A Multi-Attribute Task Sequencing Optimization with Neighbourhoods Method to Improve Quality in Sustainable Industrial Processes

By

Eng. Massimiliano De Martino
Department of Industrial Engineering

Research Doctorate

Industrial Engineering – XXXII cycle

at

University of Naples Federico II

Coordinator of the PhD School: Prof. Eng. Michele Grassi
University of Naples Federico II, School of Engineering

Supervisor: Prof. Eng. Stanislao Patalano
University of Naples Federico II, School of Engineering

Supervisor: Prof. Eng. Francesco Timpone
University of Naples Federico II, School of Engineering

ABSTRACT

Industrial world faces what is commonly recognized as fourth industrial revolution, also known as ‘Industry 4.0’.

Industry 4.0 requires an overall production process optimization which deals with many manufacturing industrial aspects. Consider all sources of industrial optimization is not a trivial task. This dissertation deals with some of them which have a huge impact on Industry 4.0 fundamental, which are:

- Robot Path Planning Optimization
- Defects Impacts Reduction
- Workers’ condition improvement

Compared to great efforts made in the last years, a strategy for a production with no defects has not been completely inside industries companies, so quality process must rectify defects to avoid waste of resources.

Quality process is realized by human resources manually leading to waste of human resources and significant ergonomics risk factors as awkward posture in handling job task; moreover although people are very good at this task and sometimes even better than machines, they cannot work for long periods of time as their eyes get tired, muscles and tendon could be overloaded, compromising operator’s health and quality of the work.

Automate the quality control process will bring undoubted advantages in various aspects, as the improvement of the ergonomics workplace, optimization of the process as regards processing times, and quality of the production output.

The dissertation addresses the following questions:

How to define a new kind of workstation which can be applied to a wide range of industrial application to achieve:

- Improvement in ergonomic workspace aspects
- Reduction in defects number and impact

- Optimization in robot task execution by means of a methodology minimising computational time to enable dynamic robot programming in the case of multiple and coupled tasks' attributes

Starting from this question, following contributes have been addressed:

1. How to find optimal task sequence in case of not simple shapes with partial or total overlap using Euclidean distance as key metric
    a. Define an operator able to define when to pass through overlap zones
2. How to define an attribute to evaluate and forecast collision between robot and human, considering all robot parts, starting from multi-attribute approach to find optimized task sequencing
3. How to use computer-based method to develop a tool for image recognition on a reflecting surface.

CONTENTS

List of Figures

List of Tables

Nomenclature

A-OZPS	Augmented Overlap Zone Position Search
CETSP	Close-Enough Travelling Salesman Problem
CIH	Constricting Insertion Heuristic
CoI	Collision Index
CPPS	Cyber-Physical Production System
CPS	Cyber-Physical System
C-space	Configurations space
EE	End-Effector
GTSP	Generalise Travelling Salesman Problem
GTSPN	Generalise Travelling Salesman Problem with Neighbour
OPP	Overlap Position Problem
OLP	Off-line Programming
TCP	Tool Centre Point
TPP	Touring a sequence of Polygons Problem
TSP	Travelling Salesman Problem
T-space	Tasks space
TSPN	Travelling Salesman Problem with Neighbour

Chapter 1.

1.THESIS OVERVIEW

1.1 Introduction

Industrial world faces what is commonly recognized as fourth industrial revolution, also known as 'Industry 4.0'.To be able to understand how Industry 4.0 became today's buzzword, a look at its predecessors might give us a perspective on how this revolution in particular is different [1]. The following figure shows a timeline of the evolution of manufacturing and industrial sector:

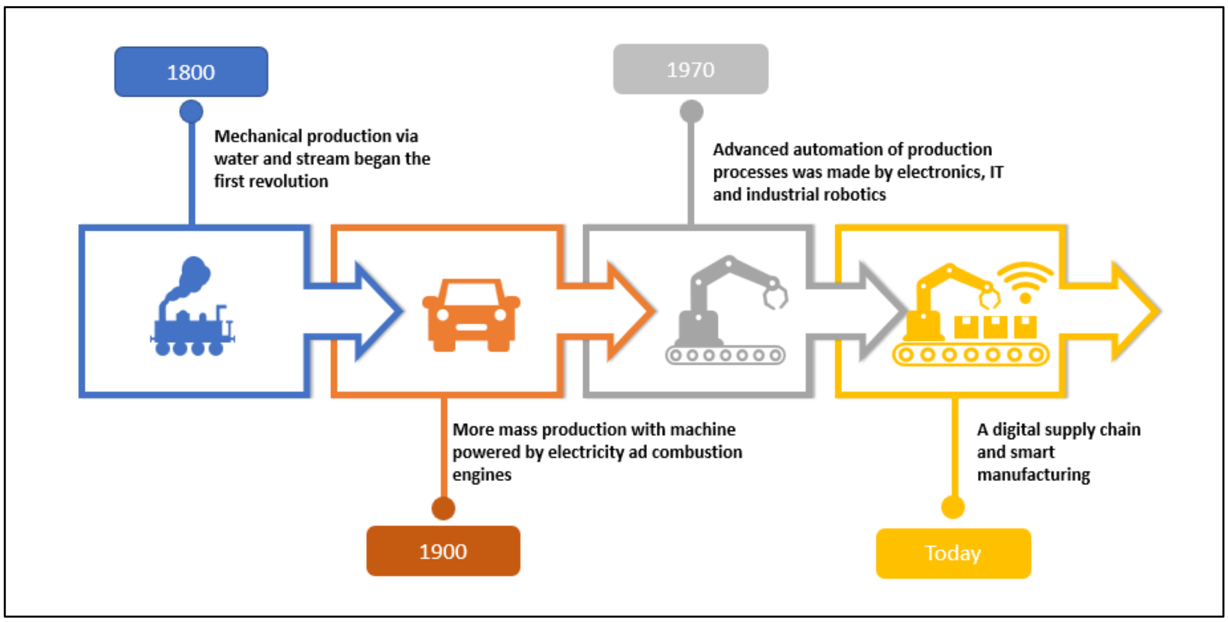

Figure 1 - Chronology of industrial revolutions

According to Lukac [2], following scheme can be followed:

- First industrial revolution began at the end of the 18th century and it was represented by mechanical production plants based on water and steam power
- Second industrial revolution started at the beginning of the 20th century with the symbol of mass labor production based on electrical energy
- Third industrial revolution began in the 1970s with the characteristic of automatic production based on electronics and internet technology

- Fourth industrial revolution, namely Industry 4.0, is ongoing, with the characteristics of cyber physical system (CPS) production, based on heterogeneous data and knowledge integration

Until now, there isn't a unique adopted definition of Industry 4.0**.**

According to the Consortium II. Fact Sheet (2013), Industry 4.0 is "the integration of complex physical machinery and devices with networked sensors and software, used to predict, control and plan for better business and societal outcomes [3].

Henning and Johannes (2013) define Industry 4.0 as "a new level of value chain organization and management across the lifecycle of products" [4].

Hermann, Pentek, and Otto (2016) define Industry 4.0 as "a collective term for technologies and concepts of value chain organization." They note that, within the modular structured Smart Factories of Industry 4.0, CPS monitor physical processes, create a virtual copy of the physical world and make decentralized decisions. They point out that over the IoT, CPS communicate and cooperate with each other and humans in real time, and that the Internet of Services (IoS), both internal and cross organizational services, is offered and utilized Industry 4.0 by participants of the value chain [5].

Otherwise, in this dissertation, it has been furnished a new and simpler Industry 4.0 definition, based on the following three questions:

- What are Industry 4.0 Instruments
- What are Object enabled by above mentioned Instruments
- What are Industry 4.0 Target

- ***What are Industry 4.0 Instruments***

Industry 4.0 is a transformation that is powered by nine foundational technologies which are:

- Augmented Realty
- Big Data and Analytics
- Autonomous Robot
- Simulation
- Horizontal & Vertical Integration System
- IOT
- Cyber-Security
- Cloud
- Additive Manufacturing

- ***What are Industry 4.0 Objects***

The industrial production will move from a physical process, with nine pillars support, to an integrated cyber-physical system, named CPS, which M. Broy et el [6] describes as embedded systems using sensors to capture data and act on physical processes by actuators over digital networks. If cyber-physical systems are applied in production systems, cyber-physical production systems arise [7]. They consist of autonomous elements and subsystems which are connected with each other and throughout the entire production system and are described by its three main characteristics: intelligence, connectedness, and responsiveness towards changes [8].Therefore, the industrial vision of future production predicts smart products which control and optimize themselves in their manufacturing process [9].

- ***What are Industry 4.0 targets***

Optimization regards:

- Productivity improvement
- Quality Improvement
- Safety Improvement
- Human-Robot Integration Improvement
- Ergonomic working condition improvement

And other aspects which must be optimized.

Below an image which summarizes our Industry 4.0 definition:

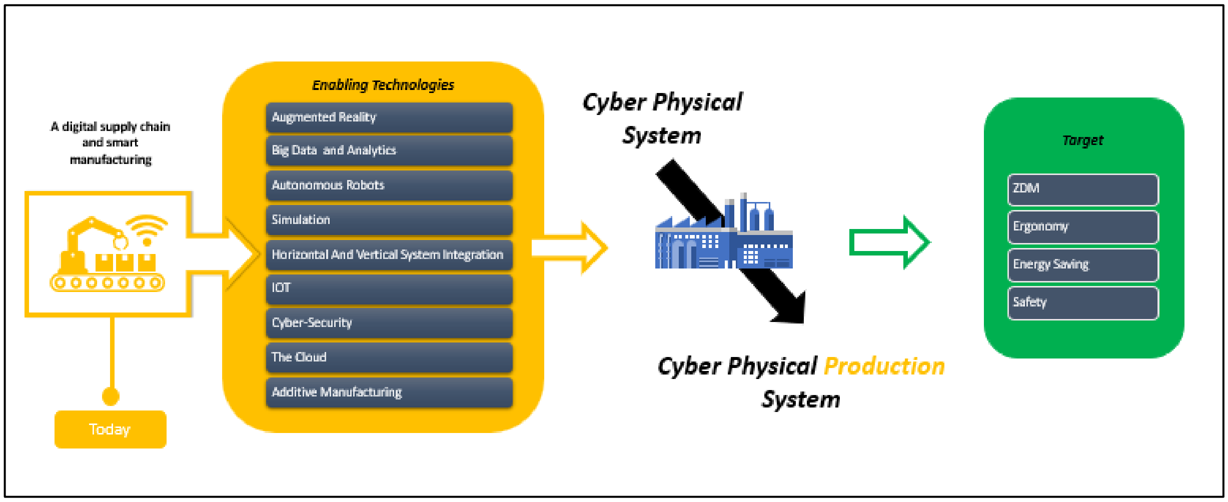

Figure 2 - Industry 4.0 Definition

1.2 Motivations

Industry 4.0 requires an overall production process optimization which deals with many manufacturing industrial aspects. Consider all sources of industrial optimization is not a trivial task, an idea about these sources can be made evaluating all industry 4.0 target listed in previous paragraph. This dissertation deals with some of them which have a huge impact on Industry 4.0 fundamental, which are:

- Robot Path Planning Optimization
- Defects Impacts Reduction
- Workers' condition improvement

Compared to great efforts made in the last years Zero Defects Manufacturing strategy has not been completely inside industries companies, so quality process must rectify defects to avoid waste of resources.

Quality process is realized by human resources manually leading to waste of human resources and significant ergonomics risk factors as awkward posture in handling job task, moreover although people are very good at this task and sometimes even better than machines, they cannot work for long periods of time as their eyes get tired, muscles and tendons could be overloaded, compromising operator's health and quality of the work.

Automating the quality control process will bring undoubted advantages in various aspects, as the improvement of the ergonomics workplace, optimization of the process as regards processing times, and quality of the production output.

1.2.1 Robot Path Planning

Process Optimization is identified as a strategy to reduce waste in term of resources, time, workflow, energy, material and consequently money.

The integration of available sensor and network technology with big data analytics and data mining have opened new way to reach process optimization.

In industry 4.0 context, optimize process involves optimize CPPS and so all elements included in it. In CPPS one of the most used and implemented machines are industrial robots, which are adopted by automotive, aerospace, electronics, appliance, chemical, plastics and rubber and material industry.

For instance, total worldwide stock operational industrial robots at the end of 2015 was about 1.6 million units (increased by 11% compared to 2014). The market value in 2015 was estimated to be US$35 billion (increased by 9% compared to 2014). The number of industrial robots deployed worldwide will increase to around 2.6 million units by 2019. That's about one million units more than in the record-breaking year of 2015 [10].

One of the largest industrial robot installations is the automotive sector. Automotive industries widely use multi-stage assembly systems consisting of multiple machine stations/stages to obtain the final product [11].As each stage is composed by one or more robots which repeat the same task multiple times, it is important to optimise robot path in order to minimise execution cost in terms of cycle time or energy [12].

In today best practice robot programs are developed off-line using CAD/CAM simulation suits, such as Delmia, RobCAD, etc, to compute collision free robot trajectory [13], which is classical off-line programming method, in which automatic path planning programming through simulation based on CAD model [14].

Another method to robot programming, is on-line robot task programming which deals with recorded robot movement in teaching mode and execute in production mode.

Defining the robot tasks entails multiple and coupled attributes, which can be summarised as follows:

- cycle time (A1) – to minimise execution time of multiple tasks
- energy consumption (A2) – to minimise energy consumption in robot transitions
- path length (A3) – to minimise length of the end-effector's path

Either (1), (2) or (3) are strongly correlated to:

- pose quality (A4) – accuracy and repeatability of the end-effector pose (both position and orientation)
- collision (A5) – robot movement must be collision free and avoid
- robot placement (A6) – robot placement (both position and orientation) with respect to the workpiece. Robot placement is directly related to the *accessibility of tasks.* For instance, it may happen that the same task though feasible in terms of collision and pose quality, can be executed by multiple paths (multiple accessible paths). As consequence, there is no guarantee of reaching the minimal cycle time attribute

Robot programming, is usually decomposed in four subproblems/ steps [15]:

- *task planning* – Robot tasks are described through (hyper) volume or regions (Task region (TR)) in a pre-defined coordinate system. That volume/regions corresponds to the envelop of the all Tool Centre Point (TCP) poses (position (X_{EE}, Y_{EE}, Z_{EE}) and orientation (α, β, γ)).
- task sequencing - Sequence of tasks is generated according to pre-defined attributes. Optimal sequence is usually computed based on TCP's positions; and, it is solved in Cartesian space, also called T-space.
- path planning - Path is generated according to task sequence and attributes. The route which leads from one robot configuration to another is named path. A path is the locus of TCP points when it moves throughout configurations, for given sequence of tasks. Path planning allows to compute robot configuration for each TCP pose and the sequence of configurations that moves the robot among configurations. The computation is performed into the configuration space of the robot, also called C-space.
- motion planning - Robot movements are generated to follow the path.

Those steps are strictly coupled as robot trajectory is strongly affected by Step [2] and Step [3]. There are few approaches that allow to generate automatic robot programming. However, none of them use a complete integration of task sequencing and path planning. As no robot information is involved in task sequencing, no attributes can be directly computed in T-space (neither A1 as it involves the information about the robot configurations). As consequence, no feasible solutions are guaranteed in T-space. Therefore, these single-attribute methods require multiple iterations between task and path to converge to a (near) optimal solution which can be far from the optimal one.

1.2.2 Defects Impact Reduction

Another Industry 4.0 key strategy is Zero Defects Manufacturing whose target is to reduce both the number and impact of product defects by improving "right-first-time" capability, with minimum waste of material and resources, including time, energy and workforce.

Compared to great efforts made in the last years Zero Defects Manufacturing strategy has not been completely inside industries companies, so quality process must rectify defects to avoid waste of resources.

Quality process is realized by human resources manually leading to waste of human resources and significant ergonomics risk factors as awkward posture in handling job task.

1.2.3 Workspace & Ergonomic Improvement

Modern manufacturing plants are based on multi-stage processes [16] by alternating humans and robotic systems.

Researchers [17] [18] claim that workstations could be more productive by combining:

- human intelligence,
- flexibility and adaptability of the manufacturing line,
- strength, endurance and accuracy of robots.

The combination of human beneficial characteristics with modern robots opens huge possibilities to simultaneously increase productivity, reduce ergonomically bad work postures [19] and increase the perceived well-being on shift [20].The combination of human-robot skill seems to be the key approach to achieve a full manufacturing plant improvement.

1.3 Goals of Research

The present dissertation, starting from the state of the art, provides an evolutionary manufacturing quality control process to enable a smart and sustainable production by using image recognition tool , optimizing ergonomic workspace aspects, which makes use of a robot whose task execution has been optimized by means of a specific developed methodology able to find near-optimum solution, by minimising computational time to enable dynamic robot programming in the case of multiple and coupled tasks' attributes.

In same industrial scenarios, quality process to visualize and eliminate defects, is characterized by the presence of an operator which recognizes defect and by means of a special tool corrects that area.

Below, it has been reported a scheme which represents this process:

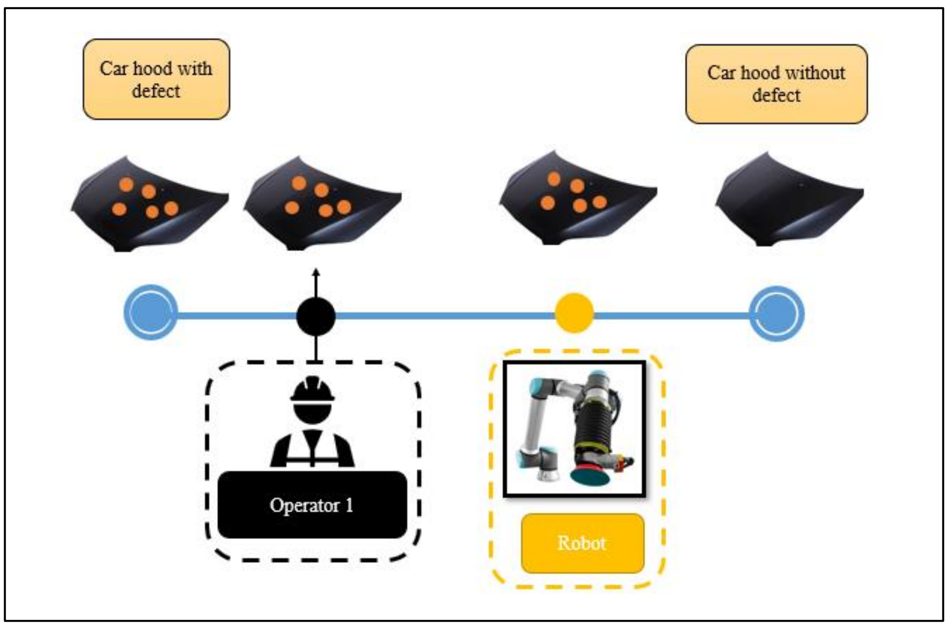

Figure 3 - Industrial Line Workflow

Target is to automatize and optimize this process, reducing physical load on the operator optimizing overall process.

Particularly a workspace has been defined, in which:

- Human Task: consists in marking object with different marker type whose position enables x.y,z coordinate-position that has to be elaborated, while marker typology defines parameter elaboration which robot has to achieve.

- Example: An object needs to be rectified in a particular position, so operator marks that position with an x-marker which will be recognized by an image processing tool in position and typology. Marker typology and coordinate will be sent to a Robot which process this information enabling process, whose parameter depends from marker typology (which is x but could have been another typology which could have enabled other process parameters), going to rectify objects.

- Robot Task : Robot by using a developed image-processing techniques recognize marker in position and typology enabling a task execution optimized by means of a specific developed methodology able to find near-optimum solution, by minimising computational time to enable dynamic robot programming in the case of multiple and coupled tasks' attributes.

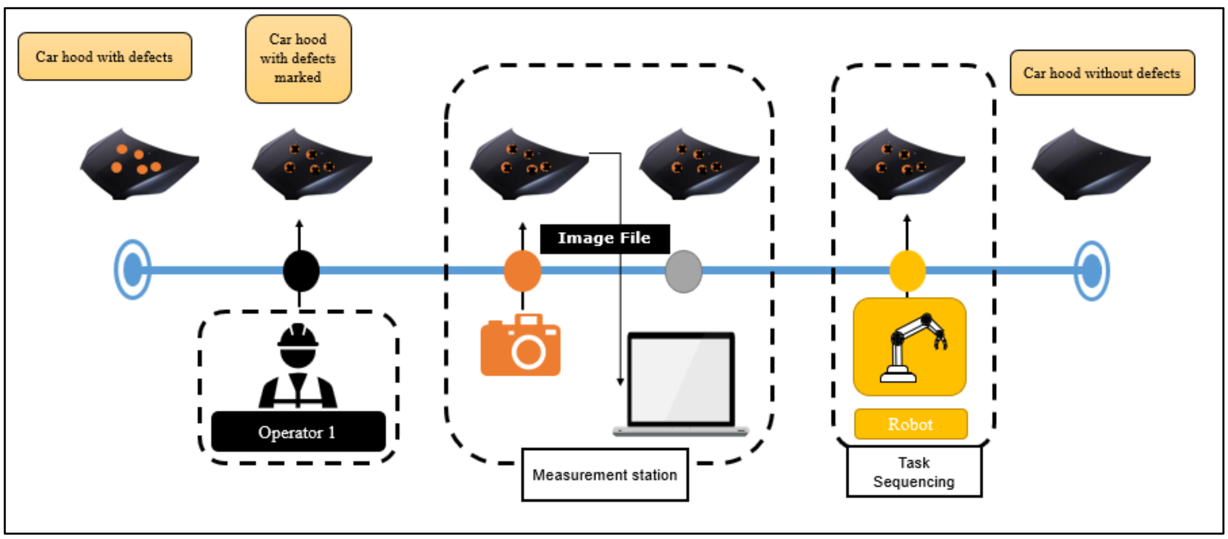

Figure 4 - Proposed Industrial Line Workflow

It's important for safety reasons to underline that cell proposed is not a collaborative one, considering that in a collaborative cell no separation between human and robot has used, while this dissertation deals with a cell in which human and robot 'collaborates' without sharing workspace.

1.4 Research Question

The dissertation addresses the following questions:

How to define a quality control scenario which simultaneously increases productivity, reduces ergonomically bad work postures, and increases the perceived well-being on shift, achieving a full manufacturing plant improvement.

Starting from this question, following questions have been addressed:

1. How to find optimal task sequence in case of not simple shapes with partial or total overlap using Euclidean distance as key metric
    a. Define an operator able to define when to pass through overlap zone
2. How to define attribute to evaluate and forecast collision between robot and human, considering all robot parts, starting from multi-attribute approach to find optimized task sequencing
3. How to use computer-based method to develop a tool for image recognition on a reflecting surface

1.5 Contribution

The dissertation introduces following contributions:

A quality control scenario which simultaneously increases productivity, reduces ergonomically bad work postures, and increases the perceived well-being on shift, achieving a full manufacturing plant improvement.

Starting from these elements, following contributions have been addressed:

- This dissertation proposes a new method to solve Travelling Salesman Problem with Neighborhood in case of complex areas in partial or total overlap
    - An operator able to recognize when to pass through overlap zones
- An attribute to define and forecast overall collision for robot
- A computer-based method to recognize marker on a reflecting surface

1.6 Industrial Application

Research Project

Research project is born inside ‘Industry 4.0’ context and aims to create a strategy to optimize overall process leading process to be more ergonomic and ‘green’. Therefore, the research activity, aims to develop, implement and test an in-process quality improvement methodology for an industrial process. Focus of this activity is on hood surface manufacturing defects. Nowadays, process to visualize and eliminate this kind of defects, is characterized by the presence of an operator which recognizes defect and by means of a special operator corrects that area.

Target is to automatize and optimize this process, reducing physical load on the operator optimizing overall process. Particularly, an innovative approach has been proposed, which consists of the following phases:

1. The operator recognizes the area that needs to be rectified, marking it with a marker
2. An image recognition tool recognizes markers in position and typology
3. The position of the areas to rectify, affected by defects, is translated into Cartesian coordinates, through a scanning system
4. The defects correction system, automatically, crosses the various areas, carrying out the rectification operation, through a path overall optimized

Attended Industrial Impact

- Defects reduction
- Ergonomic Improvement

1.7 Thesis Organization

The dissertation is arranged as follow:

- Chapter 2 - It lays down the common terms and concepts used to define innovative workspace scenarios.
- Chapter 3 - It reviews state of art and identifies current trends and limitations
- Chapter 4 - It presents the OZPS for TSPN and its comparison with well-known methods
- Chapter 5 - It introduces the multi-attribute method for robotic task sequencing, with a new attribute to evaluate overall collision
- Chapter 6 – It lays down industrial application
- Chapter 7 - It draws final remarks and potential future development.

Chapter 2.

2.BACKGROUND

This section defines common terms and concepts used for industrial robotics, image recognition method and ergonomics aspects involved in this dissertation.

This chapter has arranged as follow:

- First Section - Robot Background
- Second Section - Image Processing Technique Background
- Third Section - Ergonomics Background

2.1 First Section – Robot Background

The common industrial robot is the articulated/anthropomorphic robot belonging to the manipulating type. In this dissertation, we focus on the articulated robot, but any assumptions and results can be applied to any industrial robot type.

An industrial robotic arm is usually composed by a base, a sequence of links (rigid bodies) and an end-effector (EE) connected by kinematic pairs (joints), see Figure 5.

These components define the kinematic chain (sequence of links connecting the two ends of the chain: base and end-effector).

The common industrial robot has got six revolute joints: ϑ_1 – waist; ϑ_2– shoulder; ϑ_3– elbow; ϑ_4 – wrist rotation; ϑ_5 – wrist bend; ϑ_6 – flange rotation.

The number of joints determines the number of Degrees of Freedom (DOFs).

We define the TCP as the EE point of interest to be tracked; it can or cannot belong to the EE tool as well as exist geometrically because defined by functional parameters. The TCP position (xEE, yEE, zEE) and orientation (α, β, γ) compose the pose of the TCP.

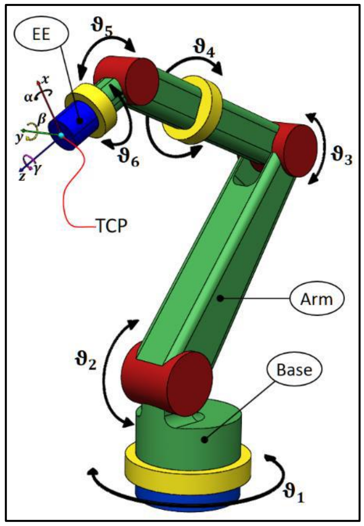

Figure 5 - Robot components: base, arms and end-effector

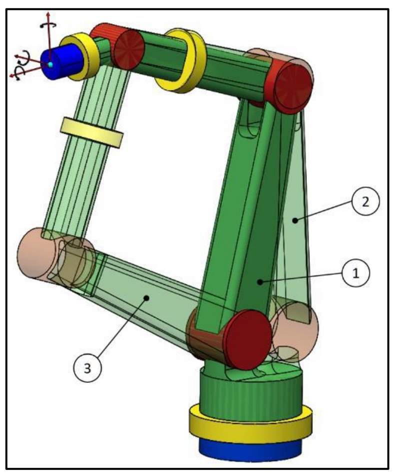

Figure 6 - Example of three configurations of a redundant robot

The main coordinate reference frame is the robot triad positioned in the "base". Henceforth, we will refer to it as robot reference. Starting from the robot reference, a pose can be defined in two ways, in two different space:

- in the task space or Cartesian space (T-space) by coordinates $(x, y, z, \alpha, \beta, \gamma)$
- in the configuration space or joint space (C-space) by joint configurations $(\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5, \vartheta_6)$

T-space is only related to the EE (it defines the pose) while C-space introduces information on the robot and its links (it defines the robot configuration). Therefore, each pose can be reached by multiple robot configurations (a typical industrial robot, with 6 DOFs, could arrange up to 8 configurations for a defined pose – see Figure 6).

Moving from C-space to T-space is called *forward kinematics* as to each robot configuration correspond only one pose. Conversely, moving from T-space to C-space gives multiple solutions and it is called *inverse kinematics*.

Robots perform any specific job following a path which robot movements correspond to. A job consists of several tasks. There are two types of paths: (1) task or effective path which is the robot path to accomplish that specific task; (2) supporting or inter-tasks path which is the robot path that connects tasks [21]. Therefore, a job path consists of task and inter-tasks paths.

The route that leads from a robot configuration to another is named path. A path is the geometrical description of the robot motion, i.e. locus of points; a trajectory is a path with a motion law [22]. Here, we will refer to the path as the locus of TCP points over time. All TCP accessible points are contained in a volume named "robot workspace".

2.2 Commercial Tools

This Section introduces the existent commercial solutions, hardware and software, for the task sequencing and path planning.

2.2.1 Robot Studio

Robot Studio is an ABB's PC-based robot programming software (see Figure 7). The ability to program a robot in the virtual world before it operates in the real world has dramatically changed the way companies and individuals think about programming robots. Over the last decade it has become an increasingly popular way to test robot operation before a mistake on the factory floor results in damage, stoppage and/or loss of money. The traditional method of programming robots, using a Flex Pendant attached to the robot controller, works well for some tasks, but robots have been placed into ever more intricate and complicated operations and even the most skilled human programmer staring at a screen full of countless lines of code would be hard pressed to accomplish.

Once the program is completed in the virtual world it can simply be downloaded straight to the robot controller in the real world, and as long as everything in the real world is set up exactly as it was in the virtual world, the program will run exactly like it did on the PC.

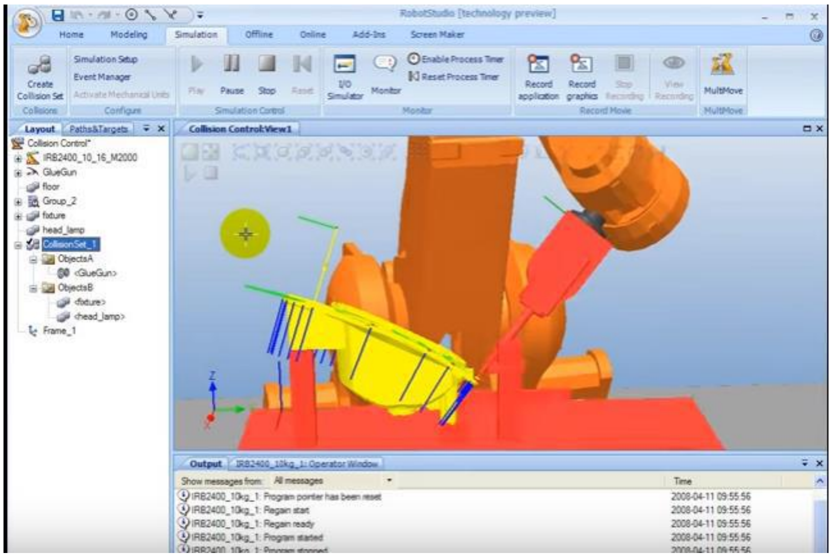

Figure 7 - Robot Studio collision detection

Robot Studio allows to check reachability, avoid collision and detect singular issues. Robot Studio has several functions for testing how robots reach and move to targets. They are useful both for finding the optimal layout when building a station and during programming. With Robot Studio we can detect and log collisions between objects in the station. A collision set contains two groups, Objects A and Objects B, in which we place the objects to detect any collisions between them. When any object in Objects A collides with any object in Objects B, the collision is displayed in the graphical view and logged in the output window.

After having created a collision set, Robot Studio checks the positions of all objects and detect when any object in Objects A collides with any object in Objects B. Activation of detection and display of collisions depend on how the collision detection is set up. If the collision set is active, Robot Studio checks the positions of the objects in the groups and indicate any collision between them according to the current color settings.

After detecting a collision, we can modify the path of the robot's tool and run the program again to check whether there are collisions also with the new setup. If now collisions are avoided, this new path is saved as a collision-free path for the robot.

2.2.2 DELMIA

DELMIA is the Digital Manufacturing and Production Solution of Dassault Systèmes, optimizing production systems and processes (see Figure 8). DELMIA Device Task Definition (DTD) delivers the capability to program and simulate forward kinematic mechanical devices, ranging from simple clamps to complex lift-assist mechanisms. It also provides the ability to manage multiple devices, integrate them within the V6 3D work-cell layout, and perform feasibility studies. Each device is individually programmed with tasks that are sequenced and simulated to eliminate any interference and obtain optimal cycle times.

DELMIA Device Task Definition provides an interactive V6 3D environment which allows users to define the tasks for each device in the context of the shop floor. Users are able to sequence the tasks of each individually programmed device in order to achieve synchronized motion between the devices in the work-cell.

Single or multiple device tasks can be simulated in 3D to locate and correct any interferences or collisions in the work-cell. Users can evaluate and optimize device activities to achieve desired cycle times.

The robot programmer can automatically optimize the robot's motion by computing standard motion parameters such as turn numbers, configuration, gantry and rail values along a robot trajectory [23]. It also provides tools which optimize cycle time and reach to create a collision-free path.

Moreover, DELMIA Robotics Path Planner (RPP) provides tools for automatically computing collision-free and optimized trajectories for industrial robots. Robotics Path Planner provides a highly efficient command for automatic collision-free path planning to facilitate robotic feasibility studies and off-line programming.

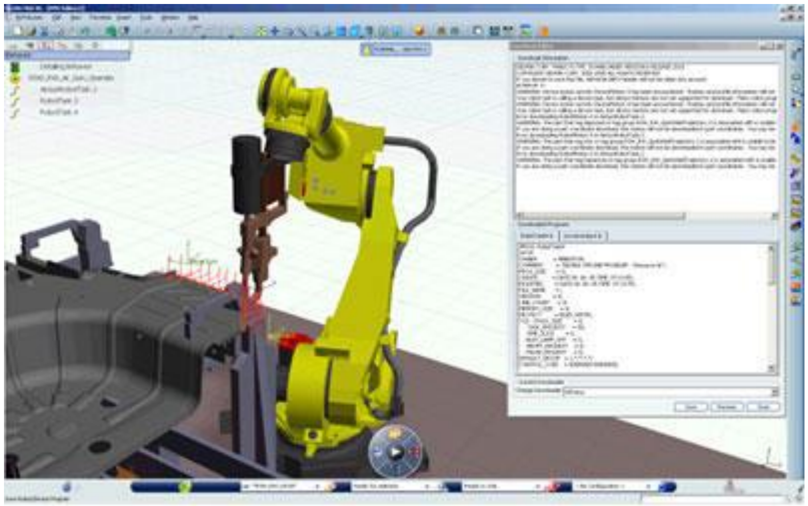

Figure 8 - Example of a robot simulation with DELMIA

Cycle times are minimized by RPP by optimizing automatically new trajectories calculated to fit exactly each new project. Path of the tool center point frame for linear motion, or path in the configuration space for joint motion, is minimized with better cycle times than can be achieved by other methods. By applying RPP to DELMIA robot task motion activity, RPP creates collision-free and optimized DELMIA motion activity.

Robotics Path Planner automatically transforms a robot task, updating a motion activity with potential collisions between the robot and its environment into a collision-free trajectory. When the robot and its environment need to be modified and updated the previously defined task can be automatically recomputed providing a fast versioning check and task update. Once a trajectory is computed, it can be optimised to reduce the robot cycle time. The non-trivial task of robot configuration space optimisation is achieved automatically by RPP. The resulting joint motion interpolation yields a faster motion and a lower risk of singularity.

2.2.3 RobCAD

RobCAD is a Siemens PLM Software for robotic work-cells verification and off-line programming. Tecnomatix RobCAD software (see Figure 9) enables the design, simulation, optimisation, analysis and off-line programming of multi-device robotic and automated manufacturing processes in the context of product and production resources. It provides a concurrent engineering platform to optimise processes and calculate cycle times. With RobCAD, you can design life-like, full-action mock-ups of complete manufacturing cells and systems. RobCAD enables manufacturers to flawlessly introduce automated processes by allowing manufacturing engineers to virtually validate automation concepts upfront.

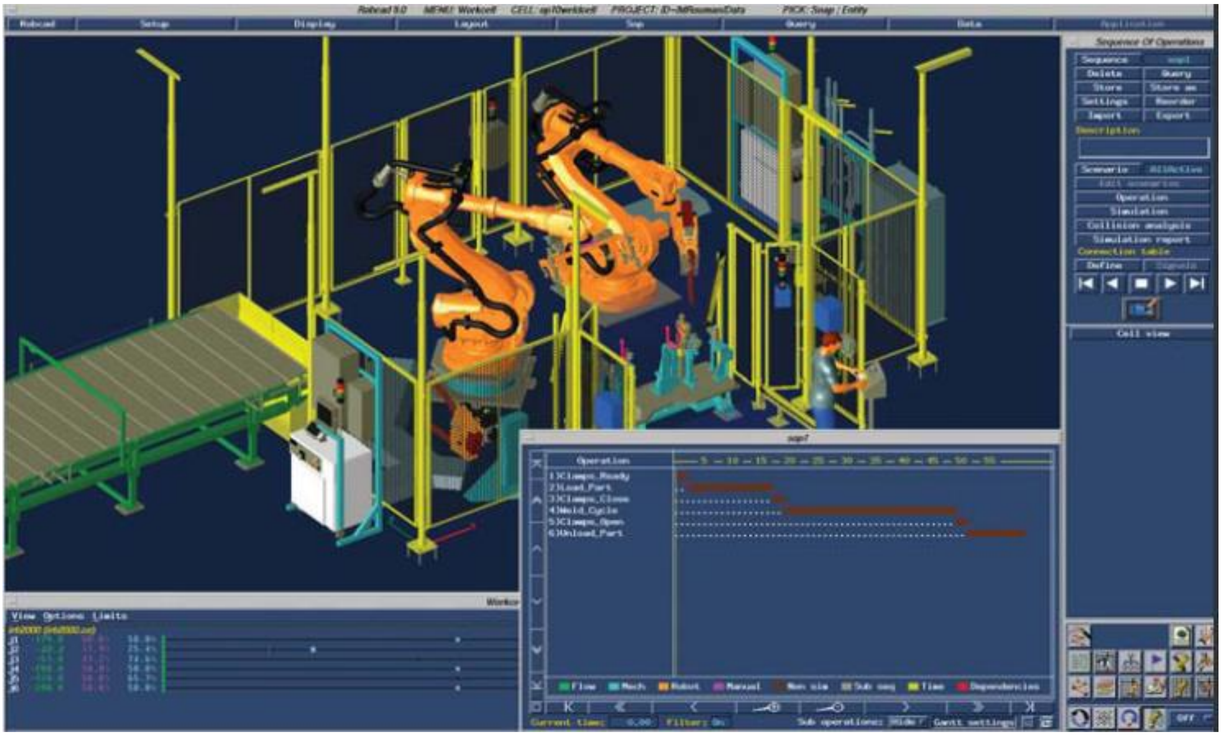

Figure 9 - Robcad software for off-line programming

RobCAD generates configurable motion plans based on the controller features. It allows calculation of cycle times, analysis of real-time performance and saves testing time. The RRS (Realistic Robot Simulation), which is based on using the real controller motion planning software, offers extremely accurate cycle time calculation.

Robcad off-line programming enables accurate simulations of robot motion sequences and the delivery of machine programs to the shop floor. Moreover, RobCAD can dynamically detect collisions during robot simulation and motion, preventing costly damages to equipment. In fact, for automatic path planning**,** RobCAD generates collision-free robot and part assembly paths by using automatic path planning technology.

2.2.4 Kineo

Kineo is a Siemens PLM software. Kineo solutions include advanced software components and standalone applications for automatic motion/path planning and collision detection (see Figure 10). Kineo products satisfy a wide range of virtual prototyping requirements, from assembly or disassembly clearance validation to collision-free robot applications. In modern end-user CAD, CAM, CAE, 3D digital mock-up and robotics systems, these productivity tools help automate path planning and clash detection factors which in turn save customers time, costs and resources.

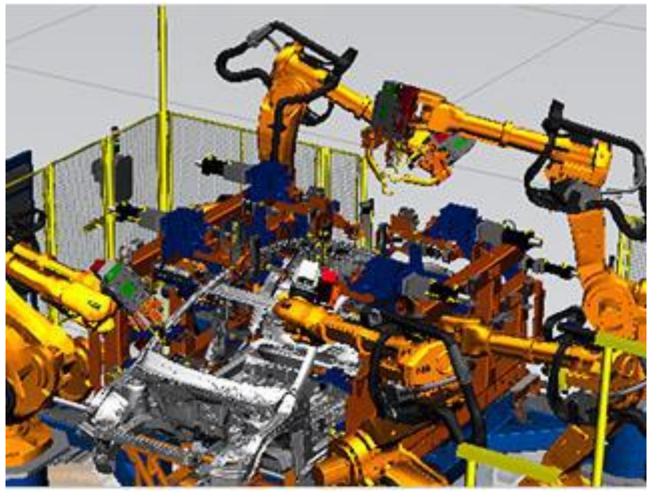

Figure 10 - Example Kineo software

Kineo components provide leading path planning and collision detection tools for CAD/CAM applications. In autonomous robotics, Kineo-based path planning and collision detection maximize the operational efficiency of robotic systems.

Kineo Collision Detector allows to check spatial interferences, or collisions, between hierarchical assemblies of triangle mesh surfaces, or polyhedrons. Kineo Collision Detector can be used to perform different kinds of interference analyses, including:

- Exhaustive Boolean check to determine if analyzed objects are colliding and, if so, reports every pair of colliding triangles.
- Exact distance to determine if analyzed objects are colliding and, if not, reports the shortest distance between them.
- Penetration to determine if objects are colliding and, if so, reports a translation vector that suppresses the collision.

Every object can have its own tolerance value, which is the size of a clearance zone added around the object. Kineo Collision Detector is optimised for low response times, with a built-in multithread capability, enabling the best hardware performance. Thanks to its stateless, thread-safe mode, Kineo Collision Detector is suitable to run different tests over the same scene in simultaneous threads. This offers new possibilities to multithreaded applications aimed at performance and reactiveness. Instead of waiting for tests to return, the process can span new tests in new threads and use all available computing power.

2.2.5 Robot & Matlab & Solidworks

An alternative to Robot commercial software solution, is to create a robot model in Solidworks and then translate it in a Matlab Simulink model by means of a Matlab Plug-In. Matlab Simulink model must be provided by m-file which implements all aspects that it's necessary to test, perform and validate.

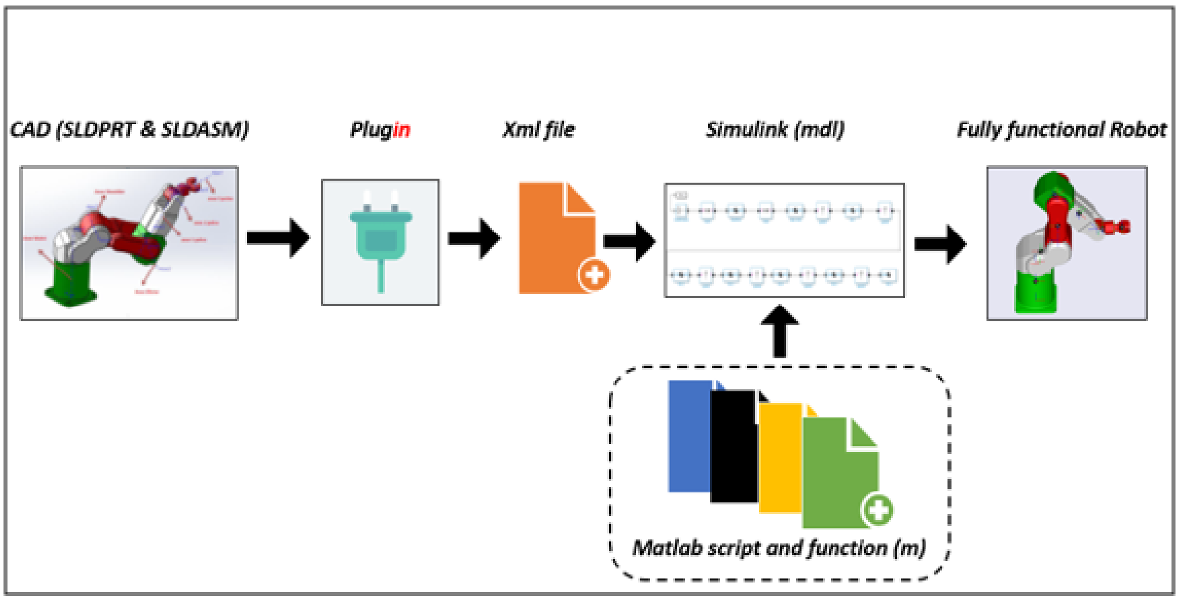

Figure 11 - Workflow to Translate a Robot CAD in a Matlab Simulink Model

By using this approach is possible to formalize by writing code all aspects that need to be examined. As an example, it's possible to:

- Create algorithm to check reachability, avid collision and detect singular issues
- Simulate forward kinematic
- Optimize Trajectories
- Calculate of cycle times, analysis of real-time performance and saves testing

Although all these aspects are available in the different software listed in the previous paragraphs, advantage to develop these tools in Matlab is that it's possible to a have a complete control of formula used to model problem of interest.

Moreover, although existing solutions for OLP and path planning are standard toolkits in modern design architectures, developed solutions are not able to modify and choose a new path automatically and to dynamically react to fluctuations and changes, has happening in real-life production systems [12].

2.3 Second Section – Image Processing Background

2.3.1 Definition

Due to the advent of computer technology image-processing techniques have become increasingly important in a wide variety of applications [24].

Image segmentation is a classic technique in digital image processing which starting from characteristics of the pixels in the image to partition an image into multiple regions.

Image segmentation could involve separating foreground from background, or clustering regions of pixels based on similarities in colour or shape [25], more generally segmentation is the process that subdivides an image into its constituent parts or objects.

The level to which this subdivision is carried out depends on the problem being solved, i.e., the segmentation should stop when the objects of interest in an application have been isolated [24].

2.3.2 How Image Segmentation Works

Image segmentation involves converting an image into a collection of regions of pixels that are represented by a mask or a labelled image. By dividing an image into segments, you can process only the important segments of the image instead of processing the entire image [25].

A common technique is to look for abrupt discontinuities in pixel values, which typically indicate edges that define a region [25].

Figure 12 - Using thresholding to convert to a binary image to improve the legibility

Another common approach is to detect similarities in the regions of an image. Some techniques that follow this approach are region growing, clustering, and thresholding [25].

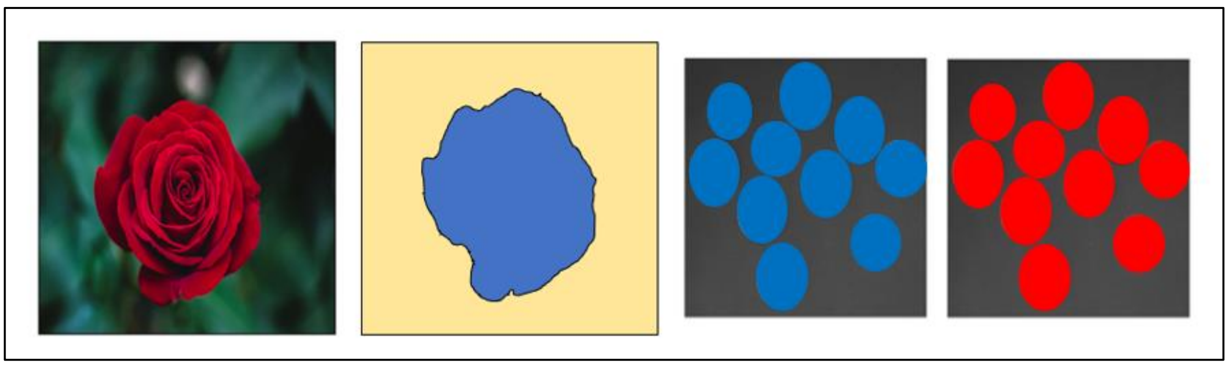

Figure 13 - Segmenting regions based on color values, shapes

2.4 Third Section – Ergonomics Background

2.4.1 Ergonomics Risk Factors (ERF)

The increasing number of injuries caused by repetitive motion, excessive force and awkward postures; ergonomics has become a critical factor in workplace safety [26].

Risk and risk factors are common concepts used in safety and applied ergonomics literature [26].

Risk includes a component of how likely or what the probability of an event is and the seriousness of the consequence or what the severity is if something does occur. Risk is often defined on how many injuries or accidents resulted for a given exposure [26].

Risk factors are defined as actions or conditions that increase the likelihood of injury to the musculoskeletal system [26].

Ergonomics Risk Factors (ERF) is situations that exist or created intentionally or unintentionally that could or might contribute to results contravene or against the principles or philosophy of ergonomics that could or might harmful to the health and well-being of workers or users at work or after work [27].

The principal ERF are repetition, force, awkward posture, vibration, contact stress, static loading and extreme temperature [26].

Long-term exposure to risk factors will reduce the quality of life, resulting in Musculoskeletal disorder (MSD). Musculoskeletal disorder (MSD) is a condition or disorder that involves the muscles, nerves, tendons, ligaments, joints, cartilage, or spinal discs [26].

According to Yelin et al. [28] , 90% of disabled older workers had MSDs. The treatment of the MSDs problems will cost tens of billions of dollars as stated by Praemer et al. [29].

2.5 Summary

In this chapter an overview about three main themes has been explored.

Off-line programming is useful tool for saving money and time when designing a new work cell.

Commercial and academic/open source robotic software for OLP mostly focuses on the motion planning optimisation, and neglect optimisation of the task sequencing. All robot targets are programmed by the operator that should simultaneously consider optimal EE placements, reachability and sequence constraints.

Although existing solutions for OLP and path planning are standard toolkits in modern design architectures, developed solutions are not able to modify and choose a new path automatically and to dynamically react to fluctuations and changes, has happening in real-life production systems[[12].

Moreover, an overview about common terms and concepts has been furnished about Image Segmentation and Ergonomics aspects widely used in this dissertation.

Chapter 3.

3.REVIEW OF PREVIOUS WORK

This chapter highlights research key topics. It analyses related works illustrating their characteristics and limitations in Robotics application.

3.1 Introduction

Although path optimisation concerns both task and inter-tasks paths, they can be computed in separate way considering that the input and output configuration of the effective path can be considered as two different configurations in the supporting path. Hereinafter, supporting path will be referred as path.

The robot paths have to efficiently avoid collisions and unnecessary movements. The path planning problem aims to find the sequence of the robot configurations to accomplish the job. Robot systems can reach a given location assuming several configurations, ideally, infinite; therefore, the sequence of tasks is affected by multiple attribute and objective function; optimisation cannot neglect them.

3.2 Robot Programming

Robot programming usually consists of four steps: (1) task planning; (2) task sequencing; (3) path planning and (4) motion planning. Task (2) and (3-4) are strongly coupled. For example, the optimal robot path, which is function of the robot kinematics, relies on the pre-defined schedule of tasks, whose sequencing is computed based on the assumption that the travelling "cost" from one task to the next is only driven by the Euclidean distance in Cartesian space. Current methods tends to decouple the problem and sequentially compute the task sequencing in the T-space, and then compute the robot path by solving the inverse kinematics in the C-space, while a good approach could be select robot configuration considering simultaneously the attributes assessing reachability, minimising configuration transition, avoiding collision and evaluating pose performance.

3.3 Robotic Task Sequencing

Sequence of robot tasks are classical tour-searching combinatorial problems modelled as the Travelling Salesman Problem (TSP) [30] [31].

The well-known Traveling Salesman Problem (TSP) is formalized as follows: Given a weighted undirected graph $G = (V; E)$, where V is a set of n vertices and E is a set of edges. The objective is to find a minimal-cost cyclic tour $T = (v_1; ....; v_{n+1})$ that visits all vertices only once returning to the original $(v_1 = v_{n+1})$ [32] . In this case, each task is formulated as point. Otherwise TSP can provide only an initial tour approximation, because of:

- Many tasks do not require strong determinism and often allow a certain degree of freedom [32]
- Robot can arrange multiple configurations for the same placement as well as multiple position to accomplish the same operation

So, it is more realistic formulate the problem with a set of points where each of them corresponds to one robot configurations or EE placements. This formulation is named Generalised TSP (GTSP) [33]. GTSP is obtained substituting each single point with a cluster of points, the shortest path visits one point for each cluster.

GTSP is applied in [34] where the authors generate clusters by sampling a set of configurations for each location. This solution is always limited among the sampled points.

An interesting link between GTSP and robotics was established in [35], where the authors introduce the multi-goal path planning problem (MTP) where a cluster of poses are modelled as a cluster of points.

3.3.1 Traveling Salesman Problem with Neighbourhoods: Introduction

Despite GTSP partially overcome the limitations on multi-inverse kinematics (multi-IK), it requires a certain discretization that means errors in the final solution and a partial task volume representation. GTSP solution quality improves with augmenting point numbers increasing computational complexity.

To obtain an acceptable optimized solution the search space has to be continuous, i.e. a region. When the points change in regions (examples: areas in 2D; volumes in 3D) the TSP becomes Traveling Salesman Problem with Neighbourhoods (TSPN) [36] where each region is visited once.

The well-known Traveling Salesman Problem (TSPN) is formalized as follows: Given a set of n polygons $A = (A_1; ....; A_n)$, find a minimal-cost cyclic tour $T = (p_1; ...; p_{n+1})$, such that it visits A_i in the point p_i and $p_1 = p_{n+1}$.

Gentilini et al. [37] use TSPN to formalise the path optimisation but they neglect the multiple inverse kinematics. Some researches formalise the multiple configurations for each location goal as a cluster of regions. This increases the computational complexity.

As for TSP, when the region is substituted by a cluster of regions the problem becomes generalized: Generalized TSPN (GTSPN) [38].

Vicencio et al. [39] use GTSPN to optimize a six-rotor path planning to overcome the limitations of the TSPN formalization.

TSPN solvers are not limited by the space rather by currently formulation that cannot allow to integrate path and task [40].

3.3.2 Traveling Salesman Problem with Neighbourhoods: Modelling [1]

TSPN is a generalization of TSP where areas have to be visited instead of points.

TSPN aims to find the shortest path via regions visiting each region once. This formulation allows to optimize both sequence and location within neighborhoods.

Starting from the literature, they have been recognized for TSPN two main features which are:

- Is there Overlap
  - Yes
  - No
- What kind of shapes have been investigated
  - Complex

[1] In this paragraph will be used acronyms as TSPN-S, TSPN-C, TSPN-WO, TSPN-S-WO, TSPN-C-WO which are used to simplify problem investigation, and so must not be considered as reference.

- Simple (Ellipses, Circumferences)

To simplify problem investigation following scheme will be adopted:

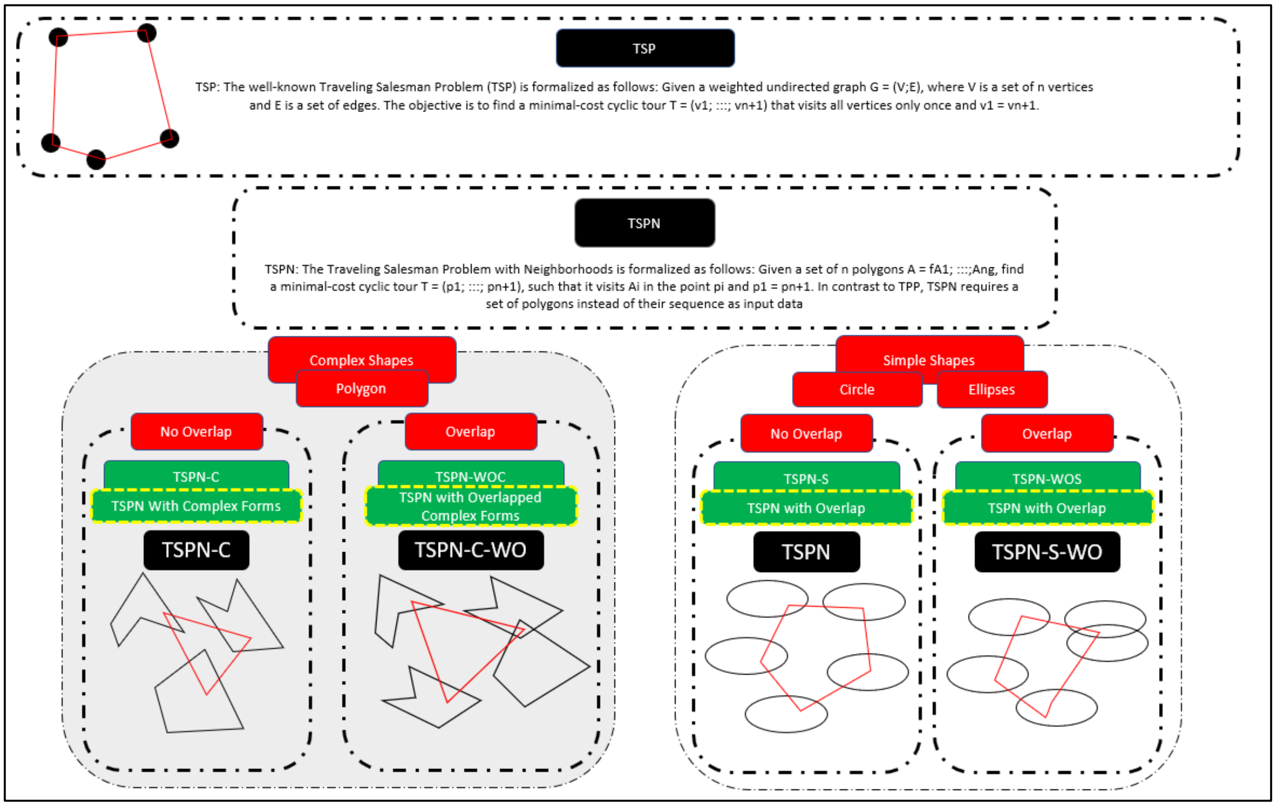

Figure 14 - Salesman Problem Overview

Starting from the top, first version of Salesman Problem is TSP (Travelling Salesman Problem) formalized as follow:

Given a weighted undirected graph $G = (V; E)$, where V is a set of n vertices and E is a set of edges. The objective is to find a minimal-cost cyclic tour $T = (v_1; \ldots; v_{n+1})$ that visits all vertices only once returning to the original $(v_1 = v_{n+1})$ [32].

If points have been replaced with areas TSPN arise (Travelling Salesman Problem with Neighborhouds) formalized as follow:

Given a set of n polygons $A = \{A_1, \ldots, A_n\}$, find a minimal-cost cyclic tour $T = (p_1; \ldots; p_{n+1})$, such that it visits A_i in the point $p_i = p_{n+1}$ [32].

If Areas are complex and not in overlap, TSPN-C arises:

The Traveling Salesman Problem - Complex (TSPN - C) is formalized as follows: Given a set of n complex areas (Polygon) $A = (A_1; ....; A_n)$, find a minimal-cost cyclic tour $T = (p_1; ...; p_{n+1})$, such that it visits A_i in the point p_i and $p_1 = p_{n+1}$.

If Areas are complex and with overlap, TSPN-C-WO arises:

The Traveling Salesman Problem – Complex – With Overlap (TSPN-C-WO) is formalized as follows: Given a set of n complex areas(polygons)$A = (A_1; ....; A_n)$ with partial or total overlap, find a minimal-cost cyclic tour $T = (p_1; ...; p_{n+1})$, such that it visits A_i in the point p_i and $p_1 = p_{n+1}$.

If Areas are simple and not in overlap, TSPN-S arises:

The Traveling Salesman Problem - Simple (TSPN) is formalized as follows: Given a set of n simple areas $A = (A_1; ....; A_n)$, find a minimal-cost cyclic tour $T = (p_1; ...; p_{n+1})$, such that it visits A_i in the point p_i and $p_1 = p_{n+1}$.

If Areas are simple and with overlap, TSPN-S-WO arises:

The Traveling Salesman Problem (TSPN) is formalized as follows: Given a set of n simple areas $A = (A_1; ....; A_n)$ with partial or total overlap,, find a minimal-cost cyclic tour $T = (p_1; ...; p_{n+1})$, such that it visits A_i in the point p_i and $p_1 = p_{n+1}$.

It will be shown that passes within Overlap Zone cannot be the best solution in term of path length, this leads to the necessity to develop special operator able to recognize when is convenient to pass through overlap zone and when not, this problem has been named henceforward Overlap Position Problem (OPP), and will be solved by means of a special operator named OPP-operator, applied to TSPN-C-WO.

Another version of Salesman Problem has been investigated by Mennell [41], named CETSP.

If regions to be visited are disks one, and in partial or total overlap, problem can be formalized as Close Enough Travelling Salesman Problem (CETSP). The Close-Enough Traveling Salesman Problem (CETSP) is another variant of the Traveling Salesman Problem (TSP).

In the CETSP, the tour must pass within a specified distance of each node. That is, the salesman must get “close-enough" to each node to visit it [41].

When all customer radii are zero, the CETSP reduces to the TSP [42], otherwise if areas are not disks but have a generical shape and in partial or total overlap, we are talking about TSPN-WO.

Below, it will be explored literature only for TSPN and CETSP which are two cases which better formalized problem of interests, it's important to underline that literature is full of method to solve Travelling Salesman Problem, so this part of dissertation aims not to give a full and complete literature review, but a survey about main method found in it.

Overlap problem is widely treated in CETSP and less in TSPN-WO, this is the reason why CETSP literature has been deeply studied as TSPN and TSPN-WO ones.

TSPN has been solved by Gentilini [43] who implemented a heuristic to speed up a Mixed-Integer Non-Linear Programming solver. This method has good performance, showing good close to optimum result, particularly its performance has been tested only for up to 16 regions.

An idea is to use algorithms develop to solve TSP domain to solve TSPN, since during the years TSP received much larger attention from researchers than TSPN, this give as results that a great number of algorithms exists to solve it.

One approach is to convert a TSPN problem to a Generalized TSP (GTSP) by replacing the areas with a set of points.

Oberlin et al. [44] showed that it is possible to convert GTSP into a TSP. Though this process is possible in theory, it immensely increases the search space and becomes practically infeasible [45].

Another method to put on algorithms from TSP to TSPN is to represent every customer's area with one point. It translates TSPN into two subproblems: TSP and Touring-a-sequence of-Polygons Problem (TPP). TPP is a NP-hard problem where the goal is to find the shortest path that passes through a given sequence of areas [46].

Yuanlong Qin and Bo Yuan [47] proposed a method for TSPN with Overlapping Neighborhoods named ACO-iRBA, iRBA in which the TSP and TPP tasks are tackled simultaneously by ACO (Ant Colony Optimization) and iRBA, an improved version of RBA (Rubber Band Algorithm), respectively. ACO-iRBA can handle situations where the neighborhoods are overlapped. This method shows good, close to optimum results.

Recently, Yang [48] in this regard, solves Travelling Salesman Problem with arbitrary neighborhoods using a particle swarm optimization and a genetic algorithm to solve Travelling Salesman Problem with arbitrary neighborhood can be reduced to CETSP when the customers to be visited are discs. The HA can find optimal solutions for instances with not very large size. The HA guarantees to find high-quality solutions for all instances, which is useful in practical application.

However, TSPN with arbitrary neighbourhoods overlapped has not got much attentions, it's important to underline it can be started to study, considering solution studied and developed for CETSP which can be considered a special case of TSPN-WO, which gives solution in case of disks shape.

Gulckensy [49] examines CETSP, by surveying six heuristic that were developed to solve the problem. All the six heuristic is composed by three steps. First, a set of super-nodes was selected so that each customer disk contained at least one super-node. Second, a TSP tour through these super-nodes and the depot was generated. Third, the tour was improved while maintaining feasibility.

Yuan [50] solves CETSP developing an evolutionary algorithm, giving results on five problems which are better than the result produced by an approximation algorithm; however, the running times were large.

Mennell [41] and [51] following the methodology developed by Gulckensy, developed different type of heuristic to solve CETSP which most based on Steiner Zone concept.

Wang [42] solves CETSP develops an algorithm based on three phases called Steiner Zone Variable neighboorhod Search Heuristic, which shows optimal results for small and large instances.

However, CETSP has limited practical application due to in that case, taking all neighborhoods as discs or squares is far divorced from the reality in many applications.

As it clearly appears from this part of our dissertation TSPN-C-WO is a problem not studied in depth, moreover we'll to consider OPP problem in our calculation, which permits us to consider when it's convenient to pass through overlap zone.

Before defining challenges and contributions of this work, in Table 1 it has been reported a scheme which summarizes Literature Review, emphasizing in red our target:

TSP	Description			Key Aspects	
Problem	Input	Optimize	Output	Shape	Overlap
TSP	Set of Points	Sequence	Sequence of Points	Points	--
TSPN	Set of Areas	Sequence & Location	Sequence of Points	Circle, Ellipses and Disks	No
TSPN-S-WO	Set of Overlapped Areas	Sequence & Location	Sequence of Points	Circle, Ellipses and Disks	Yes
TSPN-C	Set of Overlapped Areas	Sequence & Location	Sequence of Points	Polygon	No
TSPN-C-WO	Set of Overlapped Areas	Sequence & Location	Sequence of Points	Polygon	Yes

Table 1 – TSPN Literature Review Scheme

3.4 Robotic Path Planning

Robot path planning is an interested topic in the robotic community. The path planning problem is applied to all those applications with involve automated systems: painter robots [52]; cleaner robots [53]; spot welding robots [54]; remote laser welding robots [55]; underwater inspection vehicles [56]; measurement robots [57].

Regarding path planning problem, we know that path is generated according to task sequence and attributes.

The route which leads from one robot configuration to another is named path. A path is the locus of TCP points when it moves throughout configurations, for given sequence of tasks [58]. Path planning allows to compute robot configuration for each TCP pose and the sequence of configurations that moves the robot among configurations.

The computation is performed into the configuration space of the robot, also called C-space, where all robot information is present.

Spitz et al. [57] proposed heuristics method based on TSP tour construction to solve CMM path planning minimising path length. They consider obstacles but neglect robot redundancy assuming just one configuration for each pose.

Gueta et al. [59] proposed a method to avoid collisions between robot and workpiece placed on rotating table. They optimize cycle time considering system redundancy to select a different configuration for collision-free path assuming straight-line path fixed in C-space. They model the problem as a cluster of configurations solved by a heuristics TSP algorithm.

However, none of them use a complete integration of task sequencing and path planning. Classical approaches consider a simplified formulation: task sequencing and path planning problem are completely or partially decoupled [60]. As no robot information is involved in task sequencing, no attributes can be directly computed in T-space (neither A1 as it involves the information about the robot configurations). As consequence, no feasible solutions are guaranteed in T-space. Therefore, these single-attribute methods require multiple iterations between task and path to converge to a (near)optimal solution which can be far from the optimal one.

3.5 Integration Approaches

Wurll et al. [61] first introduced the integration concept. The authors introduce the multi-goal path planning problem (MTP) where a cluster of poses is modelled as a GTSP to find a collision-free path solved in T-space. Later, Faigl et al. [62] face MTP with regions. In general, given a set of robot goals, MTP stands to find a shortest path among goals.

The problem of task integration is presented in [63], even if the authors predefine the sequence by means sampling without consider it in the optimization.

Zacharia et al. [64] introduce a method to simultaneously solving motion planning and task sequencing with a TSP formulation. Indeed, they deal with a fixed task-points therefore with a fixed EE position.

Recently, in robotic remote laser welding, Kovacs [65] introduces a novel model problem: Traveling Salesman Problem with Neighbourhoods and Durative visits (TSP-ND) for task integration. He proposes a meta-heuristic approach based on Greedy Randomised Adaptive Search Procedure (GRASP) to solve TSP-ND.

3.6 Summary

Finding optimal sequence of tasks is crucial in all industrial applications where repetitive jobs are performed.

Currently, there are few approaches that allow an automatic task sequencing. None of them use a complete integration of task sequencing and path planning lead to solutions far from the optimal one.

Multi-attribute path planning with a relatively simple computation process seems be a very big challenge. Adding multiple attributes increase the search space making exact methods application difficult [66]. Researchers use mainly decomposition approaches to reduce the problem to simple ones that can be solved sequentially or parallel applying heuristics methods to get solution in reasonable time.

Chapter 4.

4.PROPOSED METHDOLGY OVERVIEW

This chapter describes the proposed methodology, highlights formulation and gives an overview of the developed methodology.

The proposed methodology aims to solve the task sequencing problem using the Travelling Salesman Problem with Neighbourhoods (TSPN) in its most complex form which is TSPN-C-WO, using Euclidean distance as key metric.

Starting from the consideration that to cross within Overlap Zone cannot be the best solution in term of path length, a distinctive feature of our solution is that it's fitted with a special operator able to recognize when is convenient to pass through overlap zone and when not, named OZPS.

We develop an Overlap Zone Position Search (OZPS) to solve TSPN-C-WO, which solve OPP by means of OPP-operator.

Comparative benchmarking results are then showed to prove the effectiveness of the proposed method.

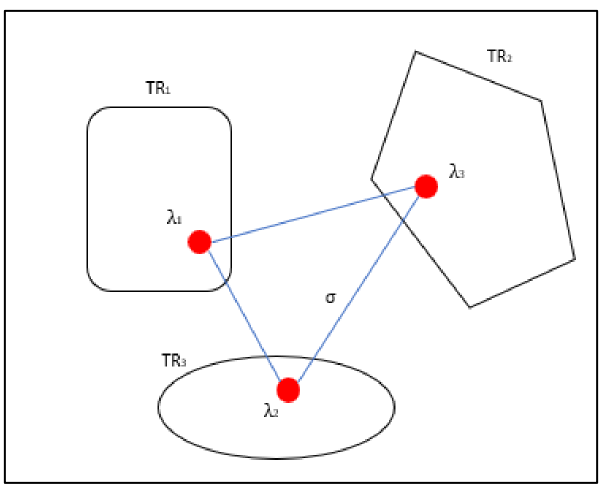

Figure 15 - Sequence of Task

4.1 Introduction

Industrial robots perform a cycle of tasks to carry out a job. For a given task, T_i, robot can reach a pose $\boldsymbol{\lambda}_i$ to perform that task.

Given n number of tasks, for each task "i" there exist a region "TR" defined by technological parameters which characterize EE-pose for the task execution. These parameters are related to specific applications. For example, technological parameters for inspection robots with optical camera system are optimal operating distance, depth, length and width of field of view.

We aim to find the optimal sequence, $\boldsymbol{\sigma}$, of poses, for pre-defined multi-attribute, by minimizing cycle time t_{min} as well as selection of the optimal pose for each task/region. Therefore, one can formally write:

$$\forall\, T_i \;\exists\, \lambda_i \;\in TR_i : \sigma \rightarrow t_{min} \quad (1)$$

$$i \;\in [1, n]$$

4.2 Methodology Overview & Key Principles

The proposed method aims to find an optimal task sequencing taking into account robot attributes into the T-space (see Figure 16).

A pose can be defined in T-space by means of position $\boldsymbol{P}=(x,y,z)$ and orientation $\boldsymbol{Or}=(\alpha,\beta,\gamma)$; whereas, in C-space through configuration $\boldsymbol{q}=(\vartheta_1,...,\vartheta_m)$ where m is joint numbers. Although in C-space it is possible to define a complete EE-pose as well as robot configuration, it is difficult to define an optimal task sequence. Therefore, it is more convenient to model the robot task sequencing problem in the T-space; therefore, the proposed approach formalizes the task sequencing problem in T-space and brings attributes from C-space, with the aim of calculating both optimal task sequence and feasible poses. The reader may notice that feasible poses can only be computed in the C-space, where robot information is made available.

Given a set of n tasks, the proposed method firstly defines task regions TRs and, then, calculates minimum distances among them. Subsequently, it selects via-pose by simultaneously optimizing attributes as:

- Pose-to-pose distances
- Pose accuracy
- Collision
- Reachability
- Robot Overall Collision

Finally, a task sequencing $\boldsymbol{\sigma}$ is generated through via-poses λ_i, $\forall i=1..,n$.

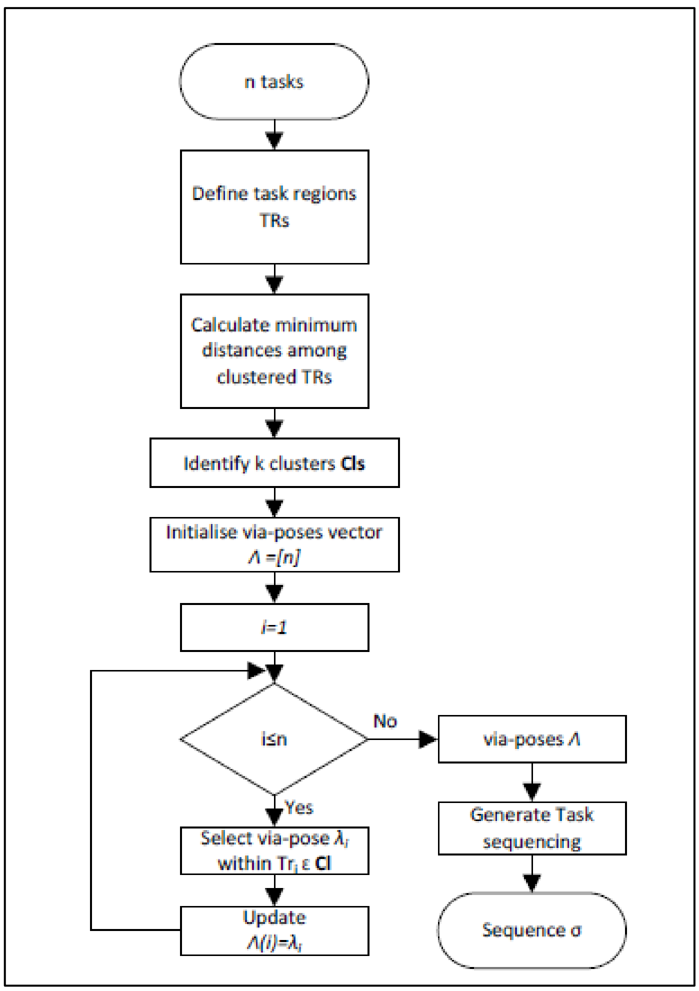

Figure 16 - Task Sequencing Flowchart

In the T-space, robotic task sequencing can be modelled as TSPN, where each neighborhood (region TR) represents robot task and any inner points represent the position, $\boldsymbol{P}$, of the EE-pose, particularly a TSPN-C-WO solving method has been developed.

Task region is the locus of feasible points, which can be visited by the end-effector to perform that task. In many applications, robot task can be formalized as a 2D task region, so TSPN for Robot Task Sequencing can be modelled as 2D area.

The proposed method, named OZPS, solves the TSPN-C-WO using Euclidean distance as key metric. Then, OZPS has been expanded and named Augmented A-OZPS to introduce pose orientation and to check he feasibility of Overall Collision.

4.3 OZPS

This chapter presents a new method to solve TSPN-C-WO. In particular TSPN-C-WO is a problem not studied in depth, moreover nobody considers OPP problem in their calculation considering that passing through overlap zone is always the best choice to solve TSPN using Euclidean distance as key metric.

There are 3 challenges accounting for the adoption of solution given by methodologies explained over it for TSPN with arbitrary neighborhoods overlapped:

1. How to take into account if to cross overlap zone
2. How to solve TSPN with not simply shapes
3. How to calculate Overlap Zone between different shapes

To overcome the challenges mentioned above, this paper studies the TSPN in a general case of arbitrary overlapped neighborhoods and makes the following contributions:

1. To develop an operator, able to recognize if to cross overlap zone
2. The proposed approach solves TSPN in case of complex forms, making the methodology especially practical in real-world applications
3. Develop a new method, to calculate Steiner Zone, making use of a mesh-grid created to find exact location of each point

Comparative benchmarking results are then showed to prove the effectiveness of the proposed method.

In the remainder of this chapter, we will first discuss problem background and then give a formal description of the problem and challenges faced. We develop OZPS. Then we present computation results on instances specially created. We give our conclusion and discuss future work in last part of this chapter.

4.3.1 Problem Background

4.3.1.1 Problem Description

Formulation is restricted to 2d space (i.e. R^2) and it is based on an Euclidean distance function d (p; q) to denote the cost of moving between two points p; q R^2. Of course, higher dimensional spaces as well as other distance functions are possible.

The Traveling Salesman Problem with Neighborhoods is formalized as follows:

> 'Given a set of n polygons A = fA_1,…,A_{ng}, find a minimal-cost cyclic tour T = (p_1,….,p_{n+1}), such that it visits Ai in the point pi and p_1 = p_{n+1}. TSPN requires a set of polygons as input data'

In its simpler version, TSPN does not consider Overlap Problem. The Traveling Salesman Problem with Neighborhoods with overlap is formalized as follows:

> 'Given a set of n polygons A = fA1,….,And in partial overlap, find a minimal-cost cyclic tour T = (p1,….,pn+1), such that it visits Ai in the point pi and p1 = pn+1,considering that if tour visits area in common to two areas, both of them can be considered visited'

A problem which is like TSPN-WO, and which is widely treated in literature is the Close-Enough Traveling Salesman Problem. The Close Enough Traveling Salesman Problem with Neighborhoods is formalized as follows:

> 'The Close-Enough Traveling Salesman Problem (CETSP) is a variant of the Travelling Salesman Problem (TSP). Unlike TSP, in CETSP, the tour must pass within a specified distance of each node. That is, the salesman must get 'close-enough' to each node to visit it'

4.3.1.2 Steiner Zone

If a tour crosses through an overlap of several areas, all customers that define those customers regions can be considered served [42].

Overlapping region in Mennell works has been defined as 'Steiner Zone'. A Steiner zone is an overlap of disks. If a Steiner zone is contained in at most *k* disks, it has degree k [42], particularly a region that is not contained in any other region will be defined a Steiner zone of degree 1. A point in a Steiner zone (including those with degree 1) is called a Steiner point. In figure 17, we show examples of Steiner zones with different degrees.

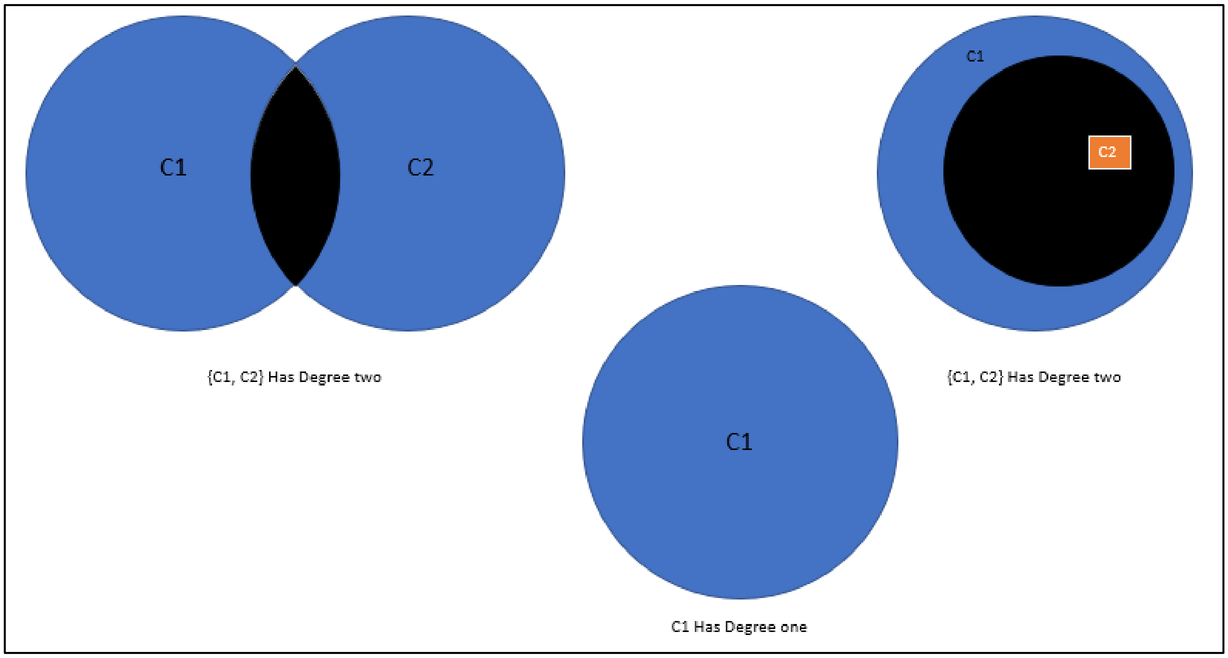

Figure 17 - Steiner Zone of Various Degrees

4.3.1.3 Problem and Challenges

In most of TSPN and CETSP literature methodology, a salesman visits 'Steiner Zone' instead of visiting each area individually, leading the path to be shorter.

In this work we question this consideration, considering that, in case of non-simple geometric shapes with a particular position in the plane, passes within overlap zone, cannot be the best solution in term of path length, this is what it has been named Overlap Position Problem OPP.

Let us to show a figure to introduce the problem, in figure 18 it has been reported two possible overlap configurations, point A and D, are path starting and finishing points, points B and C are Zona 1 and 2 nearer points to A and D, which are plausible point to cross to find best path, without passing through Steiner Zone (Black Zone).

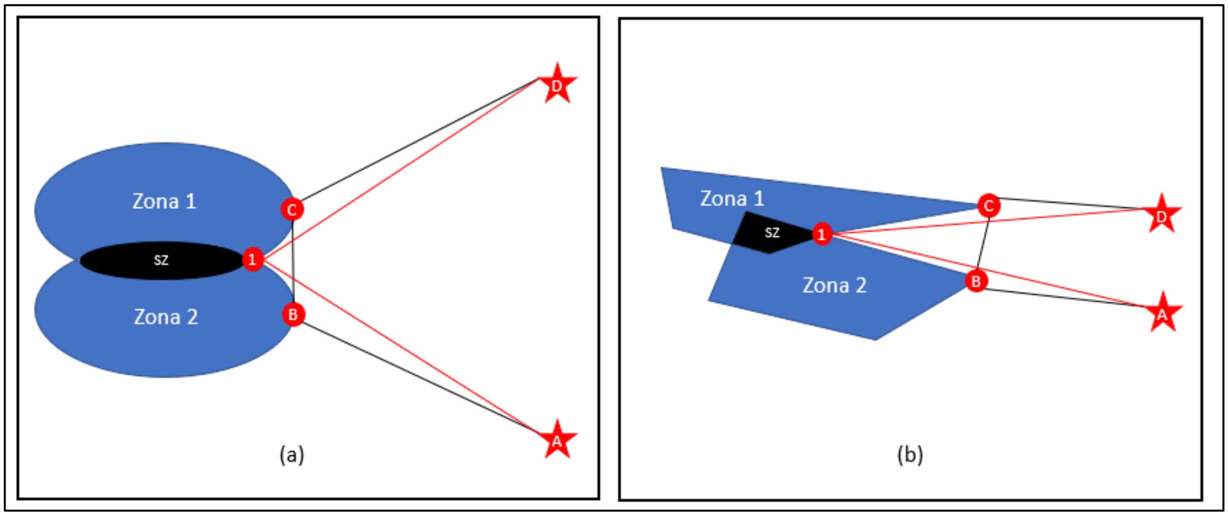

Figure 18 - Overlap Position Problem

In Fig.18 (a) it has been reported a customer configuration in which passing through Steiner Zone is convenient in term of path length, as a matter of fact ABCD route is longer than A1D one. Otherwise in Fig.18(b) passing through Steiner Zone is not convenient in term of path length, because of Steiner Zone position which is too far from route develop zone (defined by point A and D).

In this work, it has been proposed an operator named OPP-operator that is able to understand, solving classical TSPN-C-WO, when it's convenient to pass through overlap zone, and when not.

Moreover, we propose a new method to evaluate overlap zone which make use of a specific mesh-grid.

4.3.2 Proposed Approach

We develop an OZPS to solve the TSPN-C-WO problem. OZPS has 3 phases. First of all, Steiner Zone has been calculated with a method which makes use of Mesh-Grid Processing. Second, OZPS built a hull which passes through centroid of areas which forms external convex hull, named henceforward Home Zone, which geometry it has been represented in figure 19, for a specific configuration shape.

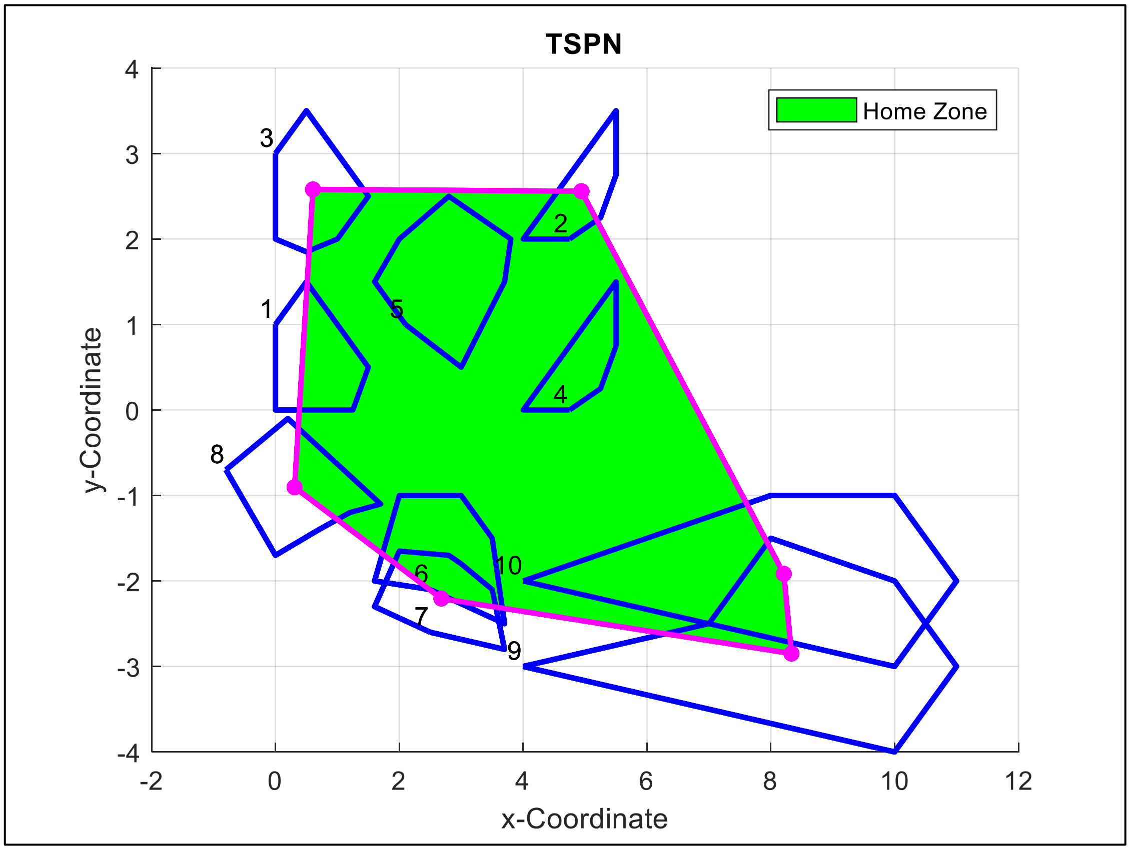

Figure 19 - Home Zone

Then for each Steiner Zone following algorithm has been developed:

- if a generic Steiner Zone Centroid is inside Home Zone, all areas that generate this Steiner Zone have been eliminated and Steiner Zone retained
- if a generic Steiner Zone Centroid is outside Home Zone, all areas that generate this Steiner Zone have been retained and Steiner Zone not considered

In Fig 20. b) four examples have been reported in which best path is reported in blue line, while home zone as red line, showing that best path is inside Home Zone.

Moreover, Fig 20 a) shows that path found by algorithm is equal to the best one.

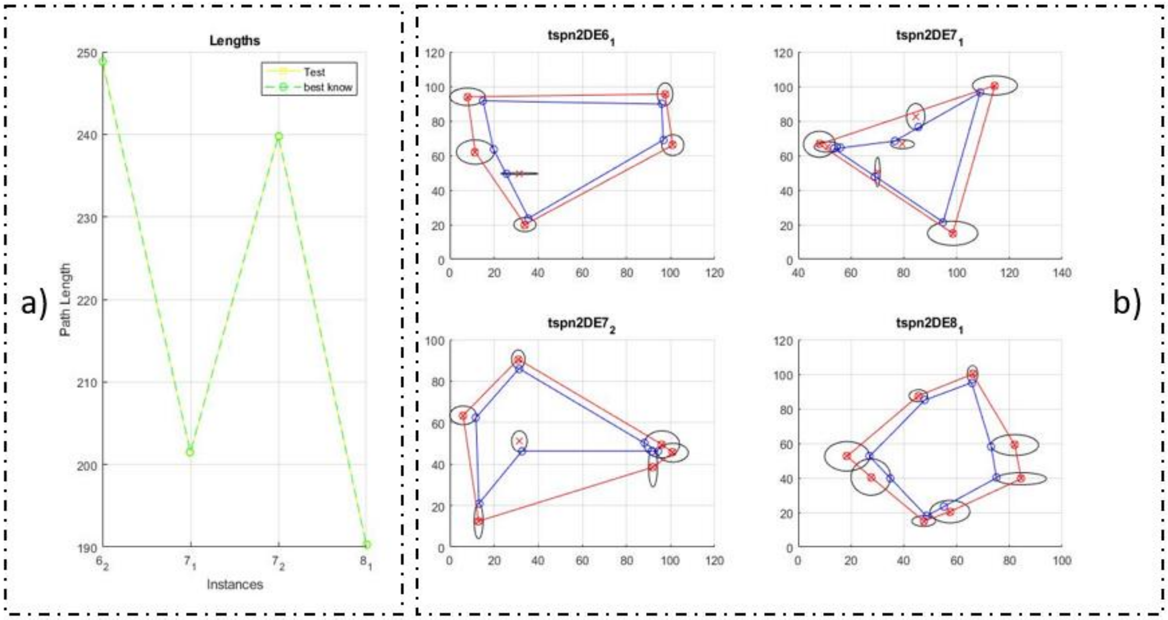

Figure 20 - Home Zone Assumption Validation

It could happen that in some cases assumption is not valid, otherwise Home Zone concept has been developed in order to solve case with complex areas which are overlapped.

In figure 21, it has been reported a classical situation in which OPP-operator works, particularly Steiner Zone whose centroid has been marked with a red and filled circle marker will be retained because of centroid is inside Home Zone Convex Hull, while Steiner Zone whose centroid has been marked with a light blue and filled circle marker will be eliminated and replaced by areas which are in overlap because centroid is outside Home Zone Convex Hull.

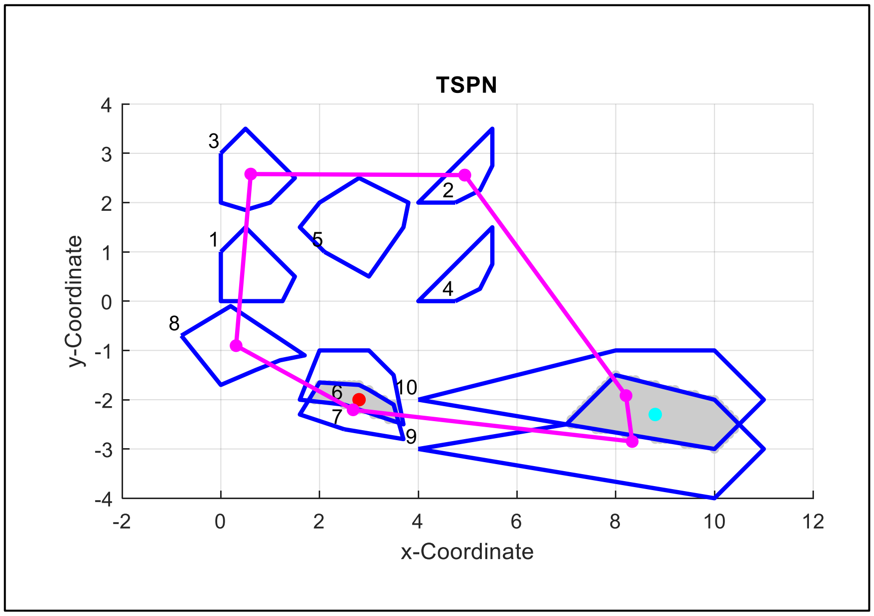

Figure 21 - Representation of Home Zone despite Steiner Zone Position

Finally, taking from previous phases a configuration of shapes, classical TSPN problem has been solved.

4.3.2.1 Steiner Zone Construction

To better explain how this method works, it will be described by following the explanation the resolution of a test instance with 9 shapes.

First, we need to initialize Geometric Shapes and a Mesh-Grid, whose role will be explained below, which have been reported in figure 22:

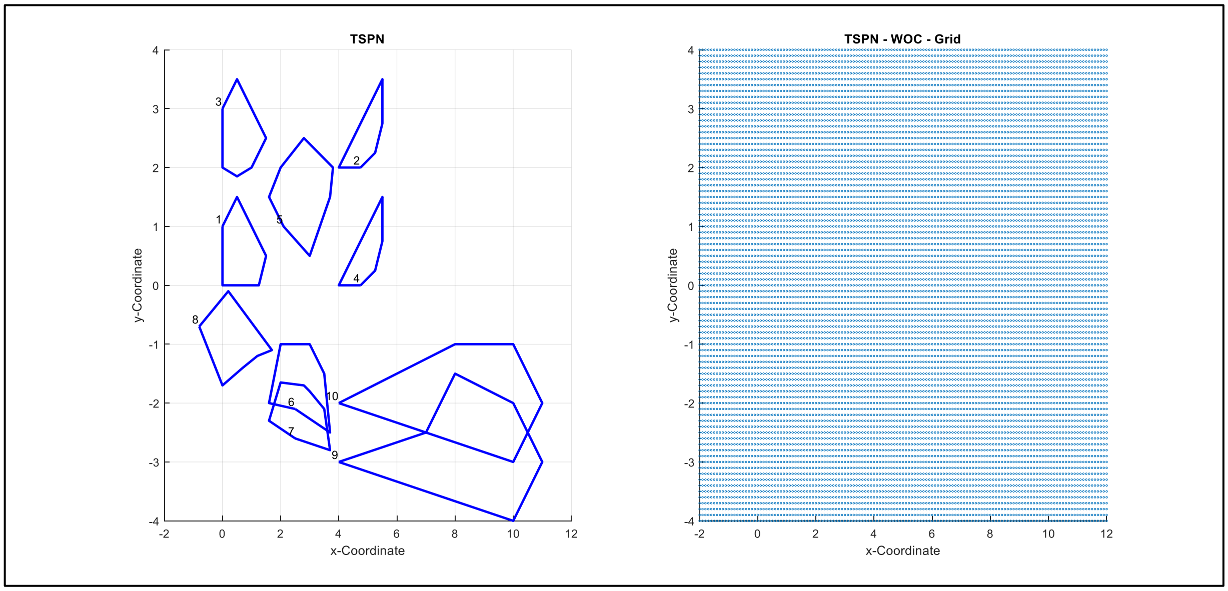

Figure 22 - left) Geometric Shapes, right) Mesh-Grid

Then Mesh-Grid and Geometric Shapes will be intersected:

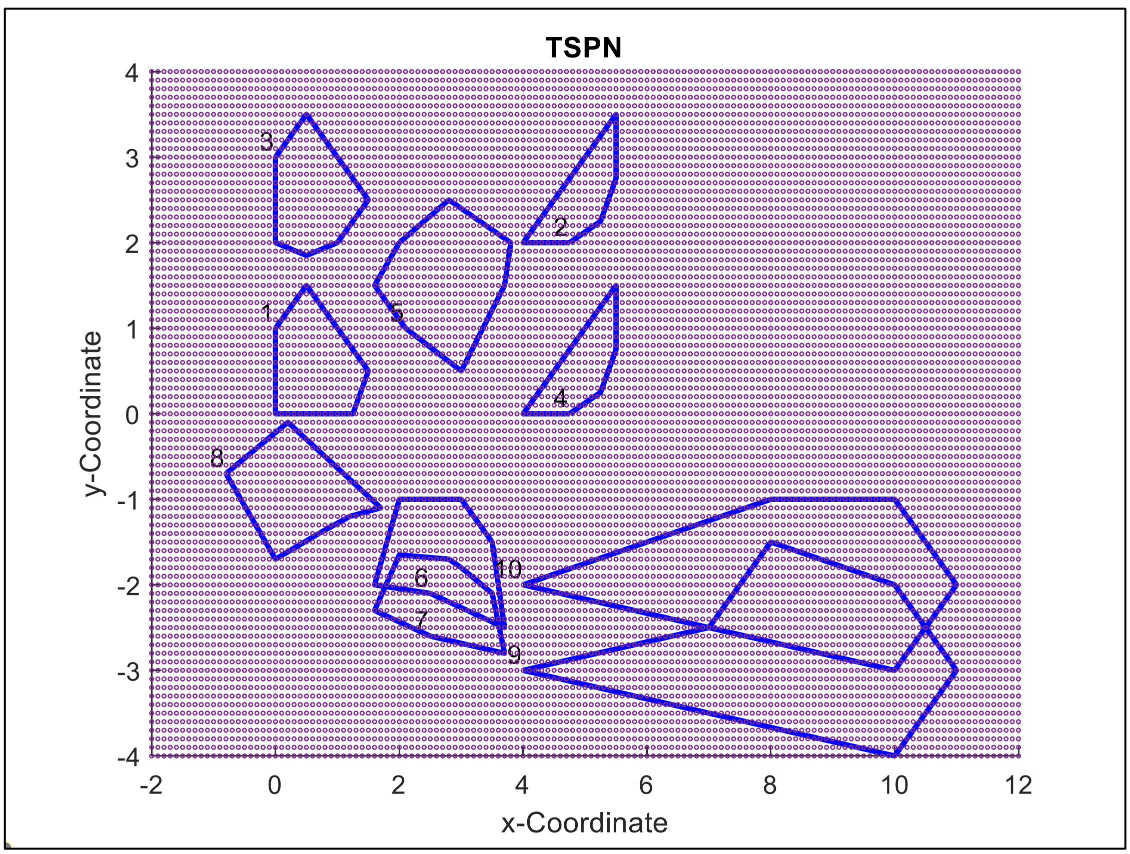

Figure 23 - Mash-Grid and Shapes Intersection

To discriminate points which are inside a shape it has been used the following logic, which can be divided into sequent steps:

1. Define a Generical Shape and a Generical Point you want to know if it's inside or outside the Generical Shape
2. Define a Line which starting from Generical Point intersects Generical Shape
    a. If Line intersects Generical Shape an even number of times it means that is outside Generical Shape
    b. If Line intersects Generical Shape an odd number of times it means that is inside Generical Shape

With this approach we can identify points inside and outside every shape in the plane.

One time that all points inside different shapes have been discovered, we define the part of plane which represents points inside the shapes with value 1 and points outside with value zeros. Figure 24 simply explains this approach.

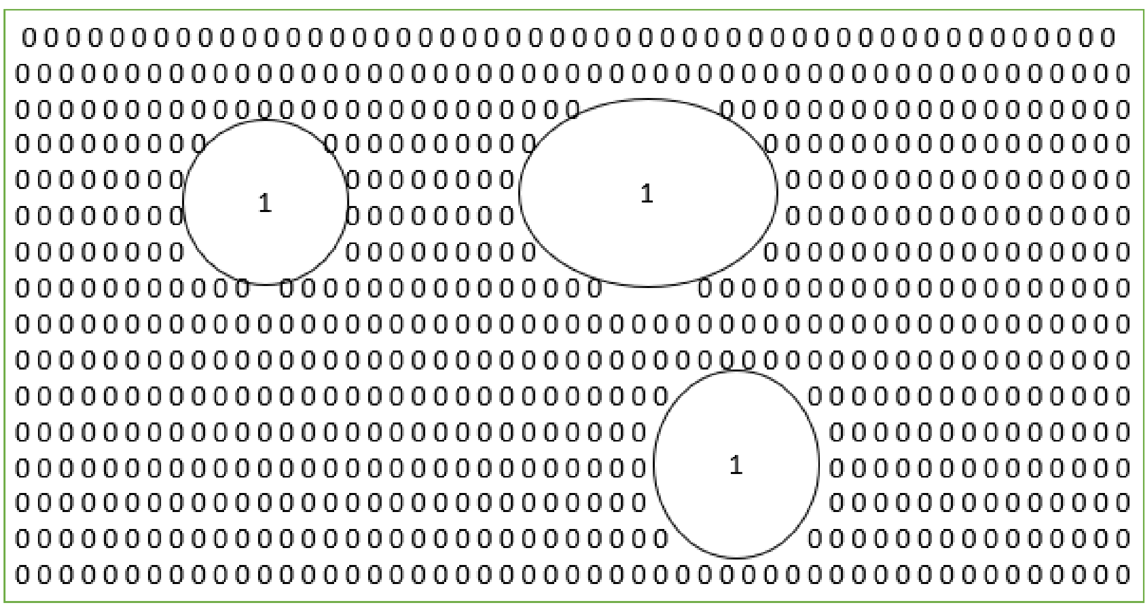

Figure 24 - Overlap Calculation Scheme

In the zone where shapes are in overlap, also the one-value are in overlap and will be sums up becoming 2 or 3 …4 depending from how many shapes are in overlap, giving us possibility to estimate Overlap Zone (2 represents two shapes which are in overlap, etc)

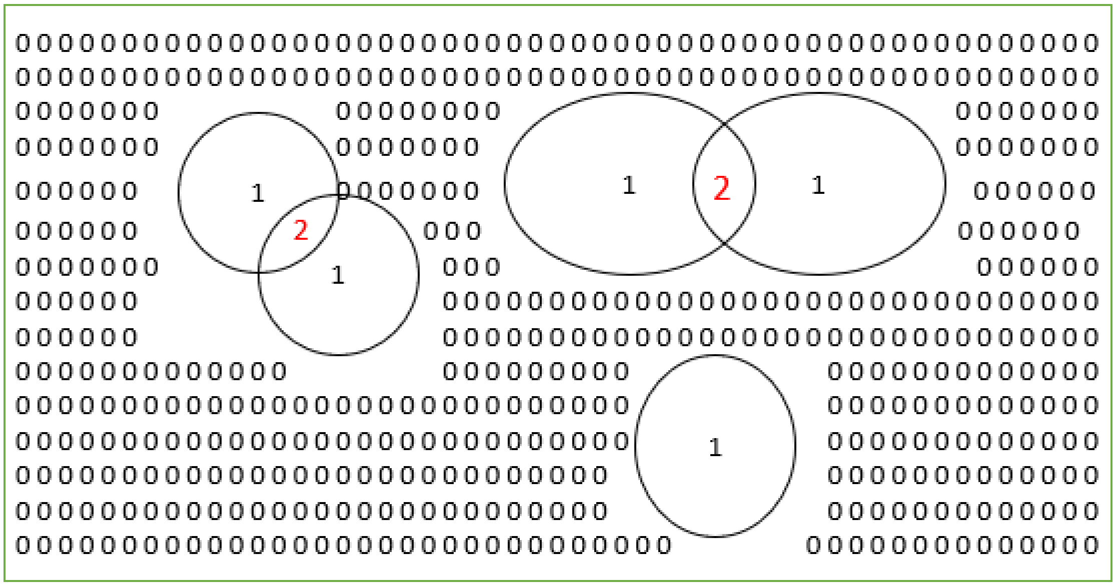

Figure 25 - Overlap Calculation Scheme

In case of simple geometric shape, Circumference and Ellipse, another approach can be used to define if a point is inside or outside a shape, which is explained in Appendix 1.

Computational problems can arise when shapes, due to high dimensions and position, are spread out strongly in the plane. To better understand this problem, it can be observed two examples with different areas layout in the plane. In figure 26 (left) it can be observed that a low dispersion and dimension affects grid so that with a pre-defined frequency value a good sampling of the plane can be achieved without excessive computational load, otherwise in figure 26 (right) it can be observed that due to high dispersion, with some pre-defined frequency computational problems can appear due to excessive number of grid point to manage, giving as result that the algorithm becomes unresponsive, and no mesh-grid will be generated.

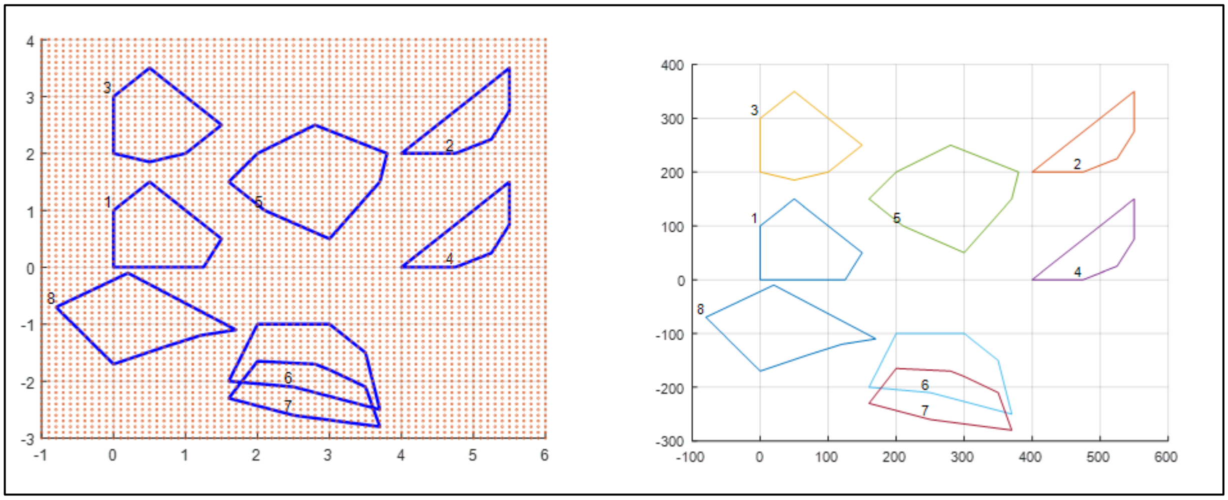

Figure 26 - left) Mesh-Grid right) Mesh-Grid Computational Problem

So, in order to maintain a low number of grid points to manage ensuring the quality of the results, before starting with OZPS algorithm, customers have been reported in a more restricted range where a small number of grid points have been used to create grid. After calculating the best path in such restricted range, result will be reported in its own range.

To verify that transposing a configuration of points from their own range to another one distance does not change, a test has been conducted. Particularly, 200 point have been randomly generated in the plane and distance between point one to all other points have been calculated (which has been named henceforward Original Distance), see figure 27.

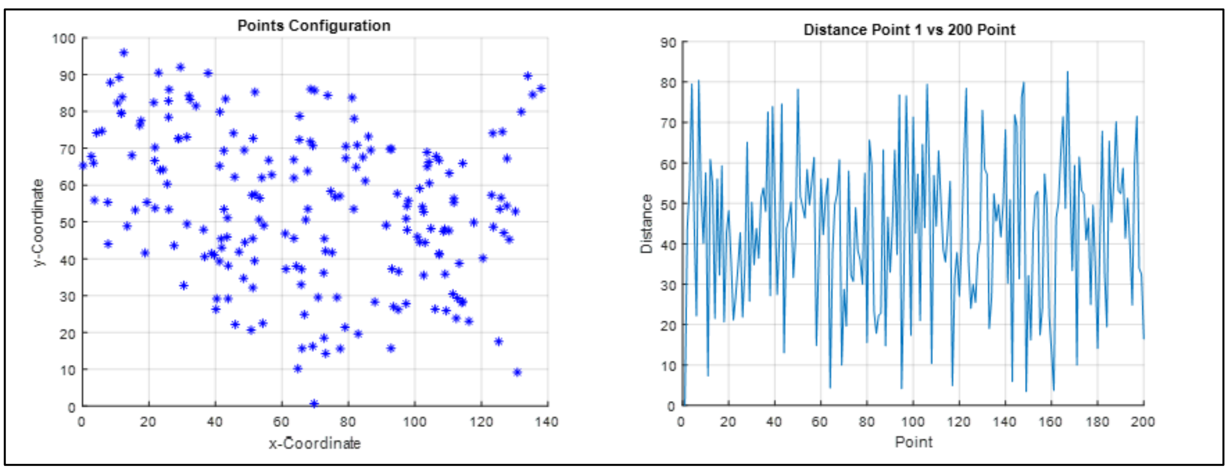

Figure 27 - left) Points Configuration, right) Distance between point one and rest of other points

Points under investigation have been reported in a more restricted range dividing each of them by 100, and distance between the new first point and all other ones have been calculated, which has been named Compared Distance.

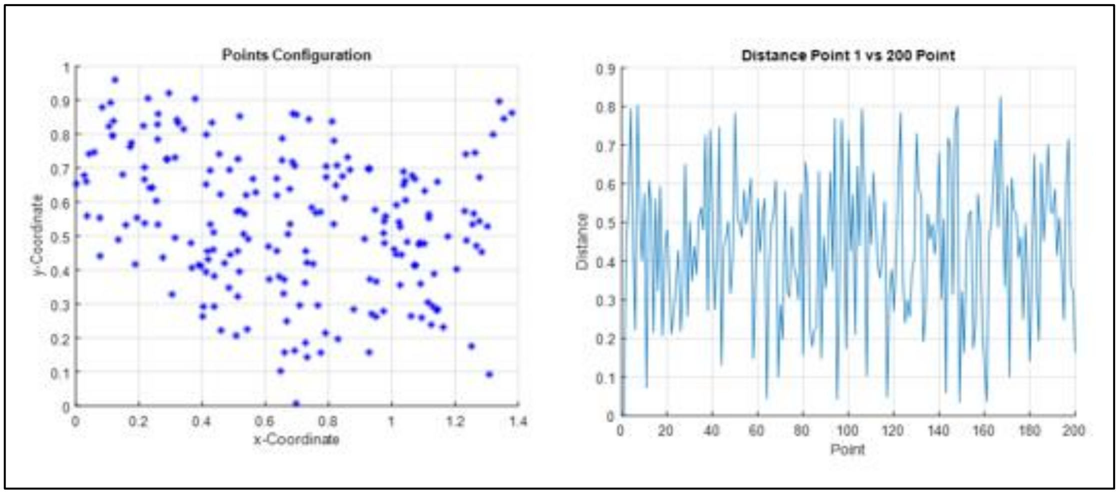

Figure 28 - left) Points Configuration, right) Distance between point one and rest of other points

Finally, Compared Distance has been multiplied by 100 and compared with original one (Original Distance). Result shows that distance value is not modified passing it through range (equal results has been reached also for other points).

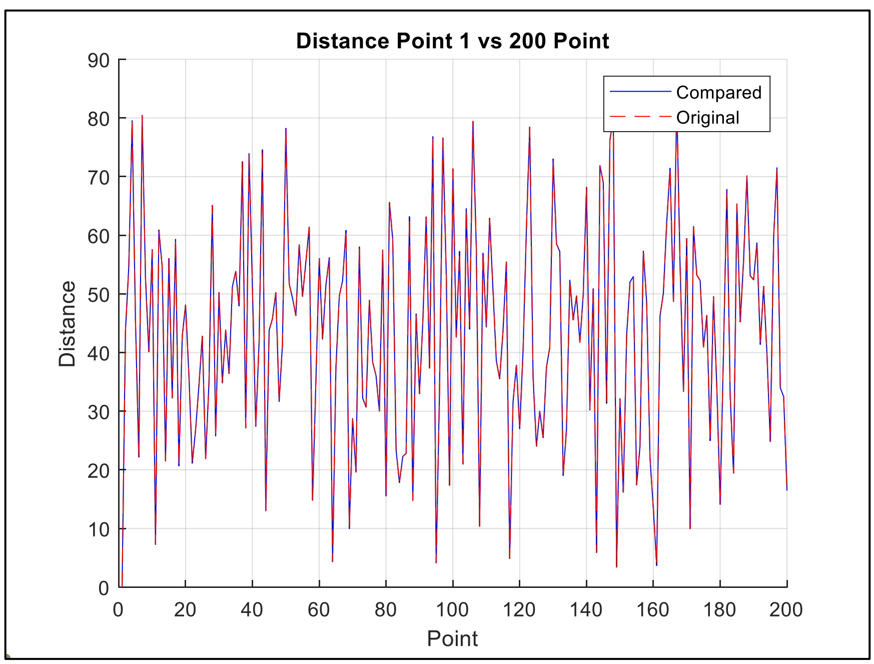

Figure 29 - Original Distance vs Compared Distance

4.3.2.2 Convex Hull & Dataset Reshaping

With reference to Home Zone and OPP-operator, our algorithm suggests that:

- Red Steiner Zone, which is inside Home Zone, has to be retained and all areas that generate this Steiner zone eliminated
- Light Blue Steiner Zone which is outside Home Zone, has to be eliminated and all areas that generate this Steiner zone retained

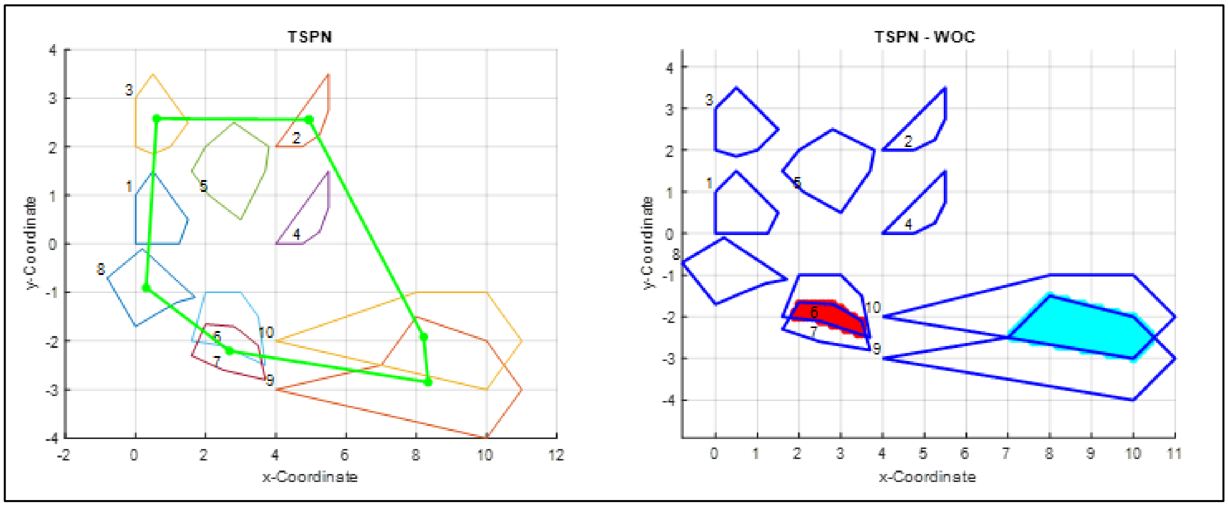

Figure 30 - left) Home Zone, right) Steiner Zone Position

We check that not all customers need to be crossed to construct a feasible tour and to obtain best solution. We start Data Reshaping phase which consist in reshaping dataset that must be considered to find best path, in our example dataset will be composed by – [1 2 3 4 5 red 8 9 10].

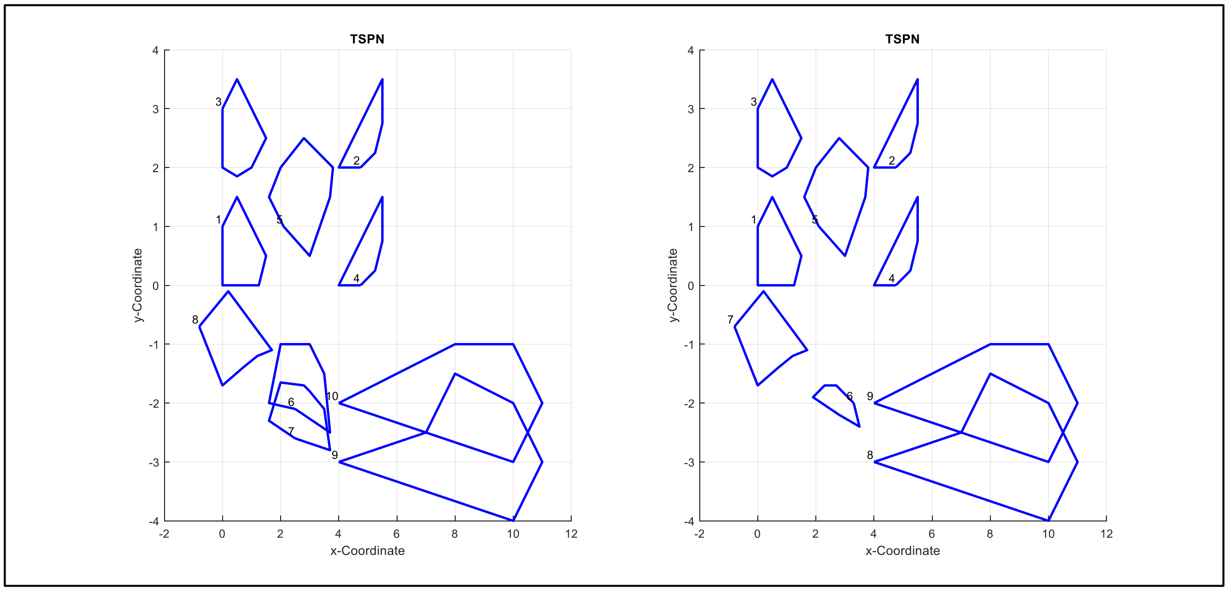

Figure 31 - left) Initial Dataset right) Reshaped Dataset

Next phase deals with solving classical TSPN problem with New Reshaped Dataset.

4.3.2.3 TSPN

In this phase we solve classical TSPN problem, having as input a a set of n polygons $A = (A_1; \ldots; A_n)$, and whose target is to find a minimal-cost cyclic tour $T = (p_1; \ldots; p_{n+1})$, such that it visits A_i in the point p_i and $p_1 = p_{n+1}$.To solve that phase, no innovative approach has been developed but it has been reproduced a classical algorithm studied in literature.

Below it has been reported results furnished by the algorithm for our own test instance. To evaluate algorithm performance, it has been evaluated best path calculated by the algorithm without considering OPP operator (figure 32) and considering OPP operator enabled (figure 33). As it can be observed from figure (32), path in order to pass through overlap zone, stretches along right part, giving as result a path which is longer than path evaluated using OPP operator.

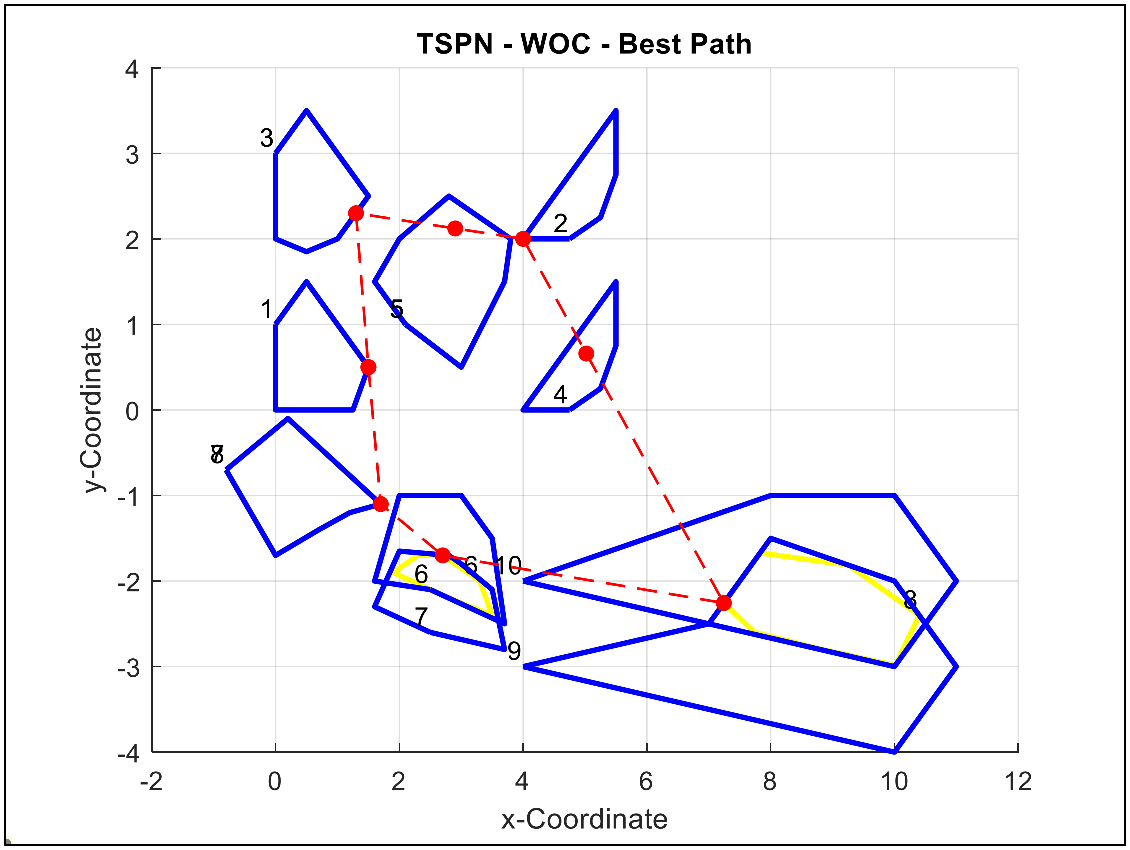

Figure 32 - Best path in case of OPP operator disabled Length→17.23

In figure 35 it has been reported results in case of OPP operator enabled. Result shows that algorithm does not consider overlap zone composed by 9 and 10 shapes because of Steiner Zone centroid of 9 and10 areas is outside Home Zone (please refers to figure 21).

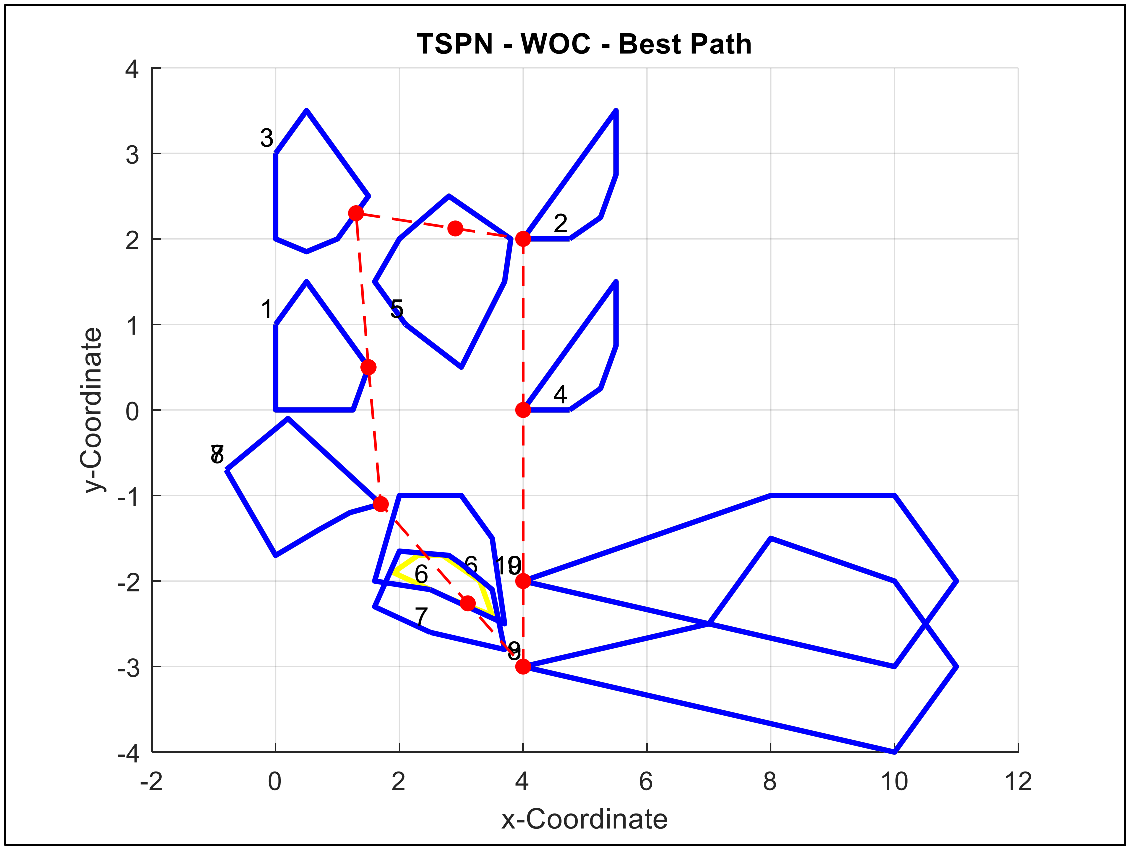

Figure 33 - Best path in case of OPP operator enabled Length→14.12

To evaluate a numerical difference, path length in case of OPP operator enabled is **14.12**, while in case of OPP operator disabled is **17.23**, showing that OPP operator is able to discern when to cross Steiner Zone instead of areas which are in overlap.

4.4 Algorithm or Problem Formulation

Algorithm 1:

Input: Set of Polygons P = $\{P_1, P_2, \ldots., P_n\}$,Desired Increment f

Output: Set_Of_Polygons P'',Tour T = $(p_1, p_2, \ldots., p_n)$

```
1.  P,P'←Overlap Zone(P,f);                          % See Algorithm 2
2.  HZ←ConvexHull(P);
3.      for i=1 to Count(P);
4.          if True←IsInside(HZ,P') ;
5.              P''←Insert(P(i)) ;
6.          else
7.              P''←Reshape(P(i));
8.          end
9.      end
10. P''←Output
11. Best_Path←T;
```

- Overlap Zone → See Algorithm 2
- ConvexHull is a function that takes centers of Poligons and returns 2-D convex hull of the centers in matrix P.
- IsInside is a function that takes Home Zone and Overlap zone and returns Boolean 1 is overlap zone centroid is inside home zone while Boolean 0 if overlap zone centroid is outside home zone.
- Insert is a function that takes Home Zone and Overlap Zones and replaces the overlap zones with the zones that compose it
- Reshape is a function that takes Home Zone and Overlap Zones and replaces the zones that make up the overlap zone with the overlap zone itself.

Algorithm from line 3 to line 9 reflects Home Zone strategy, particularly an Overlap Zone has to be taken into account in the Best Path calculation just in case that its center is Inside Home Zone, otherwise it has to be taken into account entire areas which composes this Overlap Zone.

Algorithm 2: Algorithm → Overlap Zone

Input: Set of Polygons P = {P_1, P_2,…., P_n},Desired Increment f

Output: Set of Polygons P = {P_1, P_2,…., P_n} , <u>*Set of Overlap Zones*</u> P’ = {P'_1, P'_2,…., P_m}

```
P_ext ← MostExternalPolygon(P)
p_ext ← MostEternalPoints(P_ext)
[p_ext_x_max, p_ext_x_min , p_ext_y_max , p_ext_y_min] ← MostExternalPoints(P_ext)
x = [p_ext_x_min : f : p_ext_x_max] ;
y = [p_ext_y_min : f : p_ext_y_max] ;
[Grid_x , Grid_y] ← Grid (x,y) ;
  for j=1:Number_Of_GridPoint
    for i =1:Number_Of_Polygon
        InMatrix(i,j)  = inPolygon(Gridx(j),Gridy(j),Px(i), Py(i));
    end
  end
Overlap_Vector ←Sum(InMatrix);
Max_Ovelap_Score←max(Overlap_Vector)
   for j =2:Max_Overlap_Score;
        Cumulate{i,1} ←Cumulate(Overlap_Vector == j);
   end
OverLap_Coordiante{i,1}←Translate_From_Index_To_Coordinate(Cumalte{i,1});
Cluster{i,1}(1,Number_Of_Areas_With_Overlap_i)←cluster(OverLap_Coordiante{i,1});
```

Several functions are involved during calculation.Here :

- MostExternalPolygons is a functions that takes Polygons and returns 4 most external Polygon along x and y, defined as Pext
- MostExternalPoints is a function that takes Pext and returns most external Points for each most external polygon defined by MostExternalPolygong functio
- InPolygon is a function so defined → in = Inpolygon (xq,yq,xv,yv) returns in indicating if the query points specified by xq and yq are inside or on the edge of the polygon area defined by xv and yv.
- Sum is a function that takes a Matix and returns a row vector containing the sum of each matrix Column
- Cumulate is a function that takes a row vector and groups data in different cell in such a way that data inside some cell have some value
- Translate_From_Index_To_Coordinate is a function that takes Cumulate Cell and returns Coordinate x,y relative to index inside each cell

- Cluster is a function that takes 2-D coordinate inside each cell and returns clusters based on their 2D coordinate

The algorithm takes a set of Polygonal P as an input, and a constant value f which needs to build grid.

Line 1 of Algorithm 2, thanks to MostExternalPolygons function, evaluates 4 most external polygon along x and y.

In the line 2 and 3 MostExternalPoint function takes as input most external polygons producted by previous function and calculate for each of them which are most external points along x and y axis,giving as ouput :

- $p_{ext\_x\_min}$ which is x-coordinate of most external point on the left
- $p_{ext\_x\_max}$ which is x-coordinate of most external point on the right
- $p_{ext\_y\_min}$ which is y-coordinate of most over point
- $p_{ext\_y\_max}$ which is y-coordinate of most under point

In line 4 and 5 it has been calculated x and y vector, whose max a min value are defined by [$p_{ext\_x\_min}$, $p_{ext\_x\_max}$] and [$p_{ext\_y\_min}$, $p_{ext\_y\_max}$] and whose sampling frequency has been pre-defined and it's a fixed value which is f.

In line 6 it has been produced a grid,starting from x and y vector, particularly a 2-D grid has been built.

From line 7 to line 11 a for cycle evaulates which points are inside polygons and which outside.Paricularly a Matrix has furnisched as output whose index row refers to a particular Polygon and whose index column represent an Index referring to a particular Grid Point.So if Matrix(i,j) == 1 means that Grid points whose index is j , is indisde Polygon whose index is i.

In line 12 a Suming value operation along each column has been achieved, which means that values along each column which composes InMatrix has been sumed. As reasult it will be obtained a row vector evaluating each grid point overlap score,which has been named Overlap_Vector.

In line 13 it has been evaluate max value comparing in Overlap_Vector, which give us information about what is maximums overlap score present on the field.

From line 14 to line 16 a for cycle evaulate which are grid point that have some overlap score, and insert each of them inside a unique cell.As result of this it will be obtained different cells each of them containing point with some overlap score.

Line 17 translete grid index value contained in each cell to grid coordinate value.

Line 18 take as input cluster of point which have same overlap score, and by means of a clustering operation divide them in different zone, which represent different overlap zone.

As result of this it will be obtained different overlap aras each of them rapresenting a single overlap zone.

4.5 Results and Discussion

This section presents an evaluation of OZPS on two sets of instances. It's important to underline that OZPS has been developed to solve TSPN in case of complex shapes with partial or total overlap.

OZPS has been tested on TSPN benchmarks available in literature. Gentilini et al. [43] provide a set of 64 TSPN test instances formed by ellipsoids and polyhedral in $\mathbb{R}^2$ and $\mathbb{R}^3$; particularly test has been conducted in $\mathbb{R}^2$ domain with 16 areas.

To test OZPS performance on Gentilini test instances, it's been necessary to adapt it to solve instances with ellipses approximating each shape with a certain number of vertexes, making each ellipse a polygon.

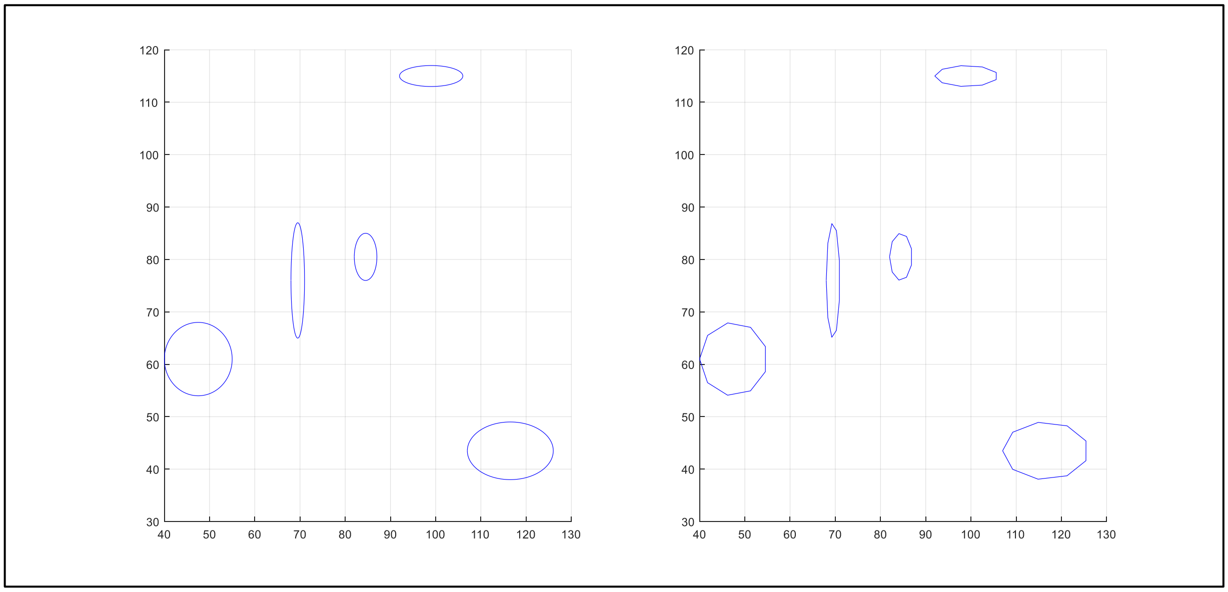

Figure 34 - Shape Approximation: left) Original shape; right) Approximated Shape

Two sets of instances were used to compare OZPS with the optimal values. The first set of instances (with up to 16 areas), the second one which is a proposed set of instances especially developed, named henceforward Polygonal Test Instances.

Evaluation of OZPS on First Test Instances (16 Areas)

The test \tspn2DE7 N" is decoded as a 2D test with 7 ellipses. N can be \1" or \2" that reflects the box size circumscribing the ellipse. Ellipses with the box size \1" are larger than with box size \2".

Using these instances, OZPS has been compared with optimal values.

It's important to underline that target of first test phase is exclusively to evaluate algorithm ability to find a path which is approximately close to optimum; it's obvious that results, due to shapes approximation, will not show an improvement despite literature results.

Despite optimal value, our algorithm shows good results close to optimal, reported in figure 35, if we consider that an approximation in results is due to approximating area with number of vertices which in this case are 20.

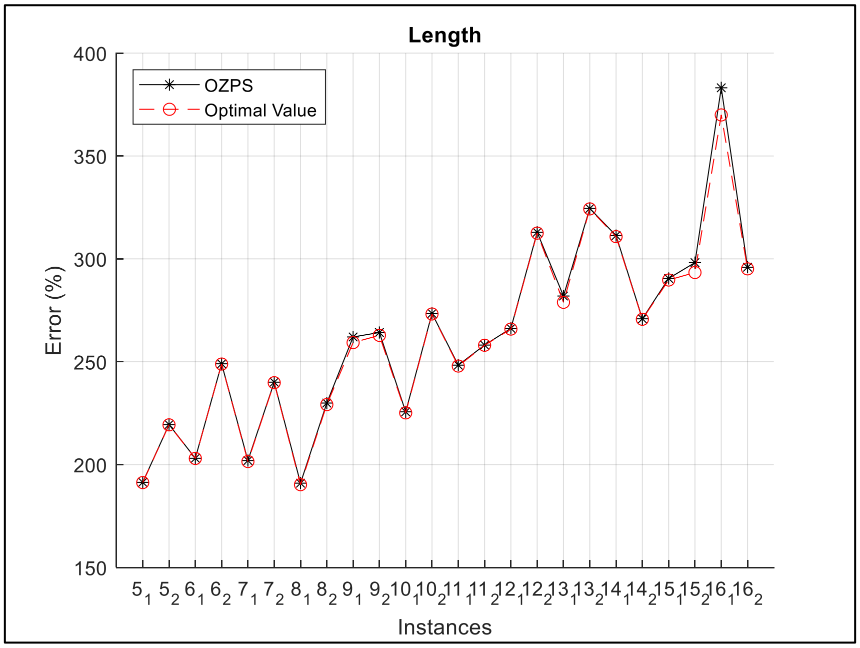

Figure 35 - Comparing Path Length among OZPS and BEST

Below it has been reported percent error committed by OZPS despite optimal value:

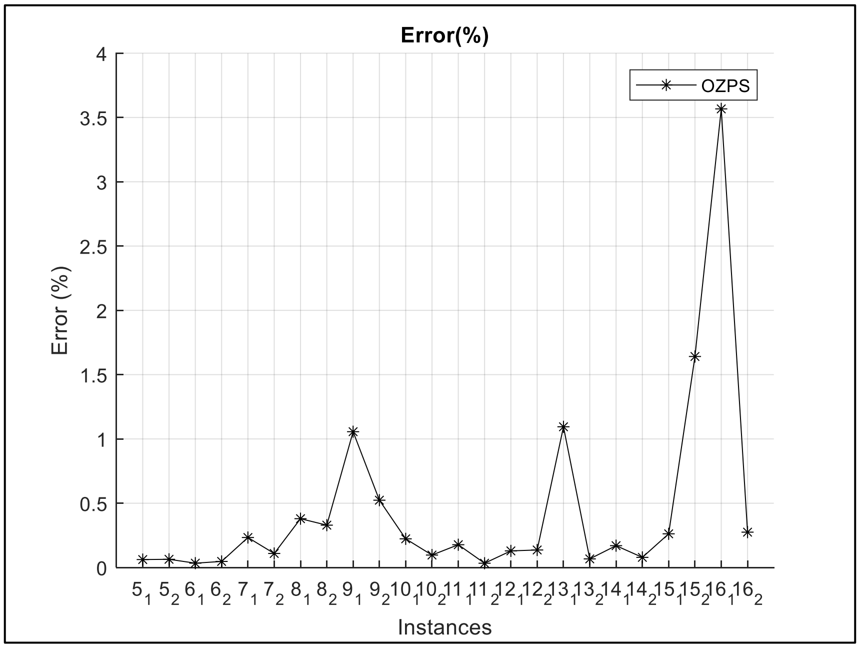

Figure 36 – OZPS error despite best

As it can be observed error committed by OZPS is under 1.5 threshold except for instances with more than 15 areas, particularly error increasing is due to areas approximation with a polygon.

Using 2D ellipses instances, we compare OZPS with CIH developed by Alatartsev et al. [32].

Results shows that our algorithm commits a percent error which is minor than CIH ones for instance with large number of areas, while for instances with small number CIH error is minor than one committed by our algorithm. Results are presented in figure 37 and in table 2.

Max error committed by OZPS (3,50) is under Max error committed by CIH (6,24).

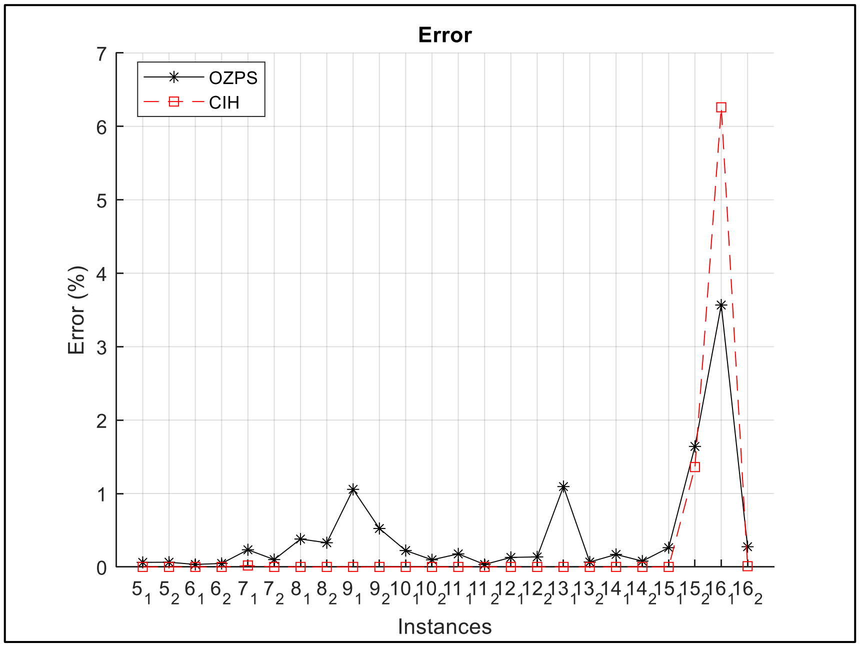

Figure 37 - Comparing errors among CIH and OZPS

Instance	Error (%)		
	Optimal Value	OZPS	CIH
tspn2DE5_1	191,255	0,061931	0
tspn2DE5_2	219,307	0,063933	0
tspn2DE6_1	202,995	0,033824	0
tspn2DE6_2	248,860	0,047361	0
tspn2DE7_1	201,492	0,235679	0
tspn2DE7_2	239,788	0,111216	0
tspn2DE8_1	190,243	0,33025	0,28
tspn2DE8_2	229,150	0,397875	0
tspn2DE9_1	259,290	1,024604	0
tspn2DE9_2	262,815	0,297977	0
tspn2DE10_1	225,126	0,3427	0
tspn2DE10_2	273,192	0,096962	0
tspn2DE11_1	247,886	0,177326	0
tspn2DE11_2	258,003	0,033406	0
tspn2DE12_1	265,858	0,129353	0
tspn2DE12_2	312,493	0,144841	0
tspn2DE13_1	278,876	1,345172	0
tspn2DE13_2	324,271	0,067075	0
tspn2DE14_1	310,794	0,16996	0
tspn2DE14_2	270,638	0,146602	0
tspn2DE15_1	289,716	0,222932	0
tspn2DE15_2	293,357	1,595773	1,36
tspn2DE16_1	369,945	3,471521	6,24
tspn2DE16_2	295,130	0,25276	0,00

Table 2- Comparing errors among CIH and OZPS

Evaluation of OZPS on Polygonal Test Instances

For the evaluation of OZPS on test instances with a generical shape, a Polygonal Test Instances has been developed.

Each polygon was created randomly with a first attempt value of 8 edges. All polygons were distributed randomly in the 2D space and some of that with overlap.

Figure 38 shows exampled with 10 and 20 areas; particularly left column represent path found in case of OPP operator enabled, while right side with OPP operator disabled.

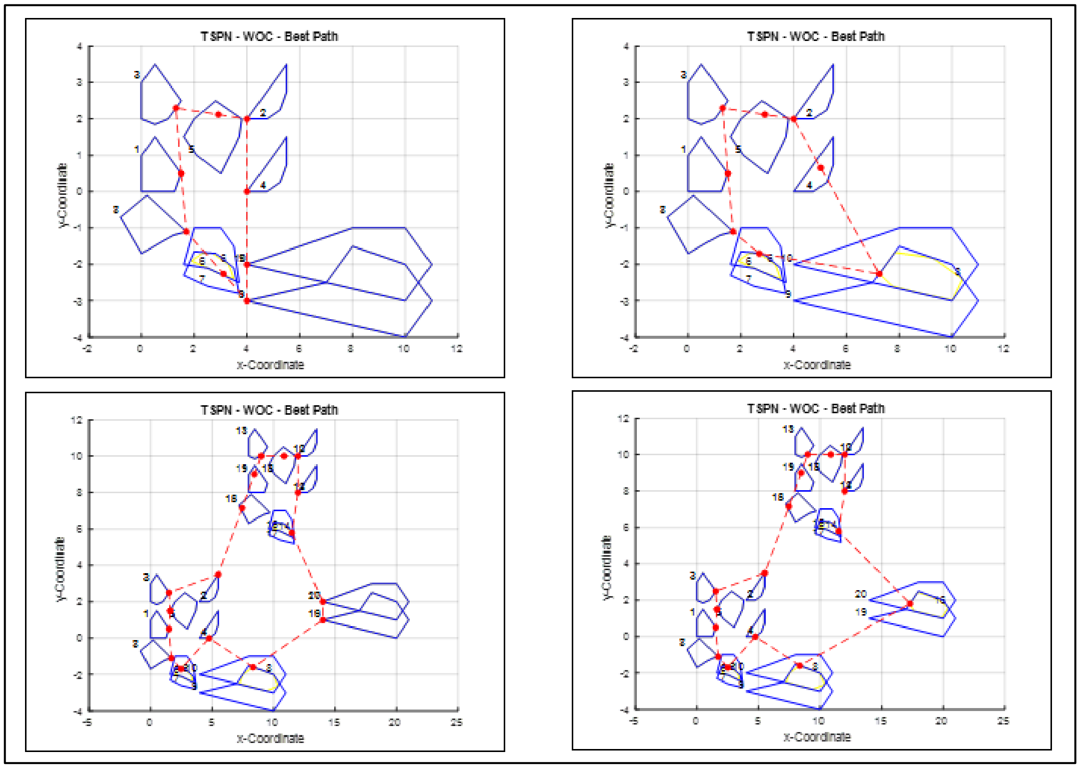

Figure 38 -(Top | Left | 10 Areas) Best path in case of OPP operator enabled - Length→14.12 ; (Top | Right | 10 Areas) Best path in case of OPP operator disabled Length→17.23; (Lower | left | 20 Areas) Best path in case of OPP operator enabled Length→41.91 ;(Lower | Right | 20 Areas) Best path in case of OPP operator disabled Length→46.78

Figure 39 results show that OZPS with OPP-operator enabled finds a shorter tour than the one found in case of OZPS with OPP-operator disabled, showing that for some configuration passing through overlap zone is not the best choice, demonstrating that takes into account overlaps position is a real important task.

As for computational time, some problem arises considering that no reference is available to test computational time performance. Moreover, computation time is strictly related to hardware available. For all these reasons it has been preferred to explain clearly and completely algorithm so that computation time can be evaluated personally by users.

4.6 Summary

It has been proposed a new method named OZPS for solving the TSPN-C-WO problem, introducing an operator able to recognize if to cross Steiner Zone. Proposed approach is based on 3-step:

1. Steiner Zone definition
2. Home Zone construction

3. TSPN resolution

OZPS was evaluated on two different test instances:

- 2D space with small numbers of ellipses
- 2D space with high numbers of polygons

Despite optimal value, as for instances with a small number of ellipses, our algorithm shows good results close to optimal, if we consider that a results approximation is due to approximating areas with a certain number of vertexes.

As for instances with high number of polygons result shows that OZPS with OPP-operator enabled finds a shorter tour than the one found in case of OZPS with OPP-operator disabled, showing that for some configuration passing through overlap zone is not the best choice demonstrating that takes into account overlaps position is a real important task.

Chapter 5.

5.AUGMENTED OZPS FOR TASK SEQUENCING OPTIMISATION

This chapter presents the proposed methodology to solve the robotic task sequencing problem using A-OZPS. Starting from Vitolo et al [12] the proposed method optimises both sequence and via-poses by augmenting T-space with robot attributes, introducing a new attribute, which name is Overall Collision Index, to evaluate the collision tendency of a pose considering overall robot.

We have tested A-OZPS for solving task sequencing of a pick and place robot especially developed.

5.1 Introduction

In Vitolo et al [12], it has been augmented EH^2C algorithm, which is a methodology to solve TSPN, for solving robotic task sequencing problem. It is based on integration of multiple attributes to identify optimal via-poses.

As in T-space there are no robot information, it has been defined an index for each attribute in order to evaluate the attribute impact on sequence of tasks.

Three robot attributes are calculated for each generated pose $\boldsymbol{\lambda}$:

(1) pose accuracy's index λA.

(2) pose reachability's index λR; and,

(3) pose collision's index λC.

Poses eligibility is calculated for each task region TR and sequence of tasks is generated via elected $\boldsymbol{\Lambda}$ which has same position $\boldsymbol{P}$ for all poses $\boldsymbol{\lambda} s$.

In this way, although there is no information on path planning, we can generate an optimal sequence which corresponds to the best *feasible* sequence.

Accuracy Index (AcI)

Accuracy index AcI aims to evaluate the quality of task execution. It is calculated as average of all pose accuracy indices λAs within set of poses **Λ**.

	$AcI = mean(\lambda A_S)$	(2)

Reachability Index (RI)

Reachability index RI aims to evaluate the feasibility degree of the poses. Solving inverse kinematics for each pose $\boldsymbol{\lambda}$ of the set **Λ**, a pose reachability λR is calculated as number of solutions by admissible solutions.

	$\lambda R = \frac{no.of\ solutions}{no.of\ admissible\ solutions}$	(3)

RI is calculated as average of all pose reachability indices λRs within set of poses **Λ**.

	$RI = mean(\lambda R_S)$	(4)

Collision Index (CoI)

Collision index aims to evaluate the collision tendency of a pose. If collision exist, count collision, not count. For each pose $\boldsymbol{\lambda}$, pose collision index λC is 1 if collision exist, otherwise 0.

	$\lambda C = \begin{cases} 1\ Collision\ True \\ 0\ Collision\ False \end{cases}$	(5)

CoI is calculated as average of all pose collision indices λCs within set of poses **Λ**.

	$$CoI = \frac{\lambda C_s}{m}$$	(6)

CoI evaluates collisions between end-effector and workpiece. There are other collisions to take into account: end-effector and obstacles; robot and obstacles; end-effector and robot; robot and obstacles; robot and workpiece. An improvement of the collision attribute is required to increase the feasibility of the solution.

5.2 Overall Collision Index

Overall Collision index aims to evaluate the collision tendency of a pose considering all robot arms and joints, which makes this index particularly interesting in some human-robot collaboration scenarios where Human-Robot contact has to be avoided or at least forecasted.

As for Collision Index (CoI), a Boolean defines if collision exists.

To forecast trajectory of each robot element (Joint or Link), it has been used Denavit & Hartemberg method. Particularly D-H method give us possibility to calculate and forecast end-effector trajectory after defining position and orientation of a pre-defined reference frame. Considered henceforward end-effector as n-element of a common industrial robot, its trajectory can be calculated by means of formula 7, where coordinate frames are attached to the joints between two links such that one transformation is associated with the joint, [Z], and the second is associated with the link [X] :

	$$[T] = [Z_1][X_1][Z_2][X_2] \dots \dots \dots [X_{n-1}][Z_n][X_n]$$	(7)

To define trajectory of a generic element m, whit m ∈ [1, n], it can consider 'to cut' formula (3), until arriving to m matrix element.

	$$[T] = [Z_1][X_1][Z_2][X_2] \dots \dots \dots [X_{m-1}][Z_m][X_m]$$	(8)

To calculate Overall collision index, and so to evaluate collision, each point whose trajectory is about to be tracked by means of (8) will be considered, depending from its geometry, approximated with a Platonic Solid and a Gilbert-Johnson-Keerthi (GJK) algorithm implemented to forecast collision between element and external object, also approximated with a platonic solid.

The Platonic Solid envelope of each joint is obtained considering as centre of the Solid the mean point of the joint and as radius the biggest distance from a point of the joint to this mean point.

Collision will be evaluated as interaction between platonic solid which represents m-joint and platonic solid which represents object.

5.3 Results and Discussion

The proposed methodology has been applied using a classical anthropomorphs robot which characteristic has been introduced in chapter 2.

The robot in question (reported in figure 39) is not a faithful reproduction of a product available on the market but a free interpretation of a revolute geometrically, cinematically and inherently compatible with reality.

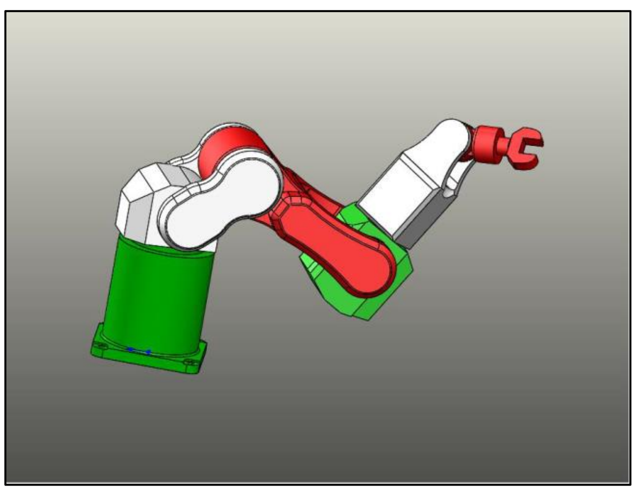

Figure 39 - Pick and Place Robot under investigation

5.3.1 Case-in-point

In literature, there is no benchmarks for task sequencing problem, this impeded us to compare our approach with existing ones. To evaluate Overall Collision Index performance three different path has been required to be followed by the robot end effector, and for each of this an object has been created and positioned randomly in the space. It has been evaluated if Overall Collision index efficiently check the collision. Below it has been reported 3d trajectories created:

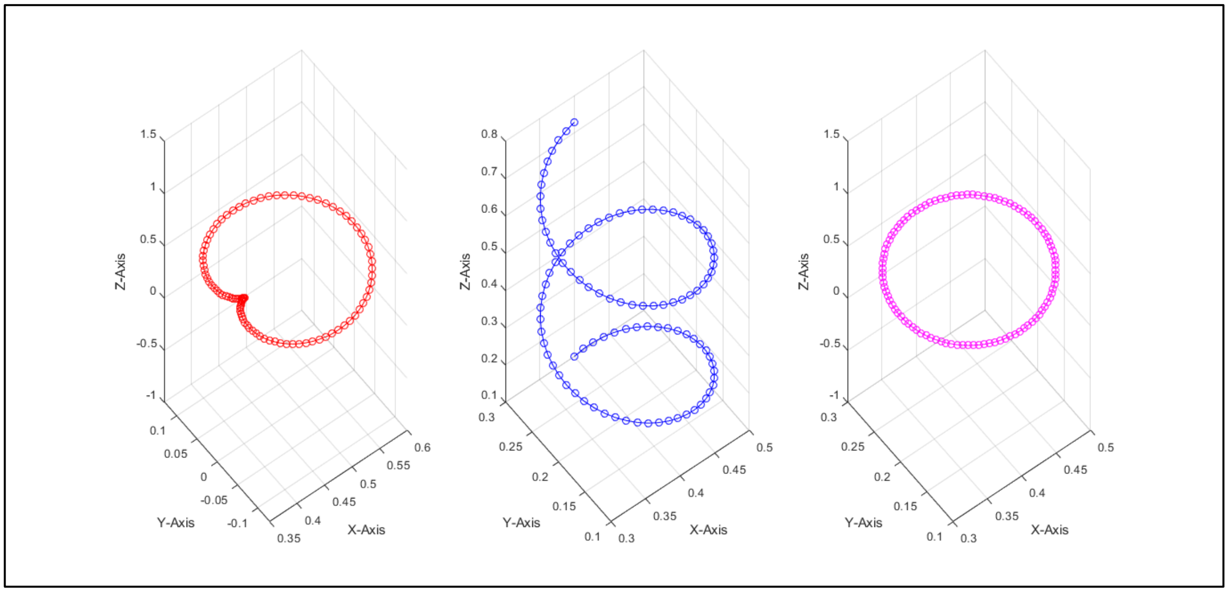

Figure 40 - End Effector Trajectories

In order to assess the benefits of a such attribute approach, we have compared path planning solutions generated after an iteration using attribute and no attribute respectively. Particularly tests on first trajectory have been explained below, while second and third one, has not been reported to not weigh down the dissertation.

First Trajectory

Test has been conducted on the first trajectory. Particularly first Joint has been isolated by means of equation 8, and trajectory studied and reported in figure below:

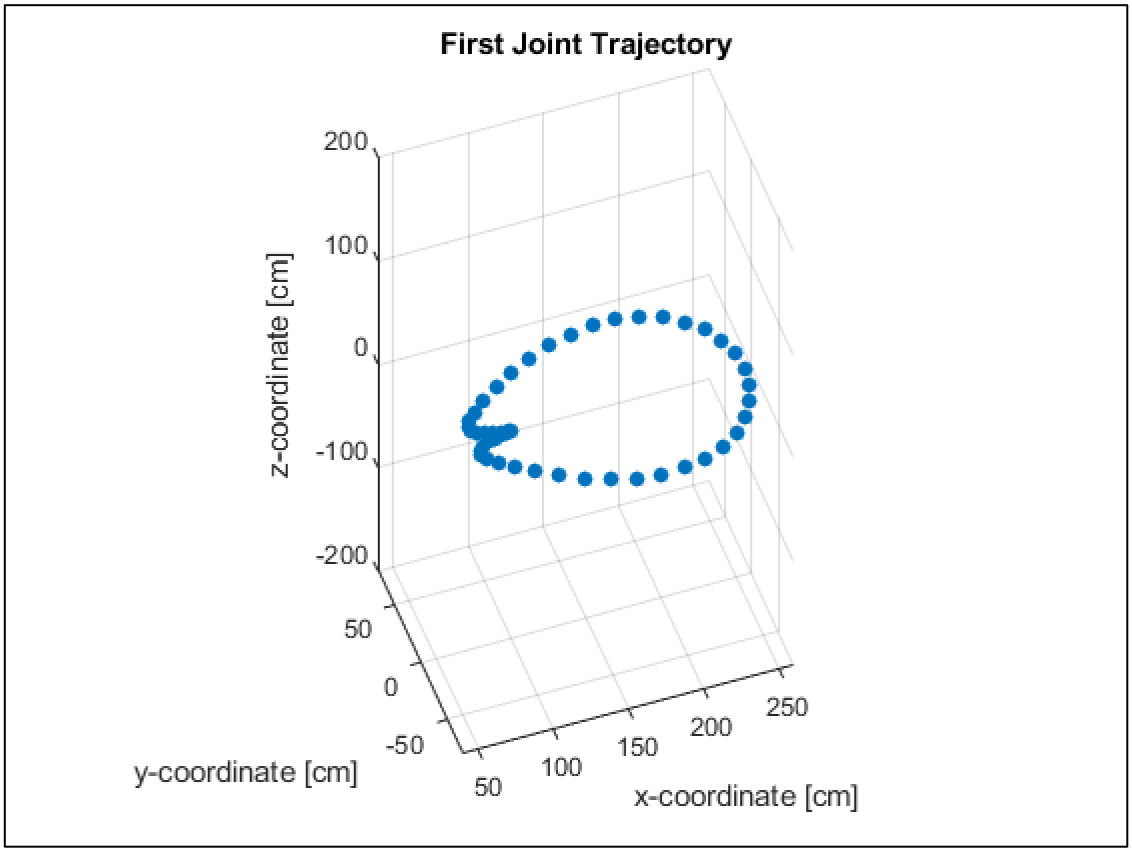

Figure 41 - First Joint Trajectory under investigation

To evaluate overall collision index performance, an object has been positioned voluntarily on the trajectory covered by the third joint in the 3D space, which has been represented by red marker in figure 42:

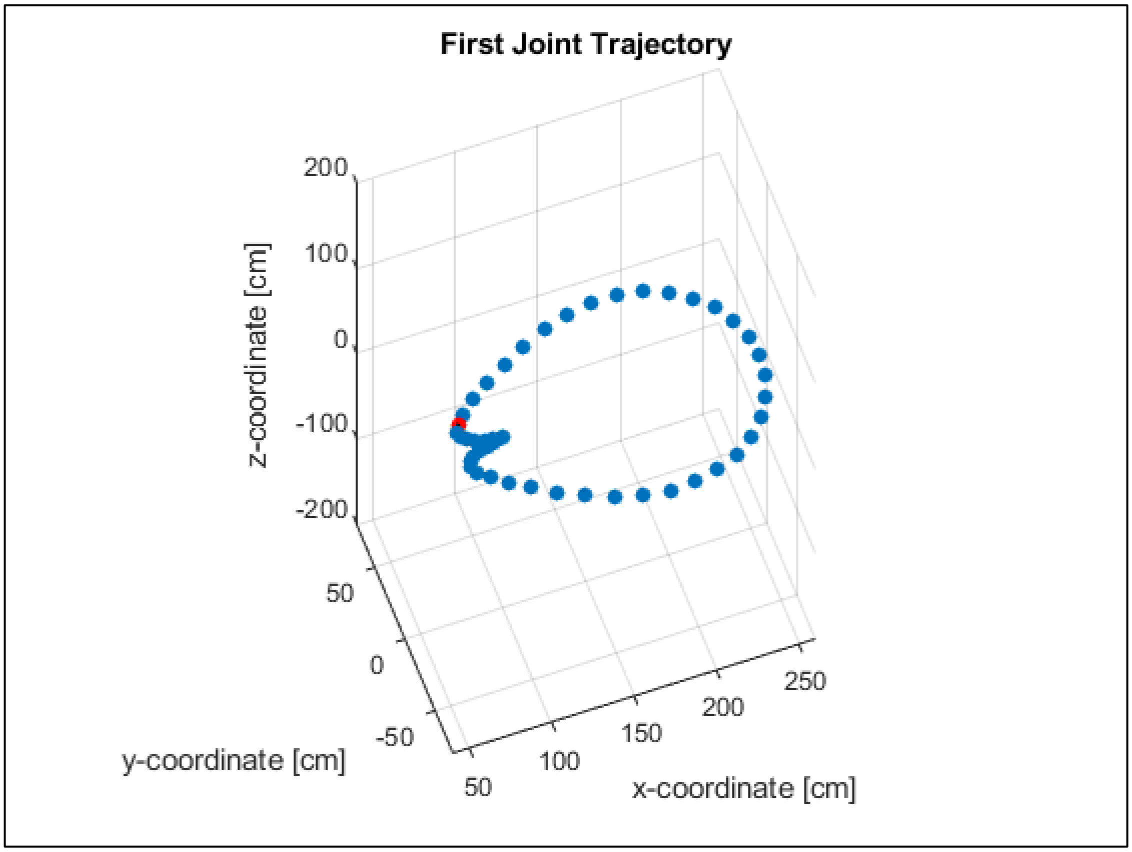

Figure 42 - First joint trajectory under investigation with obstacles enabled and stressed as red marker

When overall collision index is not enabled, collision exists but has not been detected, otherwise when index is enabled collision has checked and reported on the screen with a warning advertising, enabling evaluation for another robot path configuration; loop terminates when a free collision path will be found.

Although there exist instances for TSP, TSPN and CETSP, there is no benchmarks for task sequencing problem. This impedes to compare existing approaches. For this reason, researchers use different case study to test their approaches solving specific case as well as features and constraints. Developing a benchmark for robotic task sequencing problem is an open research question.

5.4 Summary

The proposed method defines an attribute to evaluate and forecast collision between robot and obstacles. Obstacles can be classified in two way: fixed and mobile. Our focus is on fixed obstacles, and it has been solved approximating obstacles and part of robot to investigate by means of platonic solid by applying Gilbert-Johnson-Keerthi (GJK) algorithm to solve collision problem.

Chapter 6.

6.INDUSTRIAL APPLICATION

This chapter describes the proposed methodology referred to a robot for industrial polishing to rectify the car hood of automotive SUV.

6.1 Introduction

Modern manufacturing plants are based on multi-stage processes [16] by alternating humans and robotic systems.

Researchers [17] [18] claim that workstations could be more productive by combining:

- human intelligence,
- flexibility and adaptability of the manufacturing line,
- strength, endurance and accuracy of robots.

The combination of human beneficial characteristics with modern robots opens huge possibilities to simultaneously increase productivity, reduce ergonomically bad work postures [19] and increase the perceived well-being on shift [20], achieving a full manufacturing plant improvement. The designing of manufacturing plants has to be based on three kind of workstations:

1. human workstation, station's tasks are performed only by human.
2. robotic workstation, station's tasks are performed only by robotic systems.
3. human-robot workstation, hybrid workstation where station's tasks are performed by means of human-robot collaboration (HRC) [67].

The combination of human-robot skill in a human-robot workstation, with smart sensor and instruments open large possibilities to optimize further overall process/manufacturing process.

Below a workspace scenario has been defined, which making use of human and robot skill performs an evolutionary inspection and quality control.

Traditionally, visual inspection and quality control are performed by people [68], which after inspecting object, if necessary, rectify it. Although people are very good at this task and sometimes even better than machines, they cannot work for long periods of time as their eyes get tired, muscles and tendon could be overloaded, and so need to relax [68].

Advances in image processing technology are creating new perspectives in increasing productivity, quality, efficiency and ergonomic workspace aspects.

The idea is to develop workspace in which:

- Human Task consists in marking car hood with different marker type whose position enables x.y.z coordinate that has to be rectified, while marker typology defines parameter elaboration which robot has to achieve.
    - Example: An object needs to be rectified in a particular position, so operator marks that position with an x-marker which will be recognized by a image processing tool in position and typology. Marker typology and coordinate will be sent to a Robot which process this information enabling process, whose parameter depends from marker typology (which is x but could have been another typology which could have enabled other process parameters), going to rectify objects.
- Robot Task, by using a developed image-processing techniques recognize marker in position and typology enabling a task execution optimized by means of a specific developed methodology able to find near-optimum solution, by minimising computational time to enable dynamic robot programming in the case of multiple and coupled tasks' attributes.

6.2 Proposed Workspace Scenarios

Referring to figure 42, workstation has been organized as follow:

- Operator 1, recognize area which need to be rectified marking it with a marker whose shape depends from extent of the defect, for example
    1. Δ marker refers to a defect which need to mild processing.
    2. □ marker refers to a defect which need to medium processing.
    3. ○ marker refers to a defect which need to high processing.

- Image Acquisition: A camera takes a picture to send to pc with image elaboration tool installed
- Image Elaboration: Pc received image and elaborate them defining marker position and topology
- Polishing Phase: A robot which entails a polishing system as end-effector, receives marker coordinate position and processes defects with working parameter which depends to marker typology.

Proposed workspace scheme has been reported below:

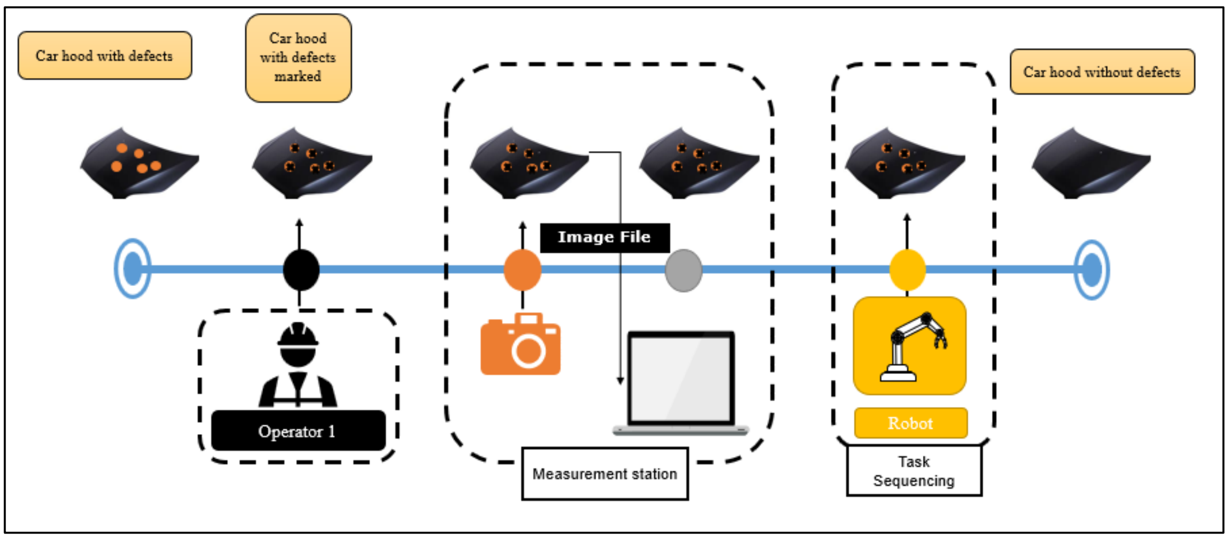

Figure 43 - Workspace Industrial Line Workflow

6.2.1 Image Segmentation Tool

Image Segmentation Tool, which works during Image Elaboration phase (please refers to figure 43), has the role to recognize marker typology affixed by human and define positions of them.

To better explain how this tool works, it will be explained by following the explanation the resolution of a test case which deals with a classical car component.

First of all, it has been reported king of image on which our algorithm works, which is a picture of a car hood marked with different kind of markers.

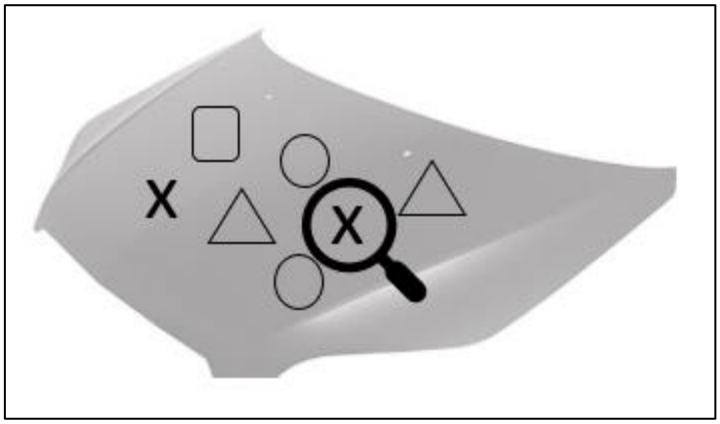

Figure 44 - Car Hood with Marker (Illustrative picture, in compliance with copyright laws)

After having translate this RGB image in a grayscale one, making a zoom on each marker it can be observed that border of each of them is not homogeneous. Lack of boundary homogeneity gives back problems in the next thresholding and binarizing steps. Lack of homogeneity is evident focusing our attention on every single marker; figure 45 reports a zoom on some of these markers:

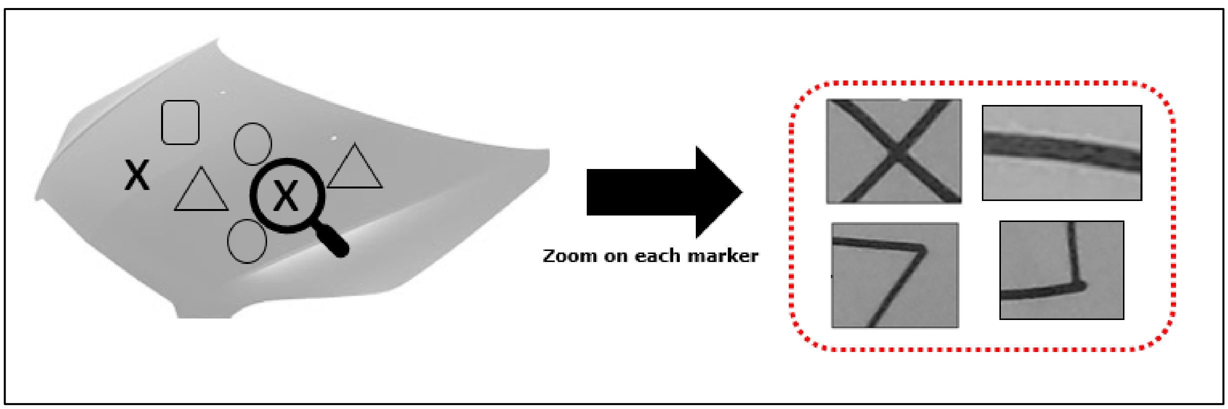

Figure 45 - Lack of Marker Homogeneity Zoom (Illustrative picture, in compliance with copyright laws)

To eliminate this lack of homogeneity a filtering operation has been achieved, by using the image grayscale histogram, which is a chart that shows the distribution of intensities in a grayscale image.

Below it has been reported Histogram relative to first marker reported:

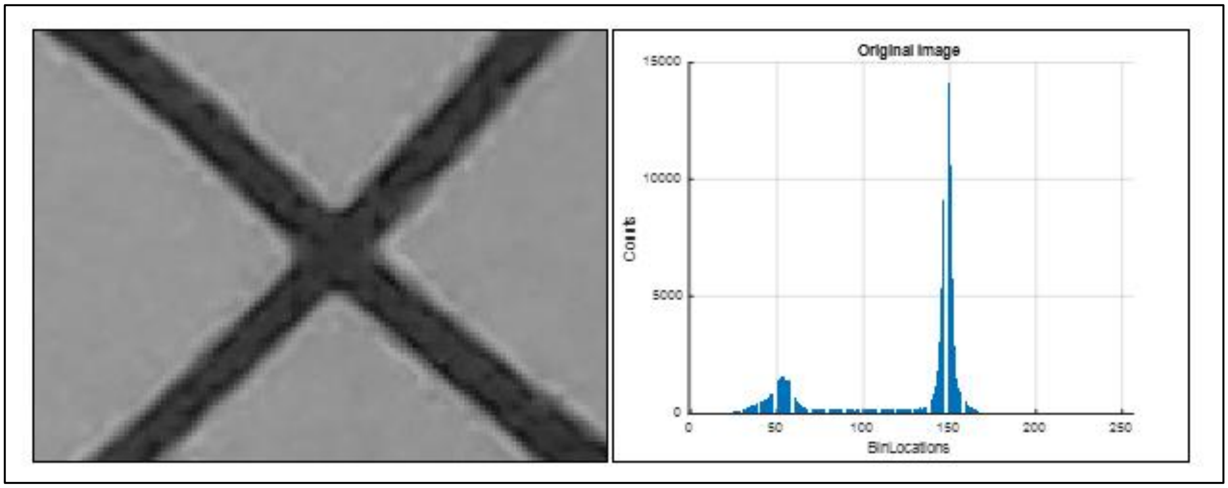

Figure 46 - left) marker under investigation right) Grayscale Histogram

Analyzing the histogram, you can recognize various peaks, which suggest the presence of pixels with the same intensity arranged along the entire image.

Allocate pixels with same intensity inside image is not a trivial task, but can be achieved remembering that histogram left side, which value is 256, represents white intensities pixels, while right side, whose extreme value is 0, represents black intensities pixels.

Referring to marker reported in image 47 (left), pixels whose value is over 100 thresholds defines gray background in the marker image (figure 47 right – gray part), while value which is under 100 threshold defines black and noise

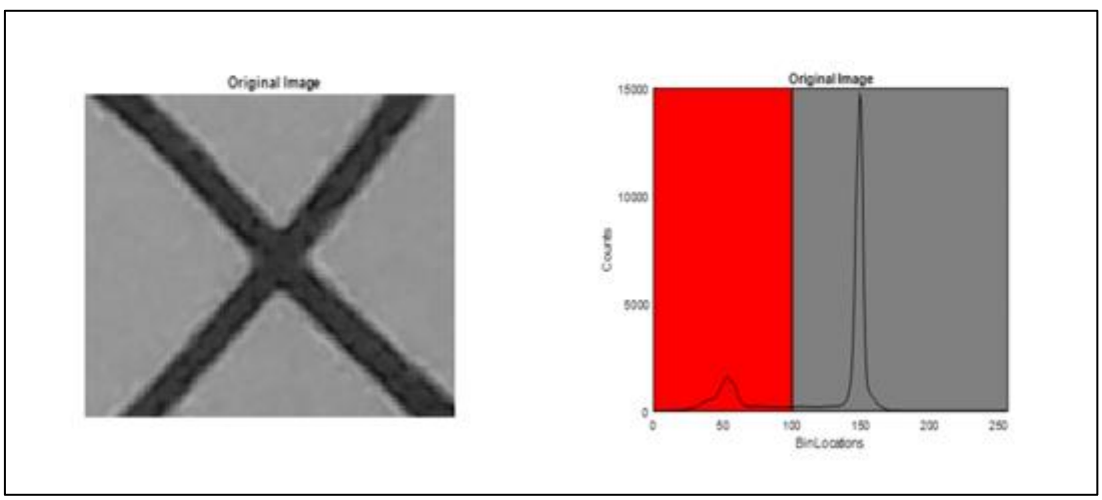

Figure 47 - left) marker under investigation right) Grayscale Histogram

To eliminate noise from marker image, all pixel whose value is under 100 thresholds, which represents noise and black part of image, has been reported to 0 value. Result of this operation, reported in figure 48, shows that marker noise has been efficiently eliminated, making marker deep and clear.

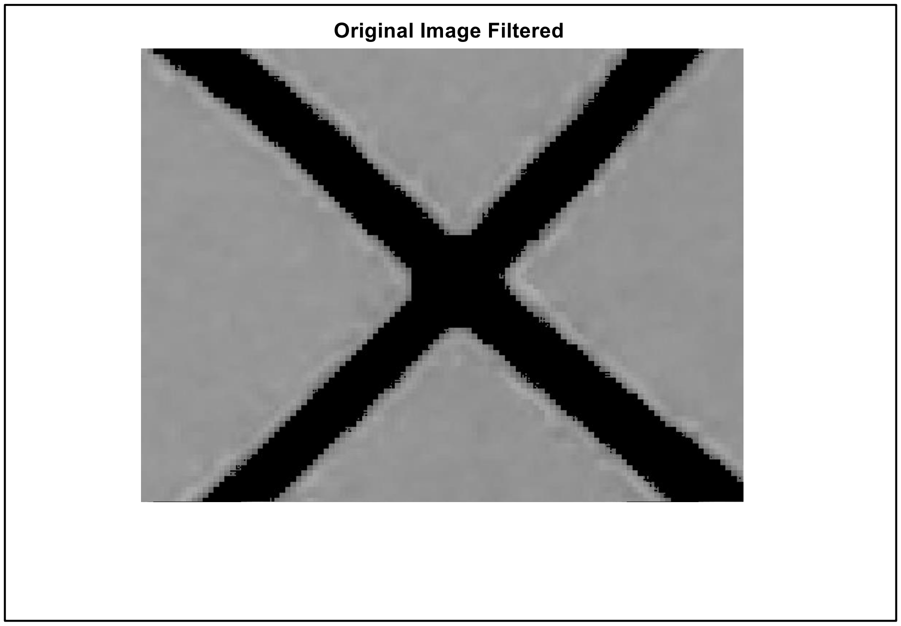

Figure 48 - Zoon on Marker after Filtering Phase

This operation has been achieved for each of the marker presents on the car hood. It has been evaluated that 100 value is the best choice to stress black in the image, because of able to clearly separate different zones in the image.

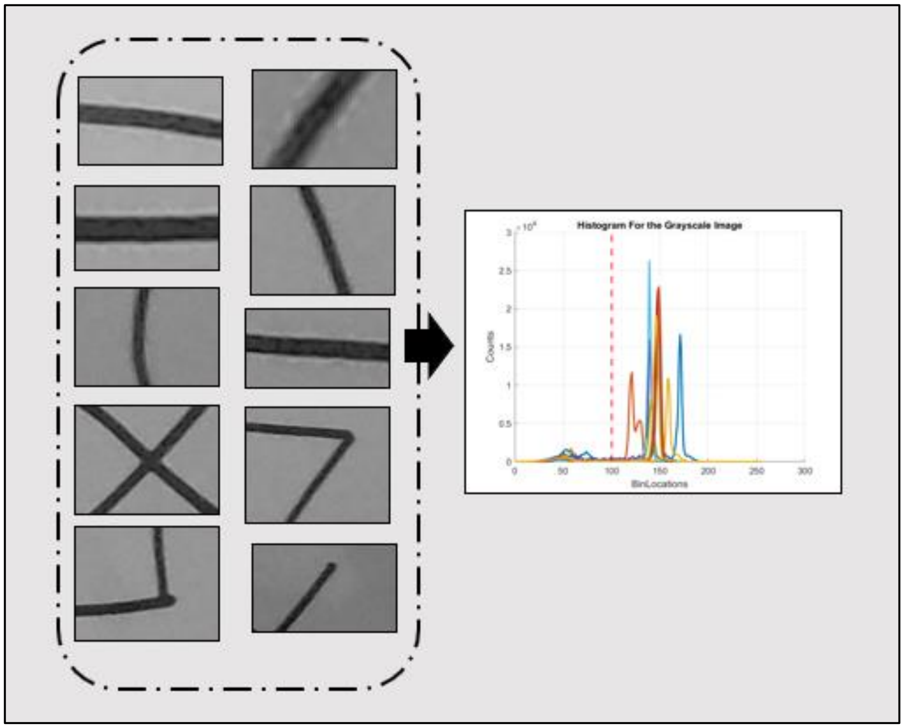

Figure 49 - Marker and Relative Grayscale Histogram

After this operation, the image is strongly divided into two parts, one of them with gray value and the second one with black value. At this point an efficiently binarizing operation can be achieved, using as thresholding value 100, achieving result reported in image 50.

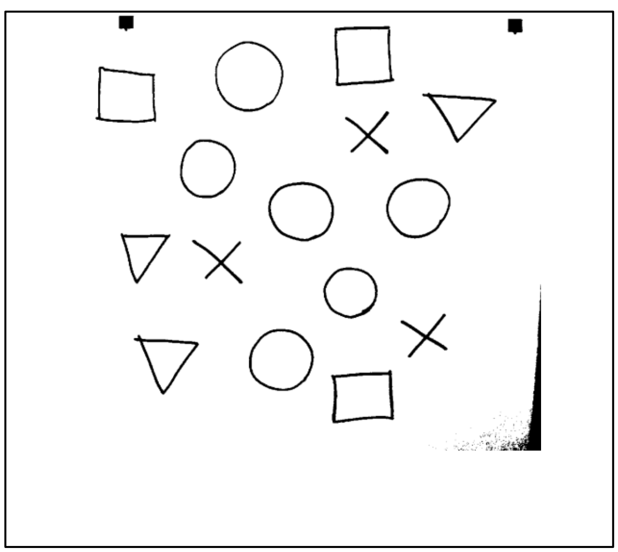

Figure 50 - Binary Image

To make our algorithm able to recognize markers' shape, is fundamental to fill each shape and filter noise that can be observed in the lower right part of the image, achieving results shown in image reported in figure 51.

Noise polishing phase filters noise but deletes x-marker; otherwise x marker due to its shape will be not recognized by algorithm and so will not be used.

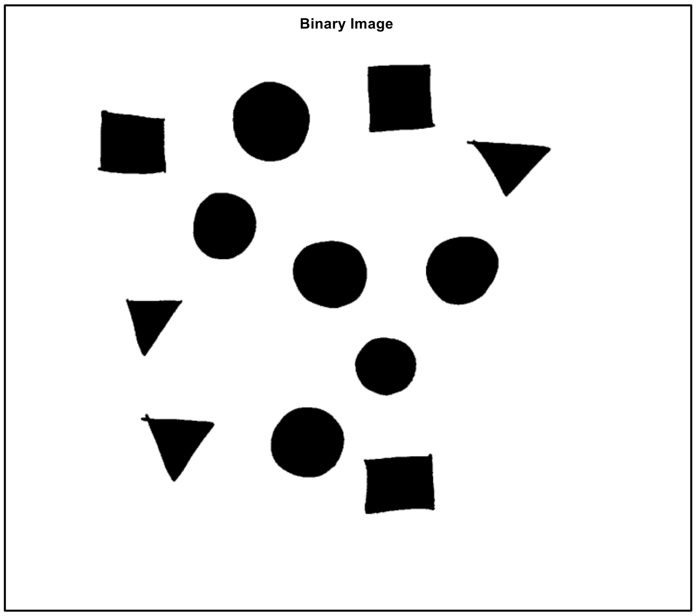

Figure 51 - Binary Image with Marker Filled

To recognize marker shape following formula has been used, which works in case some dimension shapes:

$$Circularities = \frac{Perimeter^2}{4 \times \pi \times Area}$$

Considering a pre-defined range size, a triangle is inscribed in a circle which is inscribed in a rectangle; we can recognize marker shape starting from circularities value of each marker.

Defined a marker size, threshold values have been calculated, and following algorithm has been developed in case of n-markers:

Algorithm 3: Algorithm → Overlap Zone

Input: Perimeter = {P_1, P_2,…., P_n},Area = {A_1, A_2,…., A_n},Threscold Value = {T_{MIN}, T_{MEAN}, T_{MAX}}

Output: Kind_Of_Shape = {C_1, C_2,…., C_n}

```
for i=1 to Count(P);
    C(i)←Cicularities(P(i),A(i),4,pi) ;
    if C(i) < 1.20 ;
        C(i)=Triangle;
    elseif  C(i) < 1.55;
        C(i) =Circunference;
    else
        C(i) =Rectangle;
    end
end
```

Algorithm recognition results have been reported in figure 52:

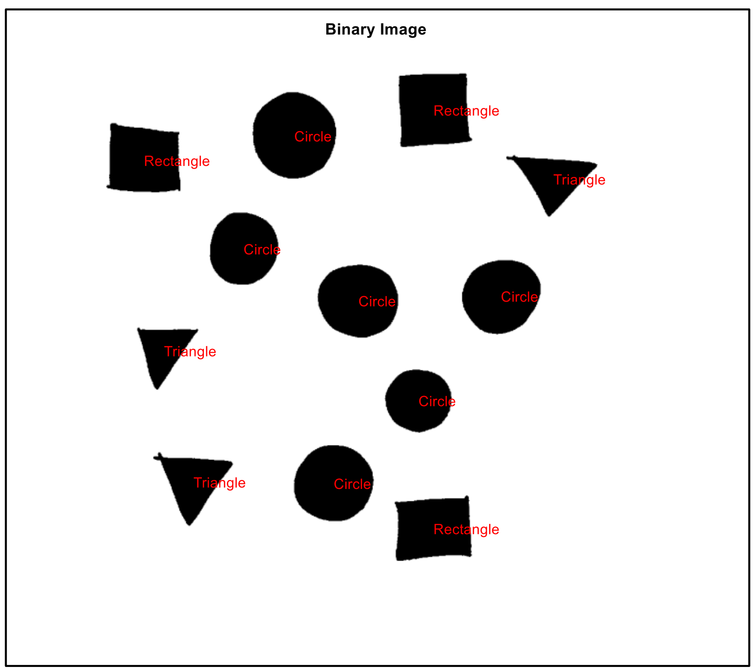

Figure 52 - Shape Detection

Position of a generical marker has been defined as number of pixels to cross along x and y axis to intersect marker centre starting from high left part of the image. In figure 53 has been reported an example, which show us how define position of an orange marker.

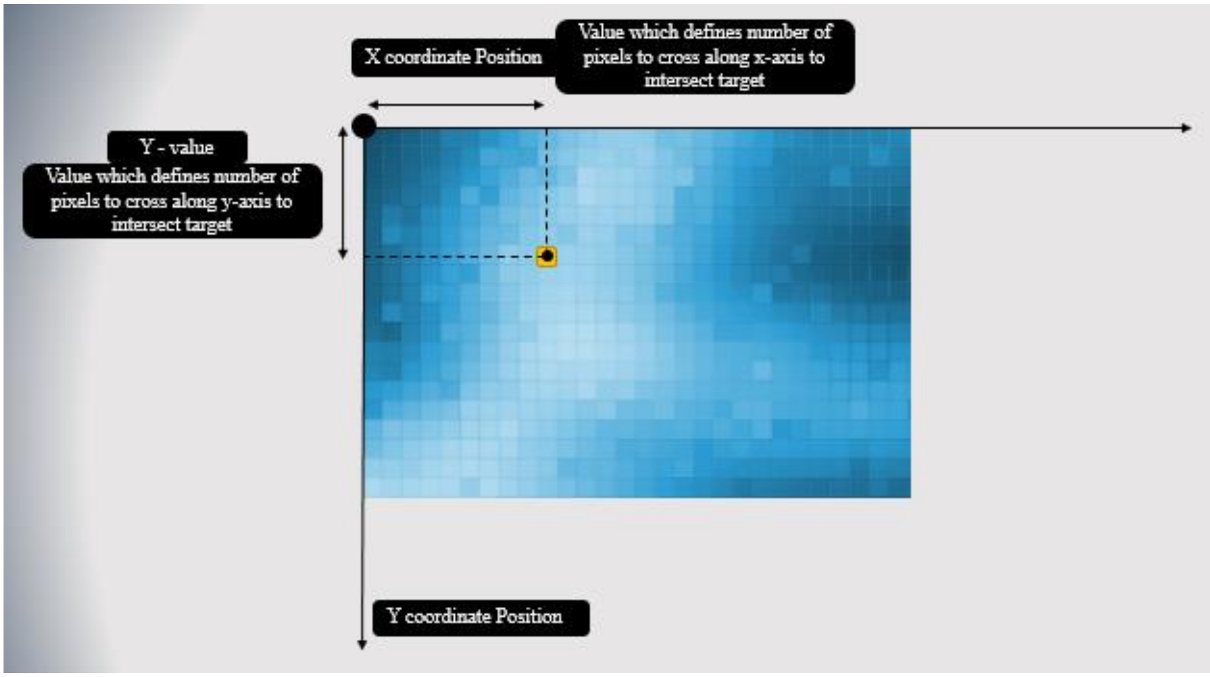

Figure 53 - Position of Orange Pixels defined in 'Pixels Coordinate Frame'

It's necessary calibrate camera to convert object position from pixels-coordinate to cm-x/y coordinate. Calibrating phase consists of the following steps:

1. Distance in [cm] between camera and our industrial target has been measured and fixed, which henceforth will be named Industrial Distance
2. An object has been photographed from Industrial Distance, whose position in [cm] are known before [x_{cm}, y_{cm}]
3. Photo dimension in [cm] has been calculated [Dimx, Dimy]
4. Pixels object position has been calculated [p_x, p_y]
5. Photo dimension in pixels has been measured [PDimx, PDimy]

Following formula has been developed, to achieve e conversation factor which permits us to convert object position from pixel to cm.

Firs of all, following proportion has been calculated:

	$p_x : PDim_x = x_{cm} : Dim_X$	(9)

Which results in:

	$x_{cm} = \frac{p_x \times Dim_x}{PDim_x}$	(10)

Then a conversation factor has been calculated

	$factor = \frac{Dim_x}{PDim_x}$	(11)

Following formula, with conversation factor just calculated, permits us to convert position of an object from pixel coordinate value to cm coordinate value:

	$x_{cm} = p_x \times factor$	(12)

Some process can be developed for y coordinate

	$p_x : PDim_x = x_{cm} : Dim_X$	(13)

Which results in:

	$x_{cm} = \frac{p_x \times Dim_x}{PDim_x}$	(14)

A conversation factor has been calculated

	$factor = \frac{Dim_x}{PDim_x}$	(15)

Following formula, whit conversation factor just calculated, permits us to convert position of an object from pixel coordinate value to cm coordinate value:

	$x_{cm} = p_x \times factor$	(16)

Some test example has been performed to verify that this formula gives back correct [] position.

Using this conversation factor, it can be achieved a conversation between object position in term of pixels to object position in term of cm.

Finally, depending from kind of shape detected, a certain group of working parameters has been activated, as example following scheme can be adopted:

4. Δ marker refers to a defect which need to mild processing.
5. □ marker refers to a defect which need to medium processing.
6. ○ marker refers to a defect which need to high processing.

The proposed methodology has been referred to a robot for industrial polishing of a automotive car hood.

Car hood is one of the key elements in automotive industry and its polishing is important to reduce defect such as 'Puntinatura', which if remained un-rectified, compromises customers quality perception.

'Puntinatura' rectification is commonly carried out to improve quality and avoid to rejects component, reducing waste of material.

For this reason, is crucial to be able to rectify a car hood, in short time and with a high accuracy.

6.2.2 System\Cell Description

The polishing system entails as end-effector on a 6-axis UR10 robot:

- AOK (Active Orbital Kit)
- ACF - ACTIVE CONTACT FLANGE - Interactive compensation for surface tolerances up to 100 mm with guaranteed consistent contact force

Below it has been reported end-effector working parameters:

End Effector

Parameter	Value []
Speed	4,000–8,000 rpm
Orbit	3.0 mm (1 ⁄ 8 ″)
Weight	0.57 kg

ORBITAL SENDER

Parameter	Value []
Max. Force (push/pull) [N]	100
Stroke [mm]	11.5
Max. overturning moment [Nm]	25
Max. torsional moment [Nm]	35

Interactive compensation for surface tolerances up to 100 mm with guaranteed consistent contact force.

ACF - ACTIVE CONTACT FLANGE

Figure 54 - End Effector Instruments Data

6.2.3 Task Planning

Task region is the locus of feasible points, which can be visited by the end-effector to perform a task.

For a polishing system, the task corresponds to feasible polishing area, i.e. the locus of feasible polishing points Pc.

Task region is fixed and will be defined as a rectangle around marker centre whose dimension depends from orbital sender dimension, while contact pressure depends from marker typology.

6.2.4 Task Sequencing

Firstly, we have characterized robot attribute based on the described polishing system.

Overall Collision Index

To calculate Overall collision index, and so to evaluate collision, each point whose trajectory is about to be tracked by means of (4) will be considered, depending from its geometry, approximated with a Platonic Solid and a Gilbert-Johnson-Keerthi (GJK) algorithm implemented to forecast collision between element and external object(which will be approximated by mans of a Platonic Solid).

The Platonic Solid envelope of each joint is obtained considering as centre of the solid the mean point of the joint and as radius the biggest distance from a point of the joint to this mean point.

Collision will be evaluated as interaction between platonic solid represented m-joint and platonic solid represented object

Collision is defined as binary state: if collision exist 1 otherwise 0.

A summary of overall cell configuration has been reported in figure (55).

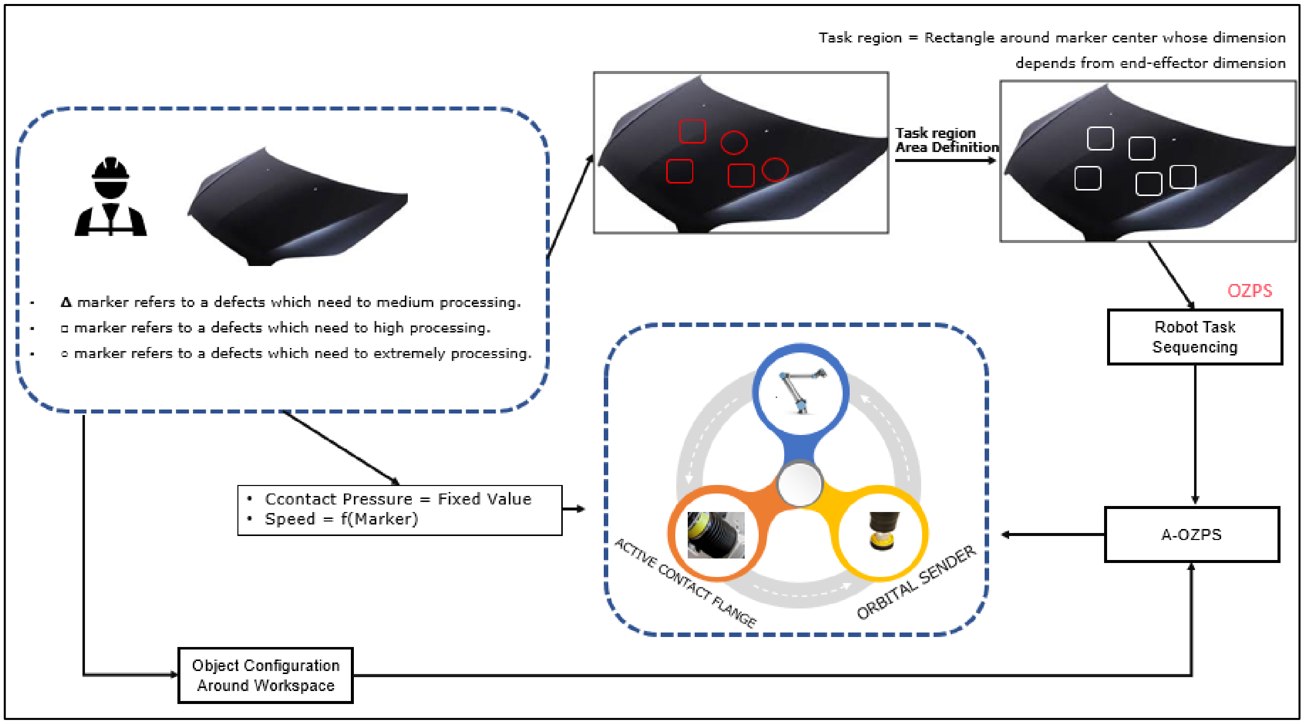

Figure 55 - Industrial Application Flowchart

6.3 Cell Application & Forecasted Impact

Workspace proposed in this dissertation can be applied to a variety of industrial task, whose impact results in an improvement in worker's working conditions, providing an ergonomic and productive workplace ensuring achievement of the goals and objectives of the organization.

In the table below it has been reported some possible workspace applications, evaluating expected industrial and ergonomic impacts:

Scenario	Human Task	Robot Task	Industrial Impact expected	Ergonomic Impact expected
Polishing	The human specifies location to polish by means of a marker, during run-time	The Robot polishes in specified location defined by marker	• Overall Process Optimization • Quality improvement • Task Time Reducing	Overall improvement of human working condition by reducing ERF
Drilling	The human specifies the drill location by means of a marker, during run-time	The Robot drills a hole in specified location defined by marker while having motion automatically constrained to drill bit's axis.	• Overall Process Optimization • Quality improvement • Task Time Reducing	Overall improvement of human working condition by reducing ERF
Surface Finish	The human specified surface to sand, by means of a marker	The Robot sands the surface in a location specified by marker, while having motion automatically constrained parallel to the surface.	• Overall Process Optimization • Quality improvement • Task Time Reducing	Overall improvement of human working condition by reducing ERF

Table 3 – Workspace Application and Impact Expected

Chapter 7

7.CONCLUSIONS AND FUTURE WORKS

This chapter briefly summarises the present research and points out the main contributions.

7.1 Conclusions & Key Findings

The dissertation addresses the following questions:

How to define a new kind of workstation which can be applied to a wide range of industrial application to achieve:

- Improvement in ergonomic workspace aspects
- Reduction in defects number and impact
- Optimization in robot task execution by means of a methodology minimising computational time to enable dynamic robot programming in the case of multiple and coupled tasks' attributes

Starting from this question, following question have been addressed:

1. How to find optimal task sequence in case of not simple shape with partial or total overlap using Euclidean distance as key metric
    a. Define an operator able to define when to pass through overlap zone
2. How to define attribute to evaluate and forecast collision between robot and human, considering all robot part, starting from multi-attribute approach to find optimized task sequencing
3. How to use computer-based method to develop a tool for image recognition on a reflecting surface

Particularly it has been proposed a new method named OZPS to solve the task sequencing problem using the Travelling Salesman Problem with Neighbourhoods (TSPN) in its most complex form which is TSPN-C-WO. A distinctive feature of our solution is that algorithm, thanks to a special operator named OPP-operator, is able to define if passing through Overlap Zone can lead the solution being the best in term of path.

According to the evaluation on instances with known optimal values, our method can obtain a solution close to the optimum, despite an error caused by approximating boundary of shapes under investigation with polygon.

According to the evaluation on our own instances developed to test OPP operator, OZPS confirms our hypothesis that passing through overlap zone is not always the best choice, using path length as key performance indicator.

The dissertation evaluates collisions between end-effector and workpiece, proposes a new attribute for solving robotic task sequencing which is particularly adapted to solve problem in case of collaborative workspace, to forecast and avoid collision between every part of robot and all possible fixed obstacle in the workspace.

The proposed method can be exploited to any industrial robotic system carrying out repetitive tasks; besides, it can be used for sequencing of those tasks which require automatic tool movements.

7.2 Further Works

Further research is clearly needed in the task sequencing and workspace domain . This section provides several possible routes for researchers involved in robotic task sequencing.

7.2.1 OPP Improvement & Task Benchmarks

The proposed method for TSPN-C-WO points out good results, operator for OPP seems to accomplish work which is required to, but no Benchmark Test Instances are available so it’s not possible to evaluate if path found is the best one, so first of all it could be interesting to create a Test Benchmark. Otherwise OPP has been just introduced in this dissertation, further investigation is needed to better understand the effect of shape position on overlap zone.3D space is not taken into account in this dissertation, so further investigation is needed to apply this method to 3d space.

7.2.2 Workspace improvement

The proposed workspace neglected possibility of Robot and human close-proximity interaction due to safety reason, as a matter of fact workspace design provide a physical separation between human and robot. Future works should introduce a Human Robot collaboration scenarios, where a collaborative robot (Cobot) and a human occupy the some workspace interacting to accomplish tasks. A scenario like this can give us possibility to enhance collaboration level introducing new collaboration features.

APPENDIX 1

Do define points inside Circumference has been searched:

$$(x - xc)^2 + (y - yc)^2 < r \qquad (1)$$

Otherwise, if point inside Ellipse has been searched, it has been used this equation:

$$\frac{(x - xc)^2}{a} + \frac{(y - yc)^2}{b} < 1 \qquad (2)$$

As defined before, one time that all points inside different forms has been discovered, we define the part of plane which represents points inside the shape with value 1 and points outside with value zeros. Below a figure which can simply explain this concept.

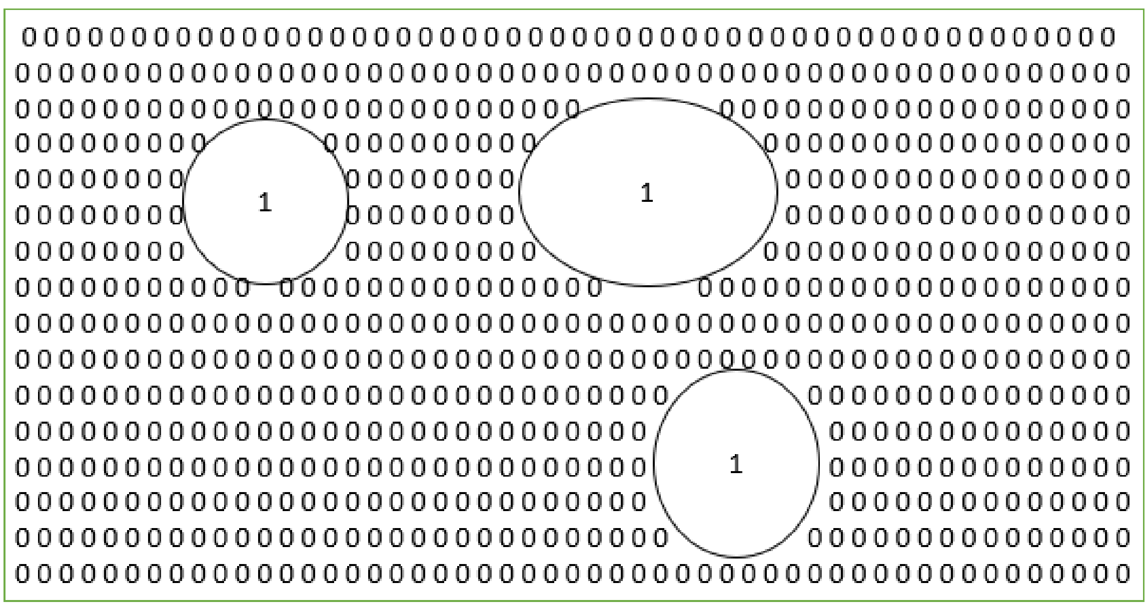

Figure 56 - Overlap Calculation Scheme

In the zone where shapes are in overlap, also the one-value are in overlap and will be sums up becoming 2 or 3 …4 depending from how many shapes are in overlap (2 represents two shapes which are in overlap,3 represents three shapes which are in overlap… etc).

Bibliografia

[1] M. Luenendonk, «Industry 4.0: Definition, Design Principles, Challenges, and the Future of Employment,» Cleverism, [Online]. Available: https://www.cleverism.com/industry-4-0/.

[2] D. Lukač, The fourth ICT-based industrial revolution" Industry 4.0"??? HMI Industry 4.0, 2015 November.

[3] «Consortium II Fact Sheet,» 2013. [Online]. Available: http: //www.iiconsortium.org / docs /IIC_FACT_SHET.pdf.

[4] K. W. Henning e H. Johannes, Recommendations for implementing the strategic Industry 4.0.

[5] M. Hermann, T. Pentek e B. Otto, Design Principles for Industrie 4.0 Scenarios, Hawaii Inernational Conference on Syste Sciences (HICSS), (2016, January).

[6] M. Broy, M. V. Cengarle e E. Geisberger, «Cyber-Physical Systems: Imminent Challenges, Large-Scale Complex IT Systems,» *Development,Operation and Management*, pp. 1-28, (2012).

[7] H. Kagermann, W. Wahlster e J. Helbig, Recommendations for implementing the strategic initiative Industrie 4.0, (2013).

[8] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn e K. Ueda, «Cyber-physical systems in manufacturing,» *CIRP Annals,65*, pp. 621-641, (2016).

[9] H. Lasi, P. Fettke, H. G. Kemper, T. Feld, M. Hoffmann e M., “Industry 4.0”. Business & Information Systems Engineering 6 (4) 239, 2014.

[10] «International Federation of Robotics,» [Online]. Available: https://ifr.org/ifr-press-releases/news/world-robotics-report-2016.

[11] J. Shi e S. Zhou, «Quality control and improvement for multistage systems: A survey,» *IIE Trans*, vol. 41, n. 9, p. 744–753, June 2009.

[12] F. Vitolo, Multi-attribute task sequencing optimisation with neighbourhoods for robotic systems, University of Naples Federico II: PhD dissertation, 2017.

[13] P. T. Zaccaria, E. K. Xidias e N. A. Aspragathos, «Task scheduling and motion planning for an industrial manipulator,» *Robotics and computer-integrated manufacturing,* vol. 29 , n. 6, pp. 449-462, 2013.

[14] Z. Pan, J. Polden, N. Larkin, S. V. Duin e J. Norrish, «Recent progress on programming methods for industrial robots,» *Robotics and Computer Integrated Manufacturing 28,* pp. 87-94, 2012.

[15] E. Kolakowska, S. F. Smith e M. Kristiansen, «Constraint optimization model of a scheduling problem for a robotic arm in automatic systems,» *Robotics and Autonomous Systems 62,* pp. 267-280, 2014.

[16] P. Franciosa, A. Palit, F. Vitolo e D. Ceglarek, «Rapid Response Diagnosis of Multi-stage Assembly Process with Compliant non-ideal Parts using Self-evolving Measurement System,» *Procidia CIRP,* vol. 60, pp. 38-43, 2017.

[17] J. Krüger, B. Nickolay, P. Heyer e G. Seliger, «Image based 3D Surveillance for flexible Man-Robot-Cooperation,» *CIRP Annals Manufacturing Technology,* vol. Vol. 54, pp. 19-22, 2005.

[18] E. Helms, R. D. Schraft e M. Hagele, «Robot assistant in industrial environments,» *Proceedings of the 11th IEEE International Workshop on Robot and Human Interactive Communication,,* pp. 399-404, 2002.

[19] J. Kruger, T. Lien e A. Verl, «Cooperation of human and machines in assembly lines,» *Keynote paper, CIRP Annals – Manufacturing Technology,* vol. Vol. 58, pp. 628-646, 2009.

[20] D. Bortot, M. Born e K. Bengler, «Directly or on Detours? How Should Industrial Robots Approximate Humans?,» *Journal of Experimental Psychology*, vol. 55(4), p. 352–358, (2013).

[21] E. Kolakowska, S. F. Smith e M. Kristiansen, «Constraint optimization model of a scheduling problem for a robotic arm in automatic systems,» *Robotics and Autonomous Systems 62*, pp. 267-280, 2014.

[22] B. Siciliano, L. Sciavicco, L. Villano e G. Oriolo, «Robotics: modelling, planning and control,» *Springer Science % Business Media*, n. ISBN 978-1-84628-641-4, 2010.

[23] Z. Ermin, P. Xiaojian e Z. Yabo, «The research of the trajectory of industrial robots modeling simulation based on DELMIA,» *Modern Manufacturing Engineering 3 007,* 2013.

[24] M. Z. Nida e J. A. Musbah, «Survey on Image Segmentation Techniques,» *Procedia Computer Science*, vol. 65, p. 797 – 806, 2015.

[25] [Online]. Available: https://it.mathworks.com/discovery/image-segmentation.html.

[26] N. Jaffar, A. H. Abdul-Tharim, I. F. Mohd-Kamar e N. S. Lop, «A Literature Review of Ergonomics Risk Factors in Construction Industry,» *The 2nd International Building Control Conference 2011.*

[27] M. Rebi e A. Rami, «Managing ergonomics risk factors at workplace,» *NIOSH 6th National Conference and Exibition on Occupational Safety and Health,* 2003.

[28] E. Yelin, L.S.Trupin e D. Sebesta, «Transitions in employment, morbidity and disable among persons aged 51 to 61 with musculosketal and non-musculosketal conditions in U.S.,» *Arthitis and Rheumatism, 42,* pp. 769-779, 1992-1994.

[29] A. Praemer, S. Furner e D. P. Rice, «Musculosketal Conditions in the United States,» *Rosemont,IL: american Academy of Orthopaedic Surgeons,* 1999.

[30] M. Jünger, G. Reinelt e G. Rinaldi, «The traveling salesman problem". Handbooks in operations research and management science 7,» pp. 225-330, 1995.

[31] D. S. Johnson e L. A. McGeoch, «The traveling salesman problem: A case study in local optimization,» *Local search in combinatorial optimization 1*, pp. 215-310, 1997.

[32] S. Alatartsev e F. Ortmeier, «Constricting Insertion Heuristic for Traveling Salesman Problem with Neighborhoods,» January 2013.

[33] G. Laporte e Y. Nobert, «Generalized Travelling Salesman Problem Through n sets of nodes: An Integer Programming Approach,» *INFOR: Information Systems and Operational Research 21(1)*, pp. 61-75, 1983.

[34] M. Saha, T. Roughgarden, J. Latombe e G. Sanchez-Ante, «Planning tours of robotic arms among portioned goals,» *The International Journal of Robotics Research 25 (3)*, pp. 207-223, 2006.

[35] C. Wurll, D. Henrich e H. Worn, «Multi-Goal Path Planning for Industrial Robots,» *In International Conference on Robotics and Application,* 1999.

[36] E. M. Arkin e R. Hassin, «Approximation algorithms for the geometric covering salesman problem,» *Discrete Applied Mathematics,* vol. 55 (3), pp. 197-218, 1994.

[37] I. Gentilini, K. Nagamatsu e K. Shimada, «Cycle time based multi-goal path optimization for redundant robotic systems,» *In IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[38] K. Vicencio, B. Davis e I. Gentilini, «Multi-goal path planning based on the generalized traveling salesman probleneighborhoods,» *IEEE/RSJ International conference on intelligent robots and systems (IROS),* 2014.

[39] K. Vicencio, T. Korras, K. A. Bordignon e I. Gentilini, «Energy-optimal path planning for six-rotors on multi-target missions,» *IEEE/RSJ International conference on intelligent robots and systems (IROS),* pp. 2481-2487, 2015.

[40] A. Kovacs, «Integrated task sequencing and path planning for robotic remote laser welding,» *International Journal of Production Research,* vol. 54 (4), pp. 1210-1224, 2016.

[41] Mennell e K. W, Heuristics for Solving Three Routing Problems: Close-E- nough Traveling Salesman Problem,Close-Enough Vehicle Routing Problem, Se- quence-

Dependent Team Orienteering Problem., Maryland PhD dissertation: University of Maryland, 2009.

[42] W. Xingyin, G. Bruce e W. Edward, «A Steiner Zone Variable Neighborhood Search Heuristic for the Close-Enough Traveling Salesman Problem,» *Computers and Operations Research.*

[43] Gentilini, I. Margot e F. Shimada, «The Travlling salesman problem with neighbourhoods:MINLP solution,» *Optimization Methods and Software 0,* p. 1–15.

[44] Oberlin, P, S. Rathinam e S. Darbha, «Today’s traveling salesman problem,» *IEEE Robotics & Automation Magazine 17(4),* p. 70–77, 2010.

[45] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu e Wang, «Particle swarm optimization-based algorithms for TSP and generalized TSP.,» *Information Processing Letters 103(5),* p. 169–176., 2007.

[46] M. Dror, A. Efrat, A. Lubiw e J. S. B. Mitchell, «Touring a sequence of polygons,» *35th annual ACM symposium,* vol. ACM Press, p. 473–482 , 2003.

[47] Q. Yuanlong e Y. Bo, «ACO-iRBA: A Hybrid Approach to TSPN with Overlapping Neighborhoods».

[48] Y. Zhao, X. Ming-Qing, Y.-W. Ge, F. Delong, Z. Lei, S. Hai-Fang e T. Xi-Lang, «A double-loop hybrid algorithm for the traveling salesman problem with arbitrary neighbourhoods,» *European Journal of Operational Research.*

[49] Gulczynski, J. D, Heath, W. J, Price e C. C, «The close enough traveling salesman problem: A discussion of several heuristics,» *Perspectives in Operations Research: Papers in Honor of Saul Gass’ 80th Birthday,* p. 271–283, 2006.

[50] B. Yuan, M. Orlowska e S. Sadiq, «On the optimal robot routing problem in wireless sensor networks,» *IEEE Trans. Knowl. Data Eng. 19 (9),* p. 1252–1261, 2007.

[51] Mennell, K. W, B. Golden e E. Wasil, « Steiner-zone heuristic for solving the close-enough traveling salesman problem.,» *Operations Research, Computing, and Homeland Defense* ., p. 162–183, 2011.

[52] H. Chen, T. Fuhlbrigge e X. Li, «Automated industrial robot path planning for spray painting process: a review,» *In Automation Science and Engineering*, pp. 522-527, 2008.

[53] C. Hofner e G. Schmidt, «Path planning and guidance techniques for an autonomous mobile cleaning robot,» *Robotics and autonomous systems 14,* pp. 199-212, 1995.

[54] X. Wang, Y. Shi, D. Ding e X. Gu, «Double global optimum genetic algorithm–particle swarm optimization-based welding robot path planning,» *Engineering Optimization 48,* pp. 299-316, 2016.

[55] A. Kovac, «A task sequencing for remote laser welding in the automotive industry,» *International Conference on Automated Planning and Sceduling (ICAPS),* pp. 457-461, 2013.

[56] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans e D. Lane, «Path planning for autonomous underwater vehicles,» *IEEE Transactions on Robotics 23,* pp. 331-341, 2007.

[57] S. N. Spitz e A. A. Requicha, «Multiple-goals path planning for coordinate measuring machines,» *IEEE International Conference on Robotics and Automation (ICRA) 3,* pp. 2322-2327, 2000.

[58] B. Siciliano, L. Sciavicco, L. Villano e G. Oriolo, «Robotics: modelling,planning and control,» *Springer Science Business Media,* 2010.

[59] L. Gueta, R. Chiba, J. Ota, T. Ueyama e T. Arai, «Coordinated motion control of robot arm and a positioning table with arrangement of multiple goals,» *IEEE International Conference on Robotics and Automation,* pp. 2252-2258, 2008.

[60] S. Alatartsev, S. Stellmacher e F. Ortmeier, «Robotic task sequencing problem: a survay,» *Journal of Intelligent & Robotic Systems,* vol. 80, pp. 279-298, November 2015.

[61] C. Wurll, D. Henrich e H. Worn, «Multi-Goal Path Planning for Industrial Robots,» *In International Conference on Robotics and Application,* 1999.

[62] J. Faigl, V. Vonasek e L. Preucil, «A multi-goal path planning for goal regions in the polygonal domain,» *In European Conference on Mobile Robots (ECMR),* pp. 171-176, 2011.

[63] G.Oriolo e M. Vendittelli, «A control-based approach to task-constrained motion planning,» *IEEE International Conference on Intelligent Robots and Systems,* pp. 297-302, 2008.

[64] P. Zacharia, E. Xidias e N. Aspragathos, «Task scheduling and motion planning for an industrial manipulator,» *Robotics and Computer-Integrated Manufacturing 29,* pp. 449-462, 2013.

[65] A. Kovacs, «Integrated task sequencing and path planning for robotic remote laser welding,» *International Journal of Production Research 54 (4),* pp. 1210-1224, 2016.

[66] S. Alatartsev, S. Stellmacher e F. Ortmeier, «Robotic task sequencing problem: a survey,» *Journal of Intelligent & Robotic Systems 80 (2),* pp. 279-298, 2015.

[67] C. D. Marino, A. Rega, F. Vitolo, P. Stanislao e A. Lanzotti, «The anthropometric basis for the designing of collabrative woskplaces».

[68] A. C. Radu e M. S. Carmen, «INDUSTRIAL APPLICATIONS OF IMAGE PROCESSING».