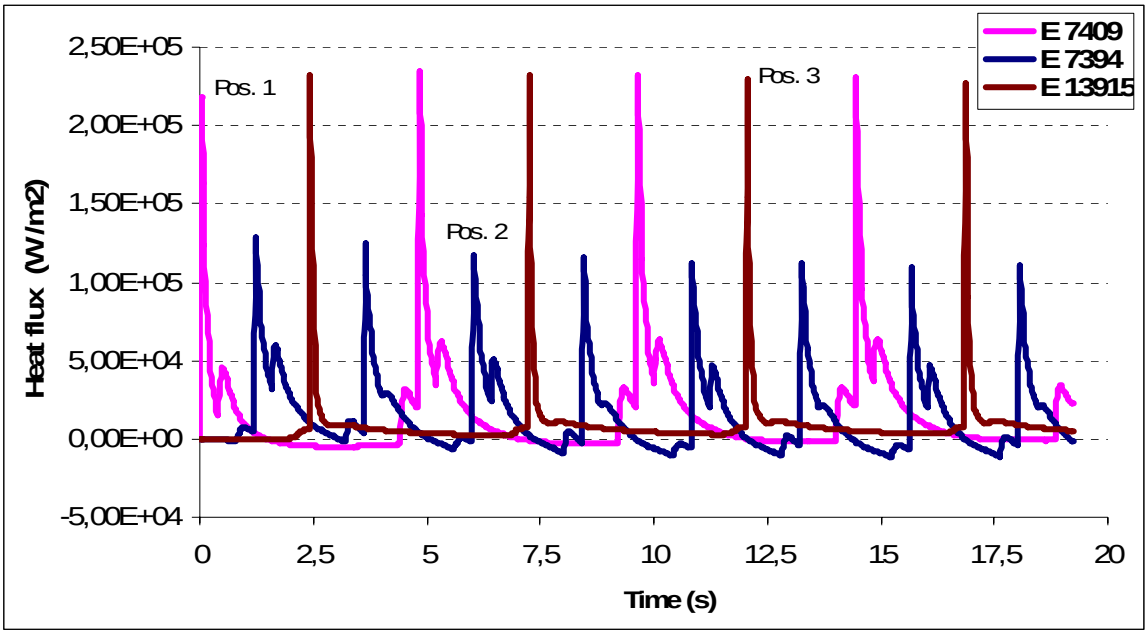


Figure 6.4 Evolution of nodal heat flux during the simulation of the coating process

In the figure 6.5 is shown the instantaneous field of heat flux inside the piece. In this figure, it is possible to visualize that the flux is mainly transported inward of the ball pivot.

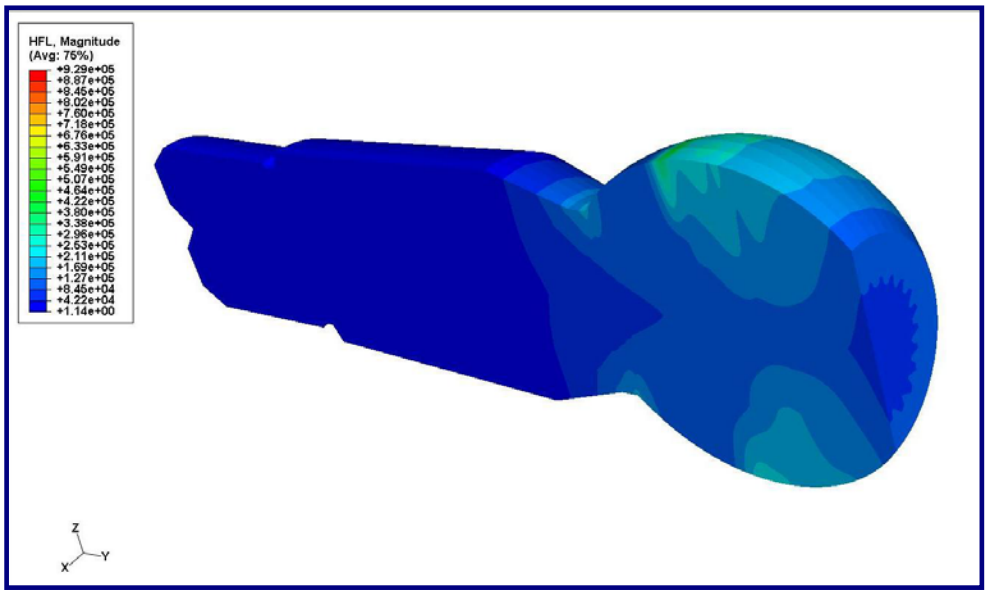


Figure 6.5 Field of heat flux in a section of the piece

The figure 6.6 shows the positions of the selected elements in the piece, these positions are located at the beginning, half and final part of the sphere.

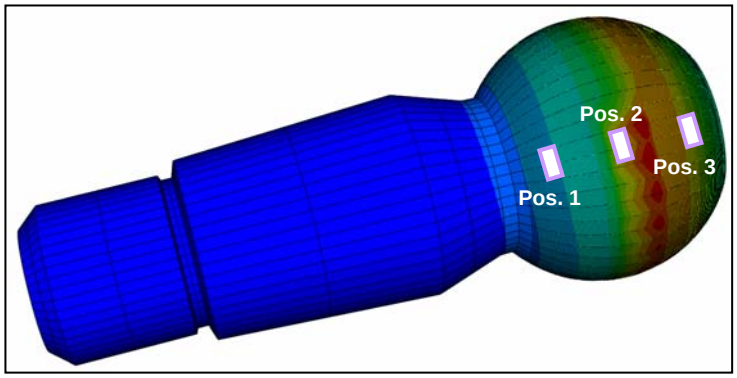


Figure 6.6 Positions of selected elements

6.2 Stress analysis

Once the field of temperature is known, it is possible to transform these data to obtain the field of stress. It is possible to choose various criterions to calculate the field of stress, in this case is used the criterion of “von Mises” stress because is the most suitable to use as yield criterion.

Figure 6.7 shows the evolution of the von Mises stress in the three nodes of the elements remarked in the figure 6.6. It is possible distinguish four cycles for the first and third node. It is important to note that for the third node, the third and fourth cycles have the same maximum value. In the case of the first node the maximum value tend to a horizontal asymptote. It is also necessary to highlight that the highest maximum value appears in the first node.

For the second node, eight peaks appear because of the jet past over this node two times per cycle, as said above. The same comment about the horizontal asymptote for the maximum value is applicable for this node.

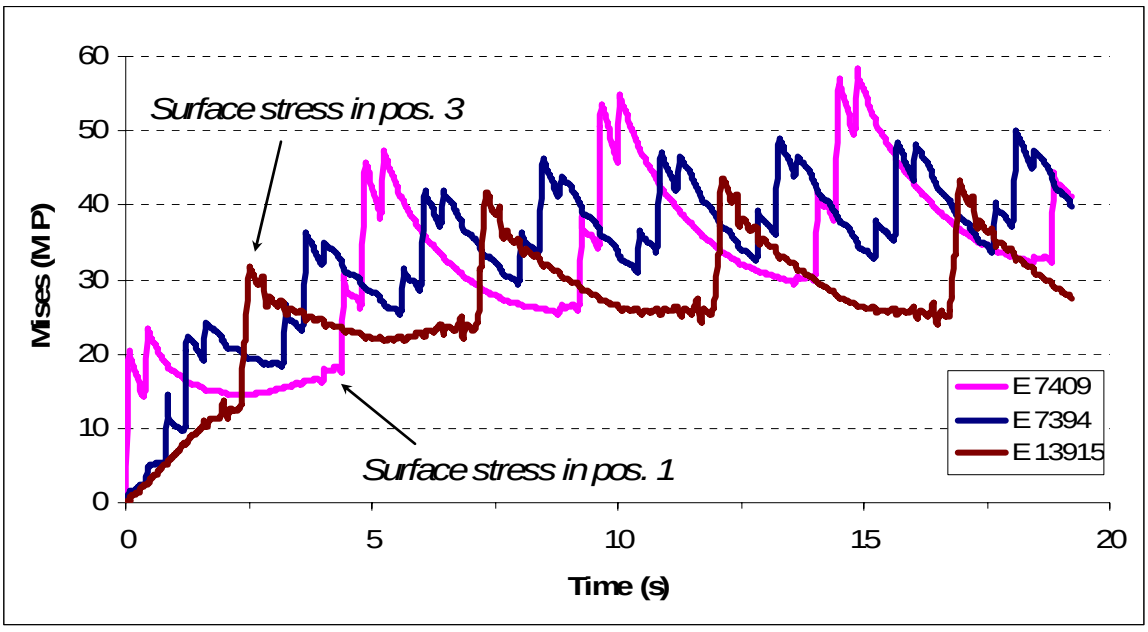


Figure 6.7 Evolution of nodal stress during the simulation of the coating process

In the figure 6.8 is shown the instantaneous field of stress inside the piece. On this figure, the jet of the heat source is over the second node. In this situation, the material in the surface is hotter than the inner material. That means that the surface of the material supports compression stress because of this high temperature the material needs more length. Inside appears traction because the material of around generates a pull outward.

The von Mises stresses on this figure point out that the maximum values appear mainly inside of the piece, specifically inside of the sphere. Other remarkable comment is to consider that the heat is applied around all circumference of the ball pivot. For this reason the stress maximum value appears in two symmetrical positions inside of the sphere.

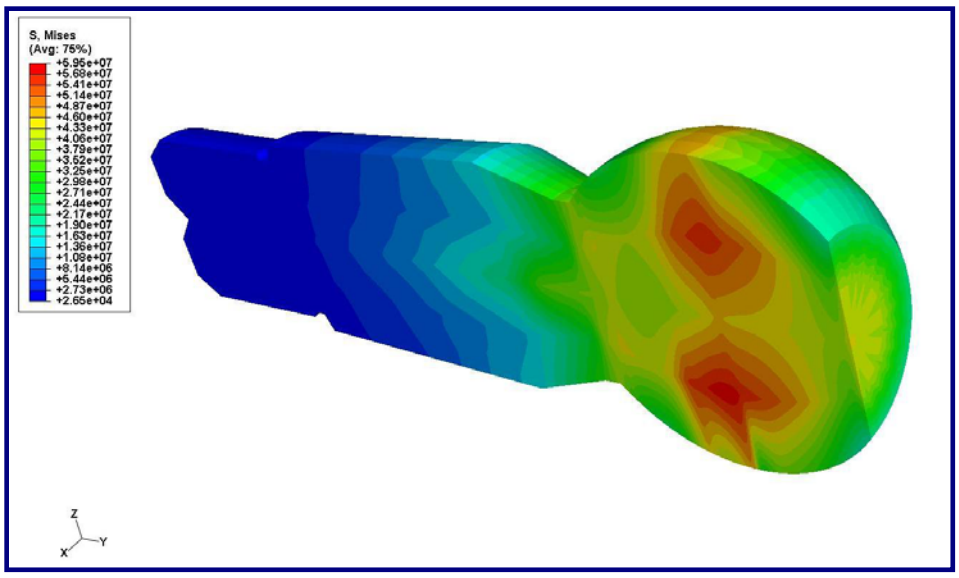


Figure 6.8 Stress field in a section of the piece

7 Conclusions

The production of net shape, high performance thermally sprayed coatings requires advanced automation systems. The torch trajectory is becoming an important process parameter as the demand of spraying complex-shaped parts increases. This project has developed possibilities to generate trajectories and program industrial robots for thermal spraying applications, with the focus on the off-line programming supported by CAD.

In order to consider the real component geometry CAD tools and optical or tactile coordinate measurement systems can be used to provide the required data for the off-line programming of handling devices. Specific software tools were developed for processing the substrate geometry, coordinate transformation and calculation of normal vectors and trajectories.

Besides it is necessary to control the high thermal load over the coated surface. Numerical simulations were developed to evaluate the temperature and stress distributions during coating. Thus torch trajectory planning was coupled with FEM-simulation.

To verify the developed codes, the programmed subroutines were applied to the coating of a ball pivot used in fast train suspension systems. The trajectory of heat source over the ball pivot was obtained. The temperature and stress evolution in the substrate (ball pivot) during the coating process of 20 seconds was calculated and discussed.

8 Bibliography

1. M.C. Acedo Fornell: Thermally sprayed coatings on piston rings, processing and characterization, Augsburg, 2005.

2. J.R. Davis: Handbook of thermal spray technology, USA, 2004. ASM International. ISBN 0-87170-795-0.

3. J. Stokes: The Theory and Application of the HVOF (High Velocity Oxi-fuel). DCU School of Mechanical & Manufacturing Engineering Website, 2003.

4. ABAQUS/Standard User's Manual, version 6.3 and 6.7, vol. I, II and III. Hibbit, Karlssons & Sorensen Inc, USA, 2002.

5. ABAQUS/ Theory Manual, version 6.3 and 6.7. Hibbit, Karlssons & Sorensen Inc, USA, 2002.

6. L. Pawlowski: The science and engineering of thermal spray coatings. John Wiley & Sons Ltd 1995, ISBN 0-471-95253-2.

7. P Fauchais: Understanding plasma spraying. Faculty of Sciences, Laboratoire Sciences des Procédés Céramiques et Traitements de Surface. University of Limoges, France. 2004.

8. M. Floristan: Optimization of the plasma spray process for oxide ceramic coatings used on printing rolls. University of Stuttgart. Institute for Manufacturing Technologies of Ceramic Components and Composites. August 2007.

9. G. Bolelli, L. Lusvarghi: Comparison between plasma- and HVOF-sprayed ceramic coatings. Int. J. Surface Science and Engineering, Vol. 1, Nº 1, 2007.

10. H. Herman, S. Sampath: Thermal spray coatings. Departament of materials science and engineering. University of New York. Stony Brook, NY 11794 – 2275.

11. P. Fauchais, M. Fukumoto: Knowledge Concerning Splat Formation: An Invited Review. Journal of Thermal Spray Technology. 2004.

12. J. Mostaghimi, S. Chandra: Splat formation in plasma-spray coating process. Centre for Advanced Coating Technologies, University of Toronto, Canada, 2002.

13. M. P. Groover, M. Weiss: Industrial robotics; Technology, Programming and Applications. Mc Graw-Hill. United States of America. 1986. ISBN 0-07-024989-X.

14. EN ISO 8373: 1994: Manipulating industrial robots – Vocabulary.

15. J.G. Keramas: Robot Technology Fundamentals. Delmar Publishers. San Francisco, 1999. ISBN 0-8273-8236-7.

16. Lung-Wen Tsai: The Mechanics of Serial and Parallel Manipulators; Robot analysis. John Wiley & Sons, Canada, 1999.

17. J. J. Graig: Introduction to Robotics, Mechanics and Control. Second Edition. Addison-Wesley Publication Co. New York, 1990.

18. T. Smithers; Class notes, Universidad de San Sebastián, 2001.

19. J.Denavit and R.S.Hartenberg: Kinematic Synthesis of linkages, Mechanical Engineering, New York, 1964. Mc Graw Hill, first edition.

20. R. Saltaren, J. Azorin, M. Almonacid, J. Sabater; “Prácticas de robótica utilizando Matlab”, Escuela Politécnica Superior de Elche, 2000.

21. M. W. Spong, M. Vidyasagar: Robot dynamics and control, John Wiley & Sons, Canada,1989. ISBN 0-471-61243-X.

22. J. L. Fuller: Robotics: Introduction, Programming, and Projects. Second Edition, Prentice-Hall, USA, 1999. ISBN 0-13-095543-4.

23. V+ Language Users Guide, Version 12.1. Adept Technology, Inc.

24. V. D. Hunt: Understanding Robotics; Technology Research Corporation, Academic Press. Canada, 1999.

25. J. P. Merlet: Parallel robots: Kluwer Academic Publishers, Netherlands, 2000. ISBN 1-4020-0385-4.

26. Lozano-Perez: Robot programming Technical Report Memo 698, MIT AI, December 1982. In Proceedings of the IEEE, Vol 71, July 1983 and IEEE Tutorial on Robotics, IEEE Computer Society, 1986.

27. Robots ABB webpage: www.abb.com / robotics

28. Robotmaster, Jabez Technologies Inc, available at http://www.delmia.com

29. S. Deng, C. Zeng, P. Charles: New Functions of Thermal Spray Toolkit A Software Developed for Off-line and Rapid Robot Programming. University of Technology of Belfort-Monbéliard, France. International Thermal Spray Conference, 2006.

30. R. Zieris and A. Schmidt: Off-line programming for spraying and laser cladding of three-dimensional surfaces. Institute for Material and Beam Technology, Dresden.

31. S. Deng, C. Zeng: New Functions of Thermal Spray Toolkit A Software Developed for Off-line and Rapid Robot Programming, Paris.

32. R. Zieris, A. Schmidt: Off-line programming for spraying and laser cladding of three-dimensional surfaces, Dresden.

33. Anwendunginsitut zur Einsatzoptimierung von Werkstoffen, Verfahren, Wärmebehandlung. Dr Sommer Werkstofftechnik GmbH. 2007. Issum.

34. J. Richter: Grundlegende thermodynamische und strömungsmechanische Betrachtungen zum Hochgeswindigkeits. Flammspritzen, Hannover. USA.

35. J. Richter, H. Schumacher: Termodynamische und Strömungsmechanische Grundlagen des Hochgeschwindigkeits – Flammspritzens. Thayer School of Engineering, Hanover, USA. 1994.

36. J. Stokes and L. Looney: Properties of WC-Co Components Produced using the HVOF Thermal Spray Process. Ireland.

37. Z. Yao, J. Stiglich, T. S. Sudarshan: Nano-grained Tungsten Carbide-Cobalt (WC-Co), Material Modifications, Inc.

38. Y. Bao, T. Zhang and D.T. Gawne: Computational model for the prediction of the temperatures in the coating during thermal spray. London. U.K.

39. Verein Deutscher Ingenieure, VDI-Gesellschaft Verfahrentechnik und Chemieingenieurwesen (Hrsg): VDI Wärmeatlas, Springen 2006, ISBN 3-540-25503-6.

40. R. Bolot, C. Verdy: Analysis of thermal fluxes transferred by an impinging HVOF jet.

41. E. Dongmo, M. Wenzelburger, R. Gadow: Analysis and optimization of the HVOF process by combined experimental and numerical approaches. Institute of Manufacturing Technologies of ceramic Components and Composites. University of Stuttgart. Germany.

42. M. Honner, P. Cerveny: Heat transfer during HVOF deposition. Department of Physics, University of West Bohemia, Czech Republic. 1997.

43. E. Schelkle: Computerunterstützte Simulationsmethoden (MCAE) im modernen Entwicklungsprozess. Modellierung und Simulation im Maschinenbau. SS 2005. University of Stuttgart. Germany.

9 Annex

9.1 Annex 4

- ***Annex 4.1***: Matlab-subroutine 4.1, which represents the measured coordinates by the teach-in method and the gun trajectory in the KBA blades.

```
% KBA, sequence of points to define the trajectory

  clear all
  close all

  % 1er Step:  introduce the coordinates of the eight points, because these
  %            points define the first line of work.
  %            introduce the lateral distance and the number of minicycles.

  n=8; % numbers of points.
  m=input('Introduce the number of minicycles, m=4, m:');
  vers=input('Introduce the vertical distance, vers=3, versatz:');
  zykle=input('Introduce the number of cycles,zykle=1, zykle:');

  % coordinate of points in the first line of work.
  ex=[1239.641 1187.252 1134.572 1098.316 1089.420 1084.945 1048.940 957.207];
  ey=[272.715 204.848 153.736 76.869 -127.911 -197.037 -268.475 -371.689];
  ez=[795.000 795.000 795.000 795.000 795.000 795.000 795.000 795.000];

  % lateral distance, it allows not to have the gun all time on the pieces.
  d=input('Introduce lateral distance, d=10, d:');

  yy=[-d,0,0,0,0,0,0,d];

  x=ex;
  y=ey+yy;
  z=ez;

  z=z(1);

  % horizontal path.
  contador=1;
  path=[x(1) y(1) z(1)];

  for k=1:zykle
      for i=1:m
  aux=1;
  aux2=0;
         for j=1:2*n
           if j<=n
               path(contador,:)=[x(aux),y(aux),z];
               contador=contador+1;
               aux=aux+1;
           else
               path(contador,:)=[x(n-aux2),y(n-aux2),z-vers];
               contador=contador+1;
               aux2=aux2+1;
           end
         end
        z=z-vers*2;
    end
   z=z+vers*2*m;
  end

  xnn=path(:,1);
  ynn=path(:,2);
  znn=path(:,3);
```

```
figure (1)
plot(ex,ey,'o');
grid on;
hold on;
axis([750 1450 -400 300 750 800]);


figure(2)
plot3(xnn,ynn,znn)
axis([ 900 1300 -400 400 750 800]);
grid on;
hold on;


figure(3)
plot3(xnn,ynn,znn)
grid on;
hold on;
for i=1:n*m*2
   plot3(path(i,1),path(i,2),path(i,3),'ko')
   hold on;
   grid on;
   axis([ 900 1300 -400 400 750 800]);
   pause
end


figure (4)
plot3(ex,ey,ez,'o');
hold on
grid on
axis([750 1450 -400 300 750 800]);
```

- ***Annex 4.2***: Subroutine “Kbauer3.v2”, which represents the coating program in the language of the robot to the KBA pieces.

```
; Represents the coating program in robot's language of KBA-blades

; Two steps


.PROGRAM kbauer3.v2

; program for the coating of blades KoenigundBauer

PROMPT "Introduce cycles", zyk
PROMPT "Introduce velocity", gesch


; tool
SET t_tool=TRANS(-100,0,950,0,0,0)
TOOL t_tool


; 1º Step: "n","m" y "vers"

    n=8;    n° measured points
    m=11;   n° minicycles
    vers=3; distance in Z-axis, units in mm


; 2º Step: introduce the value of the measured points

    SET punk1=TRANS(550,450,750,0,180,45)
    SET punk2=TRANS(550,550,750,0,180,45)
```

```
     SET punk3=TRANS(550,650,750,0,180,45)
     SET punk4=TRANS(550,750,750,0,180,45)
     SET punk5=TRANS(550,800,750,0,180,45)
     SET punk6=TRANS(550,820,750,0,180,45)
     SET punk7=TRANS(550,840,750,0,180,45)
     SET punk8=TRANS(550,860,750,0,180,45)

     DECOMPOSE punk1v[]=punk1
     DECOMPOSE punk2v[]=punk2
     DECOMPOSE punk3v[]=punk3
     DECOMPOSE punk4v[]=punk4
     DECOMPOSE punk5v[]=punk5
     DECOMPOSE punk6v[]=punk6
     DECOMPOSE punk7v[]=punk7
     DECOMPOSE punk8v[]=punk8


; generation of matrix

     matriz[0,0]=punk1v[0]
     matriz[0,1]=punk1v[1]
     matriz[0,2]=punk1v[2]
     matriz[0,3]=punk1v[3]
     matriz[0,4]=punk1v[4]
     matriz[0,5]=punk1v[5]

     matriz[1,0]=punk2v[0]
     matriz[1,1]=punk2v[1]
     matriz[1,2]=punk2v[2]
     matriz[1,3]=punk2v[3]
     matriz[1,4]=punk2v[4]
     matriz[1,5]=punk2v[5]

     matriz[2,0]=punk2v[0]
     matriz[2,1]=punk2v[1]
     matriz[2,2]=punk2v[2]
     matriz[2,3]=punk2v[3]
     matriz[2,4]=punk2v[4]
     matriz[2,5]=punk2v[5]

     matriz[3,0]=punk2v[0]
     matriz[3,1]=punk2v[1]
     matriz[3,2]=punk2v[2]
     matriz[3,3]=punk2v[3]
     matriz[3,4]=punk2v[4]
     matriz[3,5]=punk2v[5]

     matriz[4,0]=punk2v[0]
     matriz[4,1]=punk2v[1]
     matriz[4,2]=punk2v[2]
     matriz[4,3]=punk2v[3]
     matriz[4,4]=punk2v[4]
     matriz[4,5]=punk2v[5]

     matriz[5,0]=punk2v[0]
     matriz[5,1]=punk2v[1]
     matriz[5,2]=punk2v[2]
     matriz[5,3]=punk2v[3]
     matriz[5,4]=punk2v[4]
     matriz[5,5]=punk2v[5]

     matriz[6,0]=punk2v[0]
     matriz[6,1]=punk2v[1]
     matriz[6,2]=punk2v[2]
     matriz[6,3]=punk2v[3]
     matriz[6,4]=punk2v[4]
     matriz[6,5]=punk2v[5]

     matriz[7,0]=punk2v[0]
```

```
    matriz[7,1]=punk2v[1]
    matriz[7,2]=punk2v[2]
    matriz[7,3]=punk2v[3]
    matriz[7,4]=punk2v[4]
    matriz[7,5]=punk2v[5]


; generation of movement

    z=matriz[0,2]


contador=0 ; initialize the contador variable

  FOR i=0 TO m

  aux=0
  aux2=0

    FOR j=0 TO 2*n-1

      IF j<n  THEN

      SET bahn[contador]=TRANS(matriz[aux2,0],matriz[aux2,1],z,matriz[aux2,3],matriz[aux2,4],matriz[aux2,5])

      contador=contador+1
      aux2=aux2+1

      ELSE

      SET bahn[contador]=TRANS(matriz[n-aux,0],matriz[n-aux,1],z-vers,matriz[n-aux,3],matriz[n-aux,4],matriz[n-aux,5])

      aux=aux+1
      contador=contador+1

      END

    END

    z=z-2*vers

  END


; repeat the coating cycle, zyk
; movement

  FOR l=0 TO zyk-1

      FOR k=0 TO LAST (bahn[])

      SPEED gesch, MMPS

      MOVES bahn[k]

      END

  END
```

- ***Annex 4.3***: File “laser”, this file is composed by 27458 points, which are the measurement of the propeller with the laser scanners.

-177.640,-15.781,-33.831
-177.634,-17.574,-34.022
-177.631,-18.371,-34.001
-177.630,-17.799,-34.077
-177.622,-18.773,-34.055
-177.623,-18.263,-34.146
……
-160.703,-65.276,-40.315
-160.690,28.259,0.538
-160.687,-80.588,-46.859
-160.686,-80.441,-45.933
-160.681,20.384,-3.198
-160.672,-78.507,-44.294
……
-121.764,-86.956,-49.235
-121.757,-17.520,-7.600
-121.749,25.222,21.829
-121.744,23.994,21.492
-121.742,-10.909,-3.002
-121.736,-120.027,-58.852
……
-86.219,-125.269,-74.811
-86.213,12.858,16.117
-86.188,-123.004,-73.231
-86.175,-124.740,-74.022
-86.173,-125.134,-74.324
-86.167,-82.854,-58.579
……
-40.909,18.599,25.903
-40.907,-19.479,25.557
-40.904,-28.605,-24.293
-40.902,-57.865,-72.249
-40.901,18.348,26.060
-40.900,-64.230,-76.513
……
-23.418,-47.208,-84.212
-23.418,-45.205,-76.753
-23.416,-44.561,-24.358
-23.401,-47.383,-84.594
-23.394,-44.802,-71.273
-23.389,-78.729,-103.339
……
7.252,-50.518,-113.801
7.311,-50.377,-114.173
7.381,-50.003,-114.770
7.381,-49.898,-114.908
7.475,-50.153,-114.565
7.538,-49.951,-114.830

- ***Annex 4.4***: Matlab-subroutine 4.2, for representation of the measured points with the laser scanners of the propeller.

```
% Program to the representation of the propeller

% Reading of the file "laser", 27458 points
     clear all
     close all
     FileName1='laser.txt';
     %Labels of the Figures
     %Figure1
     xlabel1='X';
     ylabel1='Y';
     zlabel1='Z';
     title1='Schraube Descam ascii';
     axis1='normal';

     fig = figure;


     %Open file.
     Schmat=load(FileName1);
     clear FileName1;

     % dimensions of Schmat
     dim=size(Schmat);


     pinic=1;
     pfin=27458;    % last point of simulation dim(1)
     paso=1         % paso=20 --> 27458/20=1372 points

     % Matrix Schmat2, store the 1372 points
     k=1
      for i=pinic:paso:pfin
         Schmat2(k,:)=Schmat(i,:);
         k=k+1
      end

     % dimensions of Schmat2
     tam2=size(Schmat2)

     % Definition of FaceA with the coordinates stored in Schmat2
     FaceA=Schmat2(1:tam2(1),:);

     % Graphic representation
     for i=pinic:paso:pfin
        plot3(Schmat(i,1),Schmat(i,2),Schmat(i,3),'.');
                   hold on
     end
```

- ***Annex 4.5***: Matlab-subroutine 4.3, for representation of the measured points with the laser scanners of the propeller, also it creates a mesh of the surface and normal vectors.

```
% Program to the representation of the propeller and normal vectors

% Reading of the file “laser”, 27458 points
     clear all
     close all
     FileName1='laser.txt';
     %Labels of the Figures
     %Figure1
     xlabel1='X';
     ylabel1='Y';
     zlabel1='Z';
     title1='Schraube Descam ascii';
     axis1='normal';

     fig = figure;


     %Open file.
     Schmat=load(FileName1);
     clear FileName1;

     % dimensions of Schmat
     dim=size(Schmat);


     pinic=1;
     pfin=27458;  % last point of simulation dim(1)
     paso=20      % paso=20 --> 27458/20=1372 points

     % Matrix Schmat2, store the 1372 points
     k=1
      for i=pinic:paso:pfin
         Schmat2(k,:)=Schmat(i,:);
         k=k+1
      end

     % dimensions of Schmat2
     tam2=size(Schmat2)

     % Definition of FaceA with the coordinates stored in Schmat2
     FaceA=Schmat2(1:tam2(1),:);

     % Graphic representation
     for i=pinic:paso:pfin
        plot3(Schmat(i,1),Schmat(i,2),Schmat(i,3),'.');
                   hold on
     end

     %%%%%%
     %%%%%%

     % Definition of the mesh.
     vers=10;

     xlin = linspace(min(FaceA(:,1)),max(FaceA(:,1)),vers);
     ylin = linspace(min(FaceA(:,2)),max(FaceA(:,2)),vers);

     [x,y] = meshgrid(xlin,ylin);
     z = griddata(FaceA(:,1),FaceA(:,2),FaceA(:,3),x,y,'v4');

     hold on
     surface(x,y,z, 'Marker','o','MarkerFaceColor',[0 1 0],'MarkerSize',3);
```

```
% Normal vectors

 [Nx,Ny,Nz] = surfnorm(x,y,z);
 hold on;
 quiver3(x,y,z,Nx,Ny,Nz,0.95,'r');
```

- ***Annex 4.6***: Matlab-subroutine 4.4, which represents the coordinated points and normal vectors obtained of the mesh of the surface by STL-format in the ball pivot.

```
% Representation of the points and normal vector obtained by CAD method
% of the ball pivot.

% Reading of the file 'Ball_surface'
     clear all
     close all
     FileName1='Ball_surface.txt';

% Open file.
     Schmat=load(FileName1);
     clear FileName1;

     % Calculation of the dimensions of 'Schmat'
     dim=size(Schmat);

     ptoinicial=1;
     ptofinal=85; % last point in the simulation dim(1)
     step=1;

     % Calculation the matrix Schmat2
     k=1;
      for i=ptoinicial:step:ptofinal
         Schmat2(k,:)=Schmat(i,:);
         k=k+1;
      end

     % Calculation the dimensions of Schmat2
     tam2=size(Schmat2);

     % To define the surface 'FaceA' with the coordinates stored in Schmat2
     FaceA=Schmat2(1:tam2(1),:);

     %Graphic representation of the considered points
     xnn=Schmat2(:,1);
     ynn=Schmat2(:,2);
     znn=Schmat2(:,3);


     % figure (1)
     % plot3(xnn,ynn,znn);
     % hold on
     % grid on


     figure (2)
     for i=ptoinicial:step:ptofinal
         plot3(Schmat(i,1),Schmat(i,2),Schmat(i,3),'o');
         axis([80 140 0 30 -1 1.5]);
                  hold on
                  grid on
                  pause
     end
```

```
figure (3)
for i=ptoinicial:step:ptofinal
    plot(Schmat(i,1),Schmat(i,2),'o');
    axis([60 140 -20 60]);
             hold on
             grid on
end


figure (4)
for i=ptoinicial:step:ptofinal
    quiver3(Schmat(i,1),Schmat(i,2),Schmat(i,3),Schmat(i,4),Schmat(i,5),Schmat(i,6),'o');
             hold on
             grid on
     axis([70 150 -20 60 -1 1.5]);
end
```

- ***Annex 4.7***: Matlab-subroutine 4.5, which represents the selected points and normal vectors obtained of the mesh of the surface by STL-format to the ball pivot and also it generates the gun trajectory.

```
% Sequence of points that simulates the trajectory of the gun in the ball pivot

% Reading of the file 'Ball_points'
     clear all
     close all
     FileName1='Ball_points.txt';
     %Labels of the Figures
     %Figure1
     xlabel1='X';
     ylabel1='Y';
     zlabel1='Z';
     title1='Schraube Descam ascii';
     axis1='normal';

     fig = figure;

     % Open file. the matrix 'Schmat' keeps the read points of the file
     Schmat=load(FileName1);
     clear FileName1;

     % Calculation of the dimensions of 'Schmat'
     dim=size(Schmat);


     ptoinicial=1;
     ptofinal=8;   % last point in the simulation dim(1)
     paso=1;

     % Calculation the matrix Schmat2
     k=1;
      for i=ptoinicial:paso:ptofinal
         Schmat2(k,:)=Schmat(i,:);
         k=k+1;
      end

     % Calculation the dimensions of Schmat2
     tam2=size(Schmat2);

     % To define the surface 'FaceA' with the coordinates stored in Schmat2
     FaceA=Schmat2(1:tam2(1),:);
```

```
%Graphic representation of the considered points
xnn=Schmat2(:,1);
ynn=Schmat2(:,2);
znn=Schmat2(:,3);


figure (1)
plot3(xnn,ynn,znn);
hold on
grid on


figure (2)
for i=ptoinicial:paso:ptofinal
    plot3(Schmat(i,1),Schmat(i,2),Schmat(i,3),'o');
    axis([80 140 0 30 -1 1.5]);
             hold on
             grid on
             pause
end


figure (3)
for i=ptoinicial:paso:ptofinal
    plot(Schmat(i,1),Schmat(i,2),'o');
    axis([60 140 -20 60]);
             hold on
             grid on
 end


figure (4)
for i=ptoinicial:paso:ptofinal
     quiver3(Schmat(i,1),Schmat(i,2),Schmat(i,3),Schmat(i,4),Schmat(i,5),Schmat(i,6),'o');
             hold on
             grid on
     axis([70 150 -20 60 -1 1.5]);
end
```

- ***Annex 4.8***: Subroutine “kugelzapfen3”, which represents the coating program in the language of the robot to the ball pivot.

```
; Represents the coating program in robot's language of the ball pivot

; Two steps


.PROGRAM kugelzapfen3

; program for the coating of the ball pivot

PROMPT "Introduce cycles“, zyk
PROMPT "Introduce velocity", gesch


; tool
SET t_tool=TRANS(-100,0,950,0,0,0)
TOOL t_tool


; 1º Step: "n","m" y "vers"

    n=7;     n° measured points
    m=1;     n° minicycles
    vers=0;  distance in Z-axis, units in mm
```

```
; 2º Step: introduce the value of the measured points

    SET punk1=TRANS(1480.0,-657,-405,-20.00,90,0)
    SET punk2=TRANS(1478.3,-652,-405,-12.66,90,0)
    SET punk3=TRANS(1476.5,-647,-405,-6.333,90,0)
    SET punk4=TRANS(1475.0,-642,-405,0.0000,90,0)
    SET punk5=TRANS(1476.5,-637,-405,6.3333,90,0)
    SET punk6=TRANS(1478.3,-632,-405,12.666,90,0)
    SET punk7=TRANS(1478.3,-627,-405,20.000,90,0)

    DECOMPOSE punk1v[]=punk1
    DECOMPOSE punk2v[]=punk2
    DECOMPOSE punk3v[]=punk3
    DECOMPOSE punk4v[]=punk4
    DECOMPOSE punk5v[]=punk5
    DECOMPOSE punk6v[]=punk6
    DECOMPOSE punk7v[]=punk7


; generation of matrix

    matriz[0,0]=punk1v[0]
    matriz[0,1]=punk1v[1]
    matriz[0,2]=punk1v[2]
    matriz[0,3]=punk1v[3]
    matriz[0,4]=punk1v[4]
    matriz[0,5]=punk1v[5]

    matriz[1,0]=punk2v[0]
    matriz[1,1]=punk2v[1]
    matriz[1,2]=punk2v[2]
    matriz[1,3]=punk2v[3]
    matriz[1,4]=punk2v[4]
    matriz[1,5]=punk2v[5]

    matriz[2,0]=punk3v[0]
    matriz[2,1]=punk3v[1]
    matriz[2,2]=punk3v[2]
    matriz[2,3]=punk3v[3]
    matriz[2,4]=punk3v[4]
    matriz[2,5]=punk3v[5]

    matriz[3,0]=punk4v[0]
    matriz[3,1]=punk4v[1]
    matriz[3,2]=punk4v[2]
    matriz[3,3]=punk4v[3]
    matriz[3,4]=punk4v[4]
    matriz[3,5]=punk4v[5]

    matriz[4,0]=punk5v[0]
    matriz[4,1]=punk5v[1]
    matriz[4,2]=punk5v[2]
    matriz[4,3]=punk5v[3]
    matriz[4,4]=punk5v[4]
    matriz[4,5]=punk5v[5]

    matriz[5,0]=punk6v[0]
    matriz[5,1]=punk6v[1]
    matriz[5,2]=punk6v[2]
    matriz[5,3]=punk6v[3]
    matriz[5,4]=punk6v[4]
    matriz[5,5]=punk6v[5]

    matriz[6,0]=punk7v[0]
    matriz[6,1]=punk7v[1]
    matriz[6,2]=punk7v[2]
    matriz[6,3]=punk7v[3]
    matriz[6,4]=punk7v[4]
    matriz[6,5]=punk7v[5]
```

```
; generation of movement

    z=matriz[0,2]


contador=0 ; initialize the contador variable

  FOR i=0 TO m

  aux=0
  aux2=0

    FOR j=0 TO 2*n-1

      IF j<n  THEN

       SET bahn[contador]=TRANS(matriz[aux2,0],matriz[aux2,1],z,matriz[aux2,3],matriz[aux2,4],matriz[aux2,5])

      contador=contador+1
      aux2=aux2+1

      ELSE

      SET bahn[contador]=TRANS(matriz[n-aux,0],matriz[n-aux,1],z,matriz[n-aux,3],matriz[n-aux,4],matriz[n-aux,5])

      aux=aux+1
      contador=contador+1

      END

    END

  END


; repeat the coating cycle, zyk
; movement

  FOR l=0 TO zyk-1

      FOR k=0 TO LAST (bahn[])

    SPEED gesch, MMPS

    MOVES bahn[k]

    END

   END
```

- ***Annex 4.9***: Matlab-subroutine 4.6, which represents the coordinated points and normal vectors obtained of the mesh of the surface by STL-format to the KBA-blades and also it generates the gun trajectory.

```
% Sequence of points that simulates the trajectory of the gun in the KBA-blades

% Reading of the file 'KBA_piece_ptos'
     clear all
     close all
     FileName1='KBA_piece_ptos.txt';
     %Labels of the Figures
     %Figure1
     xlabel1='X';
     ylabel1='Y';
     zlabel1='Z';
     title1='Schraube Descam ascii';
     axis1='normal';

     fig = figure;

     % Open file. The matrix 'Schmat' keeps the read points of the file
     Schmat=load(FileName1);
     clear FileName1;

     % Calculation of the dimensions of 'Schmat'
     dim=size(Schmat);


     ptoinicial=1;
     ptofinal=161; % last point in the simulation dim(1)
     paso=1;

     % Calculation the matrix Schmat2
     k=1;
      for i=ptoinicial:paso:ptofinal
         Schmat2(k,:)=Schmat(i,:);
         k=k+1;
      end

     % Calculation the dimensions of Schmat2
     tam2=size(Schmat2);

     % To define the surface 'FaceA' with the coordinates stored in Schmat2
     FaceA=Schmat2(1:tam2(1),:);

     %Graphic representation of the considered points
     xnn=Schmat2(:,1);
     ynn=Schmat2(:,2);
     znn=Schmat2(:,3);


     figure (3)
     plot3(xnn,ynn,znn);
     hold on
     grid on
     axis([-50 600 -50 650 -10 30]);


     figure (2)
     for i=ptoinicial:paso:ptofinal
          quiver3(Schmat(i,1),Schmat(i,2),Schmat(i,3),Schmat(i,4),Schmat(i,5),Schmat(i,6),'o');
                   hold on
                   grid on
          axis([-150 600 50 800 0 20]);
     end
```

```
    figure (1)
    for i=ptoinicial:paso:ptofinal
        plot(Schmat(i,1),Schmat(i,2),'o');
        axis([-30 620 0 650]);
                hold on
                grid on
    end


  % figure (4)
  % for i=ptoinicial:paso:ptofinal
  %     plot3(Schmat(i,1),Schmat(i,2),Schmat(i,3),'o');
  %     axis([-50 600 50 450 0 20]);
  %             hold on
  %             grid on
  %             pause

  % end
```

- ***Annex 4.10***: Subroutine “Kbauer3.v3”, which represents the coating program in the language of the robot to the KBA pieces for the CAD method.

```
; Represents the coating program in robot's language of KBA-blades

; Two steps


.PROGRAM kbauer3.v3

; program for the coating of blades KoenigundBauer


PROMPT "Introduce cycles", zyk
PROMPT "Introduce velocity", gesch



; tool
SET t_tool=TRANS(-100,0,950,0,0,0)
TOOL t_tool


; 1º Step: "n","m" y "vers"

   n=23;    n° measured points
   m=10;   n° minicycles
   vers=3; distance in Z-axis, units in mm


; 2º Step: introduce the value of the measured points

   SET punk1=TRANS(0.6658137,200.9755,19.500000,-0.7433208,0.6689351,0.0000000)
   SET punk2=TRANS(16.597060,218.2542,19.500000,-0.7433208,0.6689351,0.0000000)
   SET punk3=TRANS(33.736440,234.3353,19.500000,-0.6929863,0.7209507,0.0000000)
   SET punk4=TRANS(51.994110,249.1344,19.500000,-0.6390212,0.7691891,0.0000000)
   SET punk5=TRANS(71.274490,262.5742,19.500000,-0.5817081,0.8133976,0.0000000)
   SET punk6=TRANS(92.593370,274.3887,19.500000,-0.4980884,0.8671263,0.0000000)
   SET punk7=TRANS(114.90760,284.1945,19.500000,-0.4163108,0.9092224,0.0000000)
   SET punk8=TRANS(138.02840,291.9087,19.500000,-0.3310114,0.9436268,0.0000000)
   SET punk9=TRANS(161.76010,297.4659,19.500000,-0.2429118,0.9700484,0.0000000)
   SET punk10=TRANS(185.90210,300.8191,19.500000,-0.1527572,0.9882638,0.0000000)
   SET punk11=TRANS(210.25010,301.9400,19.500000,0.0000000,1.0000000,0.0000000)
   SET punk12=TRANS(291.75010,301.9400,19.500000,0.0000000,1.0000000,0.0000000)
   SET punk13=TRANS(374.13680,301.8600,19.500000,0.0000000,1.0000000,0.0000000)
   SET punk14=TRANS(398.01680,302.9590,19.500000,-0.1223179,0.9924909,0.0000000)
```

```
   SET punk15=TRANS(421.69500,306.2466,19.500000,-0.2129576,0.9770615,0.0000000)
   SET punk16=TRANS(444.97110,311.6951,19.500000,-0.3017970,0.9533722,0.0000000)
   SET punk17=TRANS(467.64850,319.2583,19.500000,-0.3880851,0.9216235,0.0000000)
   SET punk18=TRANS(489.53530,328.8725,19.500000,-0.4710925,0.8820838,0.0000000)
   SET punk19=TRANS(510.44650,340.4562,19.500000,-0.5620251,0.8271202,0.0000000)
   SET punk20=TRANS(529.41770,353.6843,19.500000,-0.6203509,0.7843244,0.0000000)
   SET punk21=TRANS(547.38280,368.2493,19.500000,-0.6754302,0.7374240,0.0000000)
   SET punk22=TRANS(564.24790,384.0750,19.500000,-0.7433297,0.6689252,0.0000000)
   SET punk23=TRANS(579.92480,401.0786,19.500000,-0.7433297,0.6689252,0.0000000)

   DECOMPOSE punk1v[]=punk1
   DECOMPOSE punk2v[]=punk2
   DECOMPOSE punk3v[]=punk3
   DECOMPOSE punk4v[]=punk4
   DECOMPOSE punk5v[]=punk5
   DECOMPOSE punk6v[]=punk6
   DECOMPOSE punk7v[]=punk7
   DECOMPOSE punk8v[]=punk8
   DECOMPOSE punk9v[]=punk9
   DECOMPOSE punk10v[]=punk10
   DECOMPOSE punk11v[]=punk11
   DECOMPOSE punk12v[]=punk12
   DECOMPOSE punk13v[]=punk13
   DECOMPOSE punk14v[]=punk14
   DECOMPOSE punk15v[]=punk15
   DECOMPOSE punk16v[]=punk16
   DECOMPOSE punk17v[]=punk17
   DECOMPOSE punk18v[]=punk18
   DECOMPOSE punk19v[]=punk19
   DECOMPOSE punk20v[]=punk20
   DECOMPOSE punk21v[]=punk21
   DECOMPOSE punk22v[]=punk22
   DECOMPOSE punk23v[]=punk23


; generation of matrix

   matriz[0,0]=punk1v[0]
   matriz[0,1]=punk1v[1]
   matriz[0,2]=punk1v[2]
   matriz[0,3]=punk1v[3]
   matriz[0,4]=punk1v[4]
   matriz[0,5]=punk1v[5]

   matriz[1,0]=punk2v[0]
   matriz[1,1]=punk2v[1]
   matriz[1,2]=punk2v[2]
   matriz[1,3]=punk2v[3]
   matriz[1,4]=punk2v[4]
   matriz[1,5]=punk2v[5]

   matriz[2,0]=punk3v[0]
   matriz[2,1]=punk3v[1]
   matriz[2,2]=punk3v[2]
   matriz[2,3]=punk3v[3]
   matriz[2,4]=punk3v[4]
   matriz[2,5]=punk3v[5]

   matriz[3,0]=punk4v[0]
   matriz[3,1]=punk4v[1]
   matriz[3,2]=punk4v[2]
   matriz[3,3]=punk4v[3]
   matriz[3,4]=punk4v[4]
   matriz[3,5]=punk4v[5]

   matriz[4,0]=punk5v[0]
   matriz[4,1]=punk5v[1]
   matriz[4,2]=punk5v[2]
   matriz[4,3]=punk5v[3]
```

```
matriz[4,4]=punk5v[4]
matriz[4,5]=punk5v[5]

matriz[5,0]=punk6v[0]
matriz[5,1]=punk6v[1]
matriz[5,2]=punk6v[2]
matriz[5,3]=punk6v[3]
matriz[5,4]=punk6v[4]
matriz[5,5]=punk6v[5]

matriz[6,0]=punk7v[0]
matriz[6,1]=punk7v[1]
matriz[6,2]=punk7v[2]
matriz[6,3]=punk7v[3]
matriz[6,4]=punk7v[4]
matriz[6,5]=punk7v[5]

matriz[7,0]=punk8v[0]
matriz[7,1]=punk8v[1]
matriz[7,2]=punk8v[2]
matriz[7,3]=punk8v[3]
matriz[7,4]=punk8v[4]
matriz[7,5]=punk8v[5]

matriz[8,0]=punk9v[0]
matriz[8,1]=punk9v[1]
matriz[8,2]=punk9v[2]
matriz[8,3]=punk9v[3]
matriz[8,4]=punk9v[4]
matriz[8,5]=punk9v[5]

matriz[9,0]=punk10v[0]
matriz[9,1]=punk10v[1]
matriz[9,2]=punk10v[2]
matriz[9,3]=punk10v[3]
matriz[9,4]=punk10v[4]
matriz[9,5]=punk10v[5]

matriz[10,0]=punk11v[0]
matriz[10,1]=punk11v[1]
matriz[10,2]=punk11v[2]
matriz[10,3]=punk11v[3]
matriz[10,4]=punk11v[4]
matriz[10,5]=punk11v[5]

matriz[11,0]=punk12v[0]
matriz[11,1]=punk12v[1]
matriz[11,2]=punk12v[2]
matriz[11,3]=punk12v[3]
matriz[11,4]=punk12v[4]
matriz[11,5]=punk12v[5]

matriz[12,0]=punk13v[0]
matriz[12,1]=punk13v[1]
matriz[12,2]=punk13v[2]
matriz[12,3]=punk13v[3]
matriz[12,4]=punk13v[4]
matriz[12,5]=punk13v[5]

matriz[13,0]=punk14v[0]
matriz[13,1]=punk14v[1]
matriz[13,2]=punk14v[2]
matriz[13,3]=punk14v[3]
matriz[13,4]=punk14v[4]
matriz[13,5]=punk14v[5]

matriz[14,0]=punk15v[0]
matriz[14,1]=punk15v[1]
matriz[14,2]=punk15v[2]
```

```
   matriz[14,3]=punk15v[3]
   matriz[14,4]=punk15v[4]
   matriz[14,5]=punk15v[5]

   matriz[15,0]=punk16v[0]
   matriz[15,1]=punk16v[1]
   matriz[15,2]=punk16v[2]
   matriz[15,3]=punk16v[3]
   matriz[15,4]=punk16v[4]
   matriz[15,5]=punk16v[5]

   matriz[16,0]=punk17v[0]
   matriz[16,1]=punk17v[1]
   matriz[16,2]=punk17v[2]
   matriz[16,3]=punk17v[3]
   matriz[16,4]=punk17v[4]
   matriz[16,5]=punk17v[5]

   matriz[17,0]=punk18v[0]
   matriz[17,1]=punk18v[1]
   matriz[17,2]=punk18v[2]
   matriz[17,3]=punk18v[3]
   matriz[17,4]=punk18v[4]
   matriz[17,5]=punk18v[5]

   matriz[18,0]=punk19v[0]
   matriz[18,1]=punk19v[1]
   matriz[18,2]=punk19v[2]
   matriz[18,3]=punk19v[3]
   matriz[18,4]=punk19v[4]
   matriz[18,5]=punk19v[5]

   matriz[19,0]=punk20v[0]
   matriz[19,1]=punk20v[1]
   matriz[19,2]=punk20v[2]
   matriz[19,3]=punk20v[3]
   matriz[19,4]=punk20v[4]
   matriz[19,5]=punk20v[5]

   matriz[20,0]=punk21v[0]
   matriz[20,1]=punk21v[1]
   matriz[20,2]=punk21v[2]
   matriz[20,3]=punk21v[3]
   matriz[20,4]=punk21v[4]
   matriz[20,5]=punk21v[5]

   matriz[21,0]=punk22v[0]
   matriz[21,1]=punk22v[1]
   matriz[21,2]=punk22v[2]
   matriz[21,3]=punk22v[3]
   matriz[21,4]=punk22v[4]
   matriz[21,5]=punk22v[5]

   matriz[22,0]=punk23v[0]
   matriz[22,1]=punk23v[1]
   matriz[22,2]=punk23v[2]
   matriz[22,3]=punk23v[3]
   matriz[22,4]=punk23v[4]
   matriz[22,5]=punk23v[5]


; generation of movement

   z=matriz[0,2]
```

```
contador=0 ; initialize the contador variable

  FOR i=0 TO m

  aux=0
  aux2=0

    FOR j=0 TO 2*n-1

      IF j<n  THEN

      SET
bahn[contador]=TRANS(matriz[aux2,0],matriz[aux2,1],z,matriz[aux2,3],
matriz[aux2,4],matriz[aux2,5])

      contador=contador+1
      aux2=aux2+1

      ELSE

      SET     bahn[contador]=TRANS(matriz[n-aux,0],matriz[n-aux,1],z-
vers,matriz[n-aux,3],matriz[n-aux,4],matriz[n-aux,5])

      aux=aux+1
      contador=contador+1

      END

    END

    z=z-2*vers

  END


; repeat the coating cycle, zyk
; movement

  FOR l=0 TO zyk-1

      FOR k=0 TO LAST (bahn[])

   SPEED gesch, MMPS

   MOVES bahn[k]

   END

   END
```

9.2 Annex 5

- ***Annex 5.1***: Parameters of the HVOF deposition process.

Equipment	JP-5000, Tafa Inc.
Inside gun diameter	11 mm
Spraying distance	250 mm
Gun velocity	225 mm/s
Substrate	Ball pivot (Kugelzapfen)
Powder	83 % WC + 17 % Co 8.5 kg/h
Powder carrier gas	Nitrogen 414 kPa 12.5 l/min
Fuel	Kerosene 1173 kPa 19.5 l/h Oxygen 1450 kPa 53580 l/h
Electric power	150 kW
Spray gun temperature	4500 ºC
Flame temperature	3100 ºC
Particle temperature	1800 ºC
Particle impact velocity	800 m/s
Jet velocity	600 m/s
Torch angle	90 º
Rotation plate	150 rpm
Torch speed	12.5 mm/s
Spraying technique	HVOF

Table A.5.1 Parameters of the HVOF deposition process

- ***Annex 5.2***: Code of the program for the simulation in the coating process of the ball pivot in "ABAQUS 6.7" software.

```
*Heading
** Job name: Kugelzapfen_4-Zyklus Model name: Kugelzapfen_20-Sekunden
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=KUGELZAPFEN
*Node
      1, -0.00872128271, 0.0900833309, 0.0151057038
      2,           0., 0.0900833309,           0.
      3,           0., 0.0904999971,           0.
……
……

      19397, -0.00148379989, 0.0785000026,  0.011270578
      19398, -0.0013440781, 0.0874421149, 0.0102092857
      19399, -0.0102221724,  0.127588615, 0.00134577369
*Element, type=C3D8T
      1,  9850,  2527,  9849, 17024,  2525,     1,  2526,  9848
      2,  9851,  2529,  9850, 17024,  2528,     2,  2525,  9848
      3,  9852,  2531,  9851, 17024,  2530,     3,  2528,  9848
……
```

```
.....

   14302, 17023,  3086, 10377, 19399,  3866,   205,  3088, 11121
   14303, 10377,  3085, 14548, 19399,  3088,   202,  3864, 11121
   14304, 17023,  8191, 14548, 19399,  3086,   201,  3085, 10377
*Nset, nset=_PICKEDSET2, internal, generate
    1,  19399,      1
*Elset, elset=_PICKEDSET2, internal, generate
     1,  14304,      1
** Section: Section-1-_PICKEDSET2
*Solid Section, elset=_PICKEDSET2, material=STEEL-42CRMOS4
   1.,
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=KUGELZAPFEN-1, part=KUGELZAPFEN
*End Instance
**
*Nset, nset=_PICKEDSET23, internal, instance=KUGELZAPFEN-1, generate
    1,  19399,      1
*Elset, elset=_PICKEDSET23, internal, instance=KUGELZAPFEN-1, generate
     1,  14304,      1
*Nset, nset=_PICKEDSET158, internal, instance=KUGELZAPFEN-1
    572,   575,   576,   579,   654,   655,   656,   657,   984,   985,  1045,  1046,  1047,  1048,  1244,  1245
   1246,  1247,  1284,  1285,  1286,  1287,  1607,  1608,  1610,  1611,  1628,  1629,  1630,  1631,  1899,  1900
   1901,  1902,  1917,  1919,  2208,  2209,  2210,  2211,  2221,  2222,  2502,  2503,  2508,  2509,  2517,  2518
   4039,  4041,  4046,  4048,  4212,  4213,  4215,  4218,  4985,  4986,  4987,  5133,  5135,  5137,  5139,  5681
   5682,  5684,  5686,  5773,  5774,  5775,  5776,  6611,  6612,  6614,  6617,  6663,  6664,  6665,  6666,  7358
   7359,  7360,  7361,  7399,  7401,  7405,  8180,  8181,  8183,  8185,  8216,  8217,  8218,  9430,  9432,  9445
   9446,  9542,  9543,  9560,  9561,  9562,  9684,  9685,  9693,  9695,  9696,  9753,  9754,  9773,  9777,  9778
   9812,  9813,  9817,  9818,  9834,  9835,  9839,  9840, 11235, 11353, 11925, 12041, 12506, 12572, 13259, 13298
  13855, 13889, 14543, 14575, 16195, 16225, 16400, 16434, 16702, 16719, 16847, 16880, 16933, 16954, 16988, 17007
*Elset, elset=_PICKEDSET158, internal, instance=KUGELZAPFEN-1
   2611,  2614,  2615,  2618,  2782,  2783,  2787,  2788,  3597,  3598,  3599,  3600,  3763,  3766,  3767,  3770
   4470,  4471,  4473,  4474,  4563,  4564,  4565,  4566,  5588,  5589,  5591,  5594,  5647,  5648,  5649,  5650
   6447,  6448,  6449,  6450,  6499,  6502,  6503,  6506,  7492,  7493,  7495,  7498,  7551,  7552,  7553,  7554
  11611, 11614, 11617, 11618, 11704, 11705, 11707, 11710, 12234, 12235, 12237, 12240, 12335, 12336, 12337, 12338
  13231, 13232, 13233, 13234, 13283, 13286, 13287, 13290, 13727, 13728, 13729, 13730, 13821, 13824, 13825, 13828
  13962, 13963, 13965, 13968, 14057, 14058, 14059, 14060, 14161, 14162, 14163, 14164, 14252, 14253, 14255, 14256
*Elset, elset=__PICKEDSURF10_S3, internal, instance=KUGELZAPFEN-1
    22,   196,   340,   502,   682,   934,   944,   947,   948,  3067,  3079,  3091,  3103,  3115,  7407,  7411
  7419,  7423,  7861,  7865,  7873,  7877,  8005,  8009,  8017,  8021,  8029,  8033,  8041,  8045,  8053,  8057
 11451, 11564, 11567, 11589, 11600, 11809, 11851, 12903, 12907, 12975, 12979
*Elset, elset=__PICKEDSURF10_S4, internal, instance=KUGELZAPFEN-1
    19,    23,    24,   193,   197,   198,   337,   341,   342,   499,   503,   504,   679,   683,   684,   931
   935,   936,   945,  3066,  3068,  3070,  3078,  3080,  3082,  3090,  3092,  3094,  3102,  3104,  3106,  3114
  3116,  3118,  7409,  7412,  7421,  7424,  7863,  7866,  7875,  7878,  8007,  8010,  8019,  8022,  8031,  8034
  8043,  8046,  8055,  8058, 11450, 11452, 11454, 11565, 11568, 11588, 11590, 11592, 11599, 11808, 11810, 11812
 11850, 11852, 11854, 12905, 12908, 12977, 12980
*Elset, elset=__PICKEDSURF10_S2, internal, instance=KUGELZAPFEN-1
    49,    50,    51,    52,   163,   164,   165,   166,   307,   308,   309,   310,   457,   458,   459,   460
   607,   608,   609,   610,   781,   782,   783,   784,   793,   794,   795,   796, 11602, 11603
*Elset, elset=__PICKEDSURF11_S4, internal, instance=KUGELZAPFEN-1
    80,    83,    84,   181,   185,   186,   325,   329,   330,   418,   481,   485,   486,   637,   641,   642
   835,   839,   840,   844,   884,   887,   888,  7718,  7721,  7730,  7733, 10184, 10187, 10215, 10218, 10257
 10285, 10288, 10290, 10329, 10357, 10360, 10362, 10401, 10429, 10432, 10434, 10473, 10501, 10504, 10506, 10545
 10585, 10588, 10590, 11832, 11835, 11892, 11895, 11915, 11918, 11920, 12939, 12942, 12944, 12945, 12948, 12950
 13799, 13802, 13811, 13814, 13907, 13910, 13912, 13977, 13988, 13991
*Elset, elset=__PICKEDSURF11_S3, internal, instance=KUGELZAPFEN-1
    81,   184,   328,   415,   419,   420,   484,   640,   838,   841,   845,   846,   885,  7719,  7722,  7731
  7734, 10185, 10188, 10213, 10217, 10256, 10258, 10260, 10287, 10328, 10330, 10332, 10359, 10400, 10402, 10404
 10431, 10472, 10474, 10476, 10503, 10544, 10546, 10548, 10587, 11833, 11836, 11893, 11896, 11917, 12941, 12947
 13797, 13801, 13809, 13813, 13909, 13975, 13978, 13989, 13992
*Elset, elset=__PICKEDSURF11_S2, internal, instance=KUGELZAPFEN-1
    37,    38,    39,    40,   229,   230,   231,   232,   487,   488,   489,   490,   643,   644,   645,   646
   889,   890,   891,   892, 13980
.....
.....

*Elset, elset=__PICKEDSURF156_S3_3, internal, instance=KUGELZAPFEN-1
    22,   196,   340,   502,   682,   934,   944,   947,   948,  3067,  3079,  3091,  3103,  3115,  7407,  7411
  7419,  7423,  7861,  7865,  7873,  7877,  8005,  8009,  8017,  8021,  8029,  8033,  8041,  8045,  8053,  8057
 11451, 11564, 11567, 11589, 11600, 11809, 11851, 12903, 12907, 12975, 12979
```

```
*Elset, elset=__PICKEDSURF156_S4_3, internal, instance=KUGELZAPFEN-1
     19,    23,    24,   193,   197,   198,   337,   341,   342,   499,   503,   504,   679,   683,   684,   931
    935,   936,   945,  3066,  3068,  3070,  3078,  3080,  3082,  3090,  3092,  3094,  3102,  3104,  3106,  3114
   3116,  3118,  7409,  7412,  7421,  7424,  7863,  7866,  7875,  7878,  8007,  8010,  8019,  8022,  8031,  8034
   8043,  8046,  8055,  8058, 11450, 11452, 11454, 11565, 11568, 11588, 11590, 11592, 11599, 11808, 11810, 11812
  11850, 11852, 11854, 12905, 12908, 12977, 12980
*Elset, elset=__PICKEDSURF156_S2_3, internal, instance=KUGELZAPFEN-1
     49,    50,    51,    52,   163,   164,   165,   166,   307,   308,   309,   310,   457,   458,   459,   460
    607,   608,   609,   610,   781,   782,   783,   784,   793,   794,   795,   796, 11602, 11603
*Elset, elset=__PICKEDSURF10_S3_4, internal, instance=KUGELZAPFEN-1
     22,   196,   340,   502,   682,   934,   944,   947,   948,  3067,  3079,  3091,  3103,  3115,  7407,  7411
   7419,  7423,  7861,  7865,  7873,  7877,  8005,  8009,  8017,  8021,  8029,  8033,  8041,  8045,  8053,  8057
  11451, 11564, 11567, 11589, 11600, 11809, 11851, 12903, 12907, 12975, 12979
*Elset, elset=__PICKEDSURF10_S4_4, internal, instance=KUGELZAPFEN-1
     19,    23,    24,   193,   197,   198,   337,   341,   342,   499,   503,   504,   679,   683,   684,   931
    935,   936,   945,  3066,  3068,  3070,  3078,  3080,  3082,  3090,  3092,  3094,  3102,  3104,  3106,  3114
   3116,  3118,  7409,  7412,  7421,  7424,  7863,  7866,  7875,  7878,  8007,  8010,  8019,  8022,  8031,  8034
   8043,  8046,  8055,  8058, 11450, 11452, 11454, 11565, 11568, 11588, 11590, 11592, 11599, 11808, 11810, 11812
  11850, 11852, 11854, 12905, 12908, 12977, 12980
*Elset, elset=__PICKEDSURF10_S2_4, internal, instance=KUGELZAPFEN-1
     49,    50,    51,    52,   163,   164,   165,   166,   307,   308,   309,   310,   457,   458,   459,   460
    607,   608,   609,   610,   781,   782,   783,   784,   793,   794,   795,   796, 11602, 11603
*Surface, type=ELEMENT, name=_PICKEDSURF10, internal
   __PICKEDSURF10_S3_4, S3
   __PICKEDSURF10_S4_4, S4
   __PICKEDSURF10_S2_4, S2
*Elset, elset=__PICKEDSURF11_S4_4, internal, instance=KUGELZAPFEN-1
     80,    83,    84,   181,   185,   186,   325,   329,   330,   418,   481,   485,   486,   637,   641,   642
    835,   839,   840,   844,   884,   887,   888,  7718,  7721,  7730,  7733, 10184, 10187, 10215, 10218, 10257
  10285, 10288, 10290, 10329, 10357, 10360, 10362, 10401, 10429, 10432, 10434, 10473, 10501, 10504, 10506, 10545
  10585, 10588, 10590, 11832, 11835, 11892, 11895, 11915, 11918, 11920, 12939, 12942, 12944, 12945, 12948, 12950
  13799, 13802, 13811, 13814, 13907, 13910, 13912, 13977, 13988, 13991
*Elset, elset=__PICKEDSURF11_S3_4, internal, instance=KUGELZAPFEN-1
     81,   184,   328,   415,   419,   420,   484,   640,   838,   841,   845,   846,   885,  7719,  7722,  7731
   7734, 10185, 10188, 10213, 10217, 10256, 10258, 10260, 10287, 10328, 10330, 10332, 10359, 10400, 10402, 10404
  10431, 10472, 10474, 10476, 10503, 10544, 10546, 10548, 10587, 11833, 11836, 11893, 11896, 11917, 12941, 12947
  13797, 13801, 13809, 13813, 13909, 13975, 13978, 13989, 13992
*Elset, elset=__PICKEDSURF11_S2_4, internal, instance=KUGELZAPFEN-1
     37,    38,    39,    40,   229,   230,   231,   232,   487,   488,   489,   490,   643,   644,   645,   646
    889,   890,   891,   892, 13980
*Surface, type=ELEMENT, name=_PICKEDSURF11, internal
   __PICKEDSURF11_S4_4, S4
   __PICKEDSURF11_S3_4, S3
   __PICKEDSURF11_S2_4, S2
.....
.....

*Elset, elset=__PICKEDSURF156_S3_4, internal, instance=KUGELZAPFEN-1
     22,   196,   340,   502,   682,   934,   944,   947,   948,  3067,  3079,  3091,  3103,  3115,  7407,  7411
   7419,  7423,  7861,  7865,  7873,  7877,  8005,  8009,  8017,  8021,  8029,  8033,  8041,  8045,  8053,  8057
  11451, 11564, 11567, 11589, 11600, 11809, 11851, 12903, 12907, 12975, 12979
*Elset, elset=__PICKEDSURF156_S4_4, internal, instance=KUGELZAPFEN-1
     19,    23,    24,   193,   197,   198,   337,   341,   342,   499,   503,   504,   679,   683,   684,   931
    935,   936,   945,  3066,  3068,  3070,  3078,  3080,  3082,  3090,  3092,  3094,  3102,  3104,  3106,  3114
   3116,  3118,  7409,  7412,  7421,  7424,  7863,  7866,  7875,  7878,  8007,  8010,  8019,  8022,  8031,  8034
   8043,  8046,  8055,  8058, 11450, 11452, 11454, 11565, 11568, 11588, 11590, 11592, 11599, 11808, 11810, 11812
  11850, 11852, 11854, 12905, 12908, 12977, 12980
*Elset, elset=__PICKEDSURF156_S2_4, internal, instance=KUGELZAPFEN-1
     49,    50,    51,    52,   163,   164,   165,   166,   307,   308,   309,   310,   457,   458,   459,   460
    607,   608,   609,   610,   781,   782,   783,   784,   793,   794,   795,   796, 11602, 11603
*Surface, type=ELEMENT, name=_PICKEDSURF156, internal
   __PICKEDSURF156_S3_4, S3
   __PICKEDSURF156_S4_4, S4
   __PICKEDSURF156_S2_4, S2
*End Assembly
**
** MATERIALS
**
*Material, name=STEEL-42CRMOS4
*Conductivity
55.,   0.
55., 100.
54., 200.
41., 300.
39., 400.
36., 500.
34., 600.
29., 700.
```

```
29., 800.
29., 900.
29.,1000.
*Density
7780.,  0.
7750.,100.
7720.,200.
7700.,300.
7680.,400.
7630.,500.
7580.,600.
7540.,700.
*Elastic
  2.2e+11, 0.3,  0.
  2.1e+11, 0.3,100.
    2e+11, 0.3,200.
  1.9e+11, 0.3,300.
  1.8e+11, 0.3,400.
 1.75e+11, 0.3,500.
 1.65e+11, 0.3,600.
*Expansion
  1.05e-05,-100.
  1.15e-05,   0.
   1.2e-05, 100.
  1.25e-05, 200.
   1.3e-05, 300.
  1.33e-05, 400.
 1.375e-05, 500.
 1.425e-05, 600.
  1.48e-05, 700.
*Specific Heat
420.,  0.
460.,100.
490.,200.
520.,300.
540.,400.
560.,500.
580.,600.
620.,700.
**
*Material, name=WC-Co
*Conductivity
110.,
*Density
14500.,
*Elastic
5.8e+11, 0.3
*Expansion
5.4e-06,
*Specific heat
480.,
**
** BOUNDARY CONDITIONS
**
** Name: Disp-BC-1 Type: Symmetry/Antisymmetry/Encastre
*Boundary
_PICKEDSET158, ENCASTRE
**

** PREDEFINED FIELDS
**
** Name: Field-1   Type: Temperature
*Initial Conditions, type=TEMPERATURE
_PICKEDSET23, 20.
** ----------------------------------------------------------------
**
```

```
** STEP: Step-1
**
*Step, name=Step-1
*Coupled Temperature-Displacement, deltmx=300.
0.0334, 0.0334, 3.34e-07, 0.0334
**
** LOADS
**
** Name: SURFFLUX-1   Type: Surface heat flux
*Dsflux
_PICKEDSURF10, S, 8e+06
**
** INTERACTIONS
**
** Interaction: SURFFILM-1
*Sfilm
_PICKEDSURF10, F, 850., 552.
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history
*Element Output, elset=ELEMENTS
S, MISES
**
** HISTORY OUTPUT: H-Output-2
**
*Node Output, nset=NODES
NT,
**
*End Step
** ----------------------------------------------------------------
**
.....
.....

** STEP: Step-576
**
*Step, name=Step-576
*Coupled Temperature-Displacement, deltmx=300.
0.0334, 0.0334, 3.34e-07, 0.0334
**
** LOADS
**
** Name: SURFFLUX-575   Type: Surface heat flux
*Dsflux, op=NEW
** Name: SURFFLUX-576   Type: Surface heat flux
*Dsflux, op=NEW
_PICKEDSURF155, S, 8e+06
**
** INTERACTIONS
**
** Interaction: SURFFILM-575
*Sfilm, op=NEW
** Interaction: SURFFILM-576
*Sfilm, op=NEW
_PICKEDSURF155, F, 850., 552.
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
```

```
** FIELD OUTPUT: F-Output-576
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-1151
**
*Output, history
*Element Output, elset=ELEMENTS
S, MISES
**
** HISTORY OUTPUT: H-Output-1152
**
*Node Output, nset=NODES
NT,
*End Step
```