

MASTER ERASMUS MUNDUS
EMARO+ "EUROPEAN MASTER IN ADVANCED ROBOTICS"

2016 / 2017

Master Thesis Report

Presented by

Divya SHAH

29th August, 2017

**Increased Productivity of an Automated Tape Winding
Process: SPIDE-TP Platform**

Jury

President:	Olivier KERMORGANT	Assistant Professor (ECN,LS2N)
Evaluators:	Giorgio CANNATA	Associate Professor (UG)
	Kévin SUBRIN	Lecturer (IUT)
Supervisor(s):	Benoît COURTEMANCHE	Ingénierie Polymères & Composites (CETIM)
	Stéphane CARO	Researcher - HDR (CNRS)
	Anatol PASHKEVICH	Professor (IMT-A)

Laboratory: Laboratoire des Sciences du Numérique de Nantes LS2N

Abstract

Composite materials offer attractive properties compared to the traditional structural materials, such as high strength-to-weight ratio, flexibility in shaping and corrosion resistance. They are therefore being extensively used in various industrial sectors, especially aerospace and automotive industries. The traditional methods of fabrication by manual tape laying are labour intensive, time consuming and less efficient. To overcome these drawbacks and to keep up with the growing competitiveness, the enterprises look towards Automated Tape Winding for manufacturing composite components.

Automated Tape Winding is a fabrication process in which the workpiece liner is mounted on a rotating mandrel and heated fibre tows are consolidated over it with a compaction roller mounted, on an industrial manipulator. The robotic cell, comprising of a 6-axis robot and an actuated positioner, is kinematically redundant with respect to the considered task of tape winding. To exploit this redundancy to its full potential and to compensate for high raw material costs of composites, it is essential to optimize the tape winding process.

This master thesis work, completed within the framework of EMARO+, proposes a comprehensive methodology for optimizing the robot and the positioner trajectories with improved kinematic redundancy management. This work was realized in collaboration with the industrial partner CETIM's technological platform named, *SPIDE-TP Platform*, with the ultimate aim of increasing the overall productivity of an Automated Tape Winding process.

In this work, the industrial platform was modelled in a CAD environment. The robot and positioner motions were optimized by following a graph-based search using dynamic programming and complemented with intensive collision detection on the CAD model. The simulation results promise to reduce the processing time by one-third.

Acknowledgements

The author would like to acknowledge the industrial supervisor Mr. B. Courtemanche, the technician Mr. J. Lefort and all the concerned employees of CETIM for their guidance towards the work and also for providing with the adequate resources and the access to the automated tape winding system, called *SPIDE-TP Platform* on which the implementations were performed.

The author would also like to acknowledge the academic supervisors, Dr. S. Caro and Prof. A. Pashkevich, and the doctoral student at IMT-A, Mr. J. Gao; for proposing a novel approach to optimize the manipulator motion in a redundant robotic system, which was adapted in this work to generate and implement time-optimal solutions for the real system, and for their valuable guidance throughout the course of this thesis work.

Lastly, the author would like to thank CNRS and LS2N for providing the opportunity to carry out this research and all the members on the jury for evaluating the work.

Nomenclature

List of Abbreviations

AFP	Automated Fibre Placement
AFPT	Advanced Fibre Placement Technology (industry)
ATL	Automated Tape Laying
ATW	Automated Tape Winding
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CETIM	CEntre Technique des Industries Mécaniques
CIRC	Circular motion instruction in KRL
CNC	Computer Numerical Control
CNRS	Centre National de la Recherche Scientifique
DOF	degrees of freedom
ECN	École Centrale de Nantes
EMN	École des Mines de Nantes
EMARO+	Erasmus+ European Masters on Advanced Robotics
FIM	Fédération des Industries Mécaniques
IMT-A	Institut Mines-Télécom Atlantique
IUT	Institut Universitaire de Technologie de Nantes
KRL	KUKA Robot Language
LIN	Linear motion instruction in KRL
LS2N	Laboratoire des Sciences du Numérique de Nantes
PEEK	Polyether ether ketone (thermoplastic)
PP	Polypropylene (thermoplastic)
PTP	Point-to-point motion instruction in KRL
SPL	Spline motion instruction in KRL
SPM	Special Purpose Machines
TCP	Tool Centre Point
UG	Università degli Studi di Genoa
UN	Université de Nantes

List of Symbols

F_0	global frame
F_{RB}	robot base frame
F_{PB}	positioner base frame
F_{RF}	robot flange frame
F_{PF}	positioner flange frame
F_L	linear track frame
F_T	tool frame
F_{TL_i}	task location frame
F_W	workpiece frame
${}^A T_B$	homogeneous transformation matrix from frame A to frame B
q_r	robot joint coordinates
\dot{q}_r	robot joint velocities
\ddot{q}_i	robot joint accelerations
q_l	linear track joint coordinate
\dot{q}_l	linear track velocity
\ddot{q}_l	linear track acceleration
q_p	positioner joint coordinate
\dot{q}_p	positioner joint velocity
\ddot{q}_p	positioner joint acceleration
$rob()$	robot kinematics function
$pos()$	positioner kinematics function
$lin()$	linear track kinematics function
Rot_Z	homogeneous rotation matrix around Z-axis
P_X	homogeneous position matrix along X-axis
p_i	position vector of the origin of the i^{th} frame
φ_i	rotation vector of the i^{th} frame based on Z-Y-X Euler angles
ϕ	Euler angle about Z-axis
θ	Euler angle about Y-axis
ψ	Euler angle about X-axis
n_i	unit vector giving the direction of X-axis
s_i	unit vector giving the direction of Y-axis
a_i	unit vector giving the direction of Z-axis
μ	robot configuration index vector
$cols(.)$	collision detection binary function
t_i	i^{th} time instance
k	sampling index of the redundant variable
L_c	location cell
T	total travelling time
$dist()$	distance between subsequent nodes
$d_{k,i}$	shortest path up to current node
n	number of task locations
m	number of solution candidates for each task point

Contents

Abstract	2
Acknowledgement	3
Nomenclature	4
List of Figures	10
List of Tables	11
Introduction	13
1 Automated Tape Winding Process	15
1.1 Robotic Workcell	15
1.2 State-of-the-Art Techniques	17
2 SPIDE-TP Platform	21
2.1 About CETIM	21
2.2 SPIDE-TP Platform Specifications	21
2.3 Thesis Objectives & Approach	24
3 System & Task Modelling	27
3.1 CAD Modelling	27
3.2 Kinematic Modelling	29
3.3 Winding Task Modelling	30
3.4 Contributions	32
4 Optimal Trajectory Generation	33
4.1 Problem Formulation	33
4.2 Task Graph Generation	34
4.3 Collision Detection	36
4.4 Optimal Motion Planning	39
4.5 Contributions	41
5 Optimal Motion Implementation	43
5.1 Simulation & Analysis	43
5.2 Robot Motion Programming	48
5.2.1 Point-to-Point Motion	48
5.2.2 Continuous-Path Motion	49

5.3 Contributions	52
Conclusion	53
A Robot Parametrization & Geometric Modelling	55
A.1 Robot Parameters	55
A.2 Direct Geometric Model	56
A.3 Inverse Geometric Model	58
Bibliography	59

List of Figures

1	Simulation of a CNC machine manufacturing a large one-piece fuselage barrel by ATW [21, Marsh, 2011]	14
1.1	Robotic Fibre Placement System with 1 & 2 axis Positioner respectively, [21, Marsh, 2011].	15
1.2	Experimental robotic fibre placement system [29, Shirinzadeh <i>et. al.</i> , 2004].	16
1.3	Schematic for a typical fibre placement head [10, Debout <i>et. al.</i> , 2011].	17
1.4	Recursive Numerical Algorithm for Fibre Placement for Open Surfaces [30, Shirinzadeh <i>et. al.</i> , 2000].	18
2.1	SPIDE-TP Platform, courtesy CETIM.	22
2.2	Large Winding Axis, courtesy CETIM.	22
2.3	Winding Head Assembly, courtesy CETIM.	23
2.4	Tape winding tracks generated in Composicad [3].	23
2.5	Illustration of laser-assisted tape winding [16, Grouve <i>et. al.</i> , 2012].	24
2.6	Outline of the approach for optimizing ATW.	25
3.1	CAD models of workcell components.	28
3.2	Manual Simulation in CATIA V5.	28
3.3	Typical Robotic Fibre Placement System.	29
3.4	Workpiece liner CAD model.	31
3.5	Test track representing the task locations.	31
4.1	Graph based representation of discrete search space.	35
4.2	Different types of collisions detected.	37
4.3	Collision detection, considering only tool and workpiece, in CATIA V5.	38
4.4	Adapting the TCP frame with laser head angle, courtesy CETIM.	38
4.5	Modification of the shaft geometry, courtesy CETIM.	39
4.6	Data exchange through Microsoft Office Excel between MATLAB and CATIA V5.	41
5.1	Robot location on linear axis w.r.t travelling time.	44
5.2	Results comparison with different number of task locations.	45
5.3	Comparison of TCP displacement and speed among different number of task locations.	46
5.4	Result comparison with different objectives.	47
5.5	Comparison of TCP displacement and speed with different objectives.	48
5.6	Path realizations with different motion commands in KRL [5, KUKA Manual]. .	49
5.7	Orientation control with different motion commands in KRL [5, KUKA Manual].	50

5.8	Approximate positioning contour with LIN-CIRC blocks [5, KUKA Manual].	50
5.9	SPIDE-TP Platform during trial implementations, courtesy CETIM.	52
5.10	Comprehensive methodology followed for increasing the productivity of ATW. . .	54
A.1	Directions for the rotation of the joint axes [6, KUKA Specifications].	55
A.2	KR210 R3100 ultra working envelope, side view, representing the joint frames [6, KUKA Specifications].	56

List of Tables

5.1	Robot location on linear axis and corresponding travelling time.	43
5.2	Results obtained with different number of task locations.	44
5.3	Results obtained with different objectives.	44
5.4	Comparison of different robot motion programming strategies.	52
A.1	Robot Parameters	56

Introduction

The use of fibre reinforced composites is rapidly growing these days in various industrial sectors [21, Marsh, 2011], [14, Gay, 2014]. This is due to the fact that they offer attractive material properties over the conventional structural materials, such as high strength-to-weight ratio, high stiffness-to-weight ratio, durability, flexibility of shaping and corrosion resistance. [26, Peters, 2013].

The traditional methods to manufacture components with composite materials involved manual hand lay-up and tape laying process. The parts are formed by manually cutting the tape of composite material to the required length and then laying down the tape along the desired path, layer by layer on the core matrix material. These methods have several limitations. The material scrap rate is very high, incurring higher raw material costs. Also this labour intensive process is very time consuming, it's repeatability is very low and the overall efficiency is limited. Not only that, the process is also very hazardous for the labourers. The current requirements, especially in the aerospace and automotive industries, demand faster, higher quality and more cost effective fabrication technique. Thus the need of automation in fabrication of composite components was evident in order to exploit the advantages offered by the composite materials and to keep up with the competitiveness of the industries.

The Automated Tape Laying (ATL) technique was then developed, in which a technological tool is automatically driven to lay a continuous composite tape onto moulds in the shape of desired workpiece. The major drawback of this process is the limitations in the allowable shape of the moulds. The mechanical performance of the component drastically reduces with high variation curvatures, tape wrinkling and overlap for the mould surface. This drawback is overcome in the automated tape winding (ATW) technique [31, Sloan & Jeff, 2008].

Automated tape winding is a fabrication process in which the components are produced by a continuously applying fibre tapes over the workpiece liners mounted onto a rotating mandrel in a geometric pattern. ATW has much in common with ATL, but while the latter is best suited for relatively flat surfaces, ATW can be used with much more complex geometries provided that the curvature is always positive. However, this cannot be used to produce components with concave surfaces as the tensions in the fibres will bridge the concave region.

ATW is the most preferred technique for fabrication of any desired light weight, high strength pressure vessel. The terms ATW or Automated Fibre Placement (AFP) or Automated Filament Winding are often used synonymously.

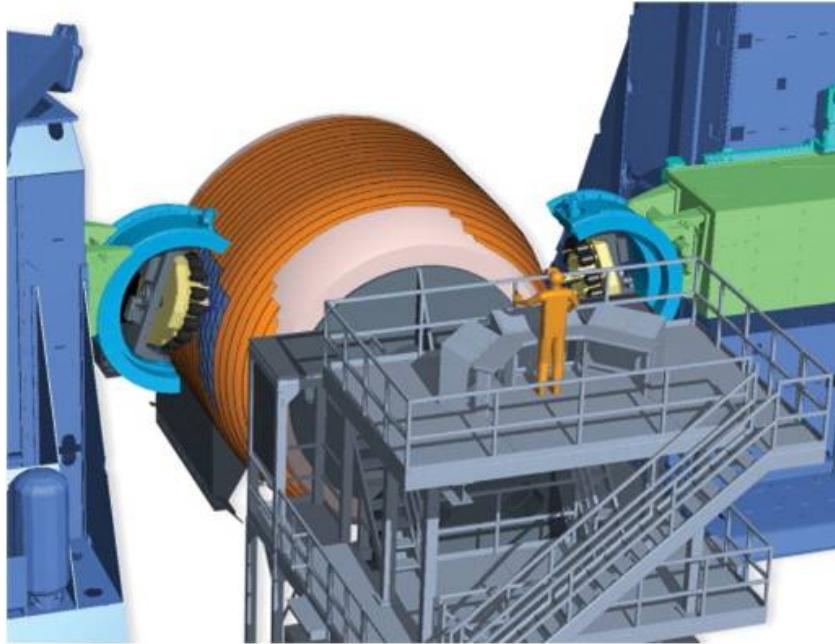


Figure 1: Simulation of a CNC machine manufacturing a large one-piece fuselage barrel by ATW [21, Marsh, 2011].

The initial automation in the manufacturing of composites was the use the CNC machines and Special Purpose Machines (SPM) as shown in the Fig. 1. One such machine and method for laying composite tapes to form a laminated structural member was patented by [15, *Grone et. al*, 1985]. More recent invention, a multiple-axis fibre placement machine for laying down fibre tows on a mandrel of desired shape was made as [7, *Benson et. al*, 2000]. These machines generally do not have limitations on the component size, however, they impose certain restrictions in terms of flexibility. They also require very large floor area and are quite expensive.

The industrial robotic systems, on the contrary, are relatively cheaper and allow more flexibility with respect to the product type. The following Chapter 1 describes the components and working of a typical robotic cell for ATW.

Automated Tape Winding Process

This chapter describes a typical robotic workcell for ATW. One of the key challenges, of kinematic redundancy of the cell, is addressed and some of the state-of-the-art techniques in literature to resolve the redundancy issues similar to ATW are discussed.

1.1 Robotic Workcell

In practice, a robotic cell for ATW consists of a 6-axis robotic manipulator and 1-axis rotating mandrel, called the positioner, on which the workpiece is rotated as shown in the left of Fig. 1.1. Sometimes, a double axis positioner can also be used, depending upon the desired geometry of the workpiece. For the case of lengthy workpieces, the robot or the positioner can be mounted on an external gantry rail, adding another axis to the system as shown in the right of Fig. 1.1.

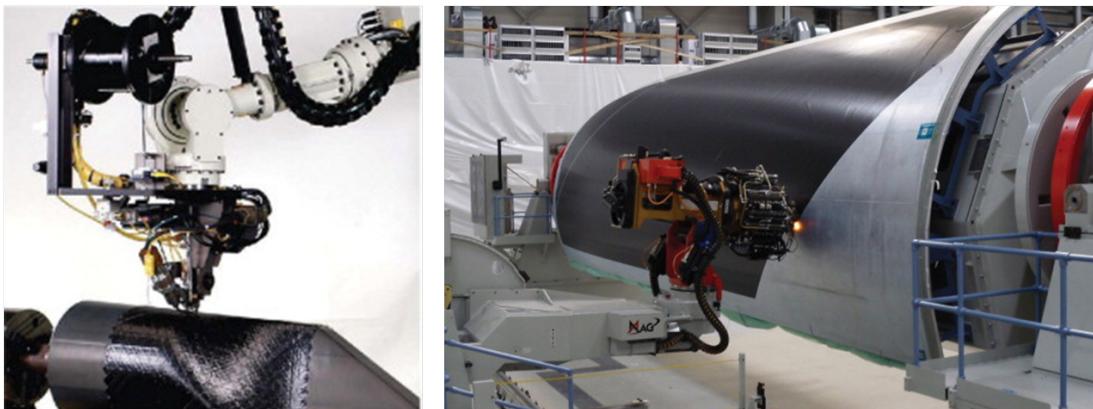


Figure 1.1: Robotic Fibre Placement System with 1 & 2 axis Positioner respectively, [21, Marsh, 2011].

An experimental robotic fibre placement system is shown in the Fig. 1.2. A creel is the disc-like system attached to the manipulator (or sometimes even directly to the tool) to store the prepreg tows. Prepreg tows are fibres pre-impregnated with a controlled amount of thermoplastics. The tows are fed through a series of guiding rollers to the placement (or winding) head attached to the robot flange. A force/torque sensor unit is installed between the end-effector

and the fibre-processing head to measure the compaction forces during the fabrication process. It can also be used as feedback for control. A hot gas torch or high-beam laser is used to apply sufficient heat to the tows to increase their tackiness such that they can be bonded together more readily. A compaction roller will then apply direct force to remove trapped air or spatial gaps (voids) between the tows and the substrate material. Further, during the fibre placement process, the robot's end-effector must be orientated such that the compaction roller pressure always remains normal to the substrate surface. The process is repeated until the desired thickness of the composite component is fabricated. The Figure 1.3 shows the schematic for a typical fibre placement head.

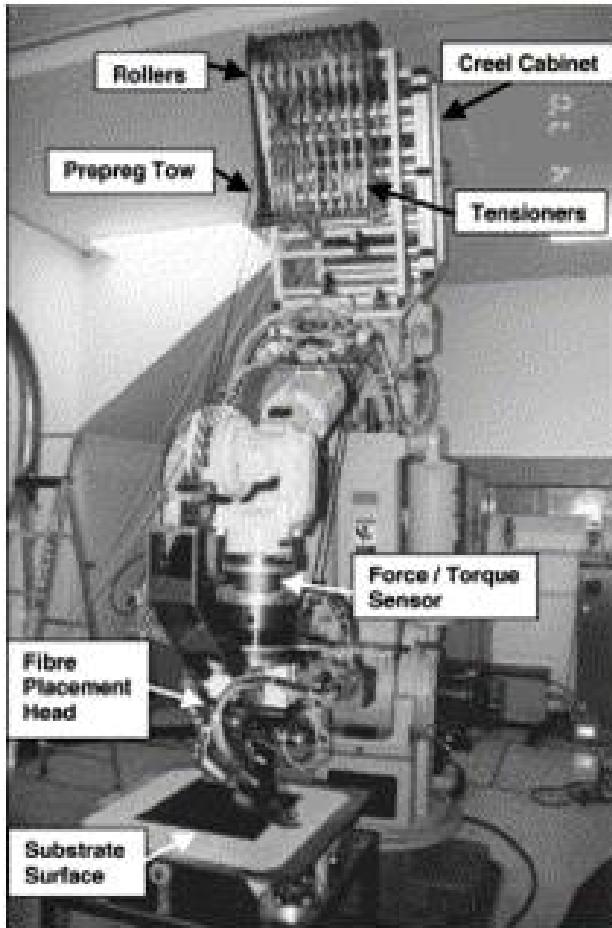


Figure 1.2: Experimental robotic fibre placement system [29, Shirinzadeh et. al., 2004].

These systems offer numerous advantageous features over the other automation systems. They include in-situ debulking and consolidation of the materials, precise control of the fibre placement head when required, cutting and restarting of fibre tows and high degree of repeatability. Furthermore, the use of robotic manipulator enhances the flexibility and allows for more complex structures to be fabricated.

An important issue in robotic fibre placement is the manipulator motion planning. The main difficulty here is the kinematic redundancy of the robotic system with respect to the manufacturing task. The robotic system has 7 to 8 DOF depending on the type of the positioner whereas the task needs only 6 DOF to be defined. To generate the desired motion, it is necessary to decompose the given task into robot and positioner motions and this cannot be done directly due to the redundancy. The following Section 1.2 discusses some of the state-of-the-art

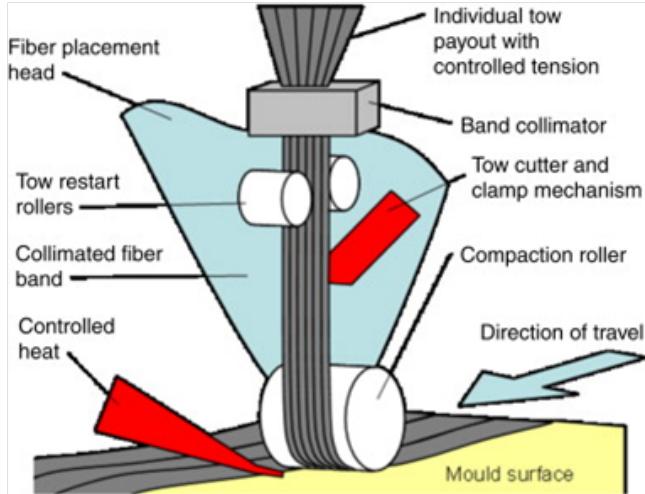


Figure 1.3: Schematic for a typical fibre placement head [10, *Debout et. al.*, 2011].

techniques in literature that deal with the redundancy issues.

1.2 State-of-the-Art Techniques

In literature there are a number of works, that propose various methods to optimize the trajectories of the manipulator, solve the kinematic redundancy and issues similar to it.

For the ATL process, approaches similar to [9, *Caro et. al.*, 2013] can be implemented; which involves the optimization of workpiece placement within the workcell based on a quality criterion of manufacturing. In [30, *Shirinzadeh et. al.*, 2000] and [29, *Shirinzadeh et. al.*, 2004] a simulation based fibre path generation technique is presented for fibre placement process of open surfaces, that is flat or contoured surfaces. Figure 1.4 shows a recursive numerical algorithm that was implemented for creation of trajectories on arbitrary B-spline curve.

The trajectory planning algorithm for industrial manipulators presented in [24, *Munasinghe et. al.*, 2003], takes into account the joint acceleration limits and end-effector velocity constraints. The optimal joint trajectory planning for manipulators performing constrained motion tasks is proposed in [28, *Shen et. al.*, 2004]. Here the corresponding cost function is discretized and the joint trajectory's intermediate positions are calculated, which are then interpolated to sustain the motion constraints.

In [10, *Debout et. al.*, 2011], a methodology to manage the degree of redundancy in order to decrease the kinematic loads of the controlled joints for applications in AFP was proposed. Whereas, in [19, *Krombholz et. al.*, 2013], an automated fibre placement process based on CAD-CAM simulations was presented.

However, these works do not directly take into account the redundancy of the robots with respect to the task. The relevant techniques to solve the kinematic redundancy are based on the pseudo inverse of kinematic Jacobian [17, *Kazerounian et. al.*, 1988]. It provides a unique solution for the differential kinematic equations corresponding to the smallest Euclidean norm

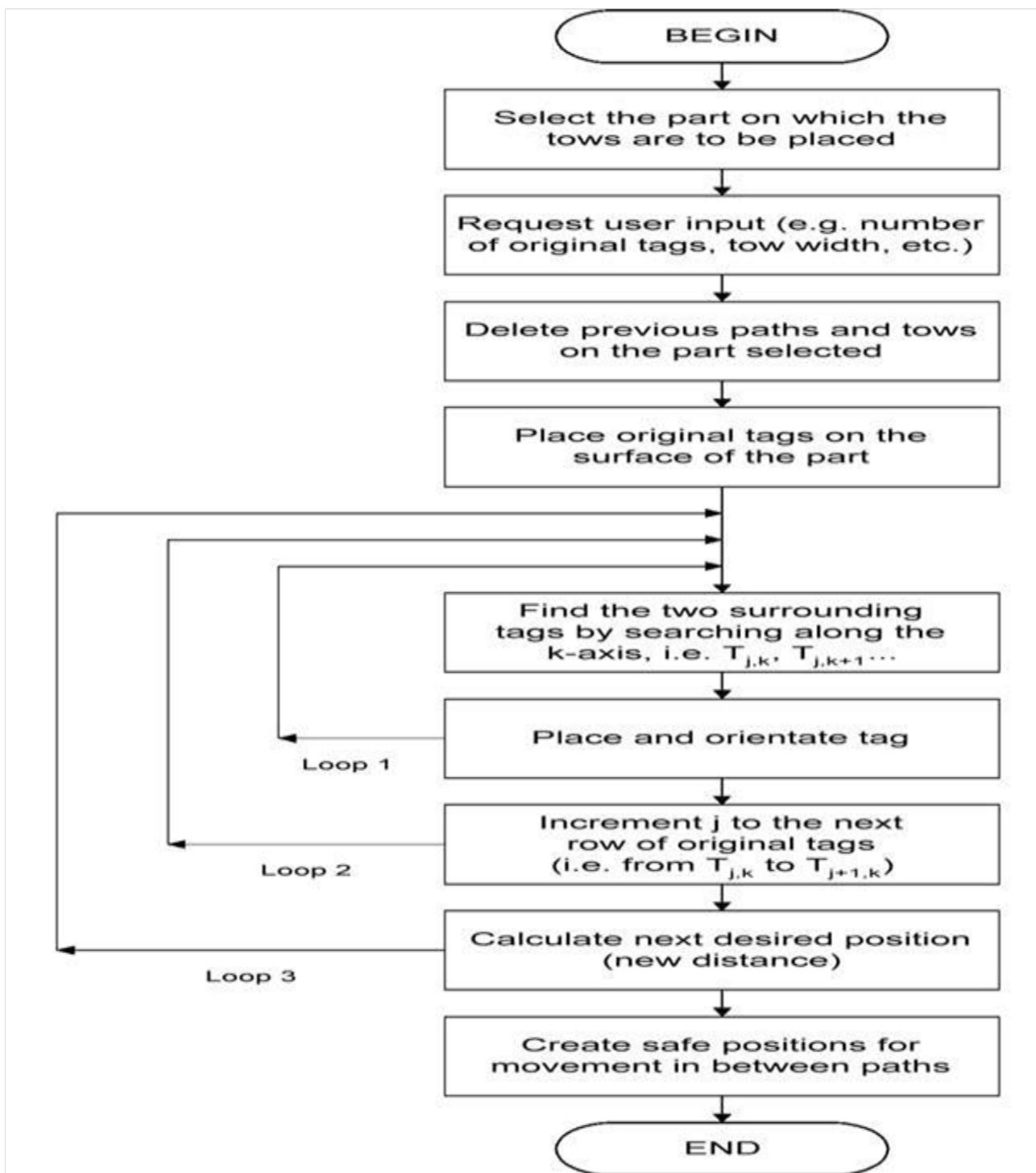


Figure 1.4: Recursive Numerical Algorithm for Fibre Placement for Open Surfaces [30, Shirinzadeh et. al., 2000].

of the displacement vector in the joint space. However, this does not generate optimal trajectories satisfying real life industrial requirements taking into account velocity and acceleration limits of the joints. [27, *Sabourin et. al.*, 2015], [32, *Subrin et. al.*, 2014] shows an optimization method to resolve the redundancy based on the projection onto the kernel of the Jacobian matrix of the gradient function generated from the required kinematic criteria, used for industrial applications.

To generate optimal motion for complex shaped workpieces taking into account various constraints, a number of alternative methods are developed using direct optimization techniques. In this approach the continuous time-optimal robot motion problem is converted to a discrete one, discretizing the robot and positioner joint space. The desired trajectory can be computed as shortest path. However, the distance assigned do not account for acceleration constraints and this may produce non-smooth trajectories with oscillations. Also the conventional search methods are computationally very heavy.

Slightly different approach, based on th dynamic programming principle was proposed in [11, Dolgui & Pashkevich, 2006] and [25, Dolgui & Pashkevich, 2009] for laser cutting operations with constant tool speed. Here, the travelling time between adjacent nodes is constant and this helps in reducing the oscillations in the joint velocities. However, in the case of fibre placement process, the assumption of constant tool speed is not favourable.

In [23, *Mlynek et. al.*, 2014] and [22, *Martinec et. al.*, 2015], authors studied the problem of fibre placement process, but the velocity of the tool was kept constant. A survey on classification and comparison between fixed angle and variable angle path planning algorithms was presented in [20, *Li et. al.*, 2015]. To the best of author's knowledge, there is minimum research addressing the time-optimal motion planning for robotic fibre placement process as proposed in [12, *Gao et. al.*, 2017], [13, *Gao et. al.*, 2017] and is presented in detail in the Chp. 4.

CHAPTER 2

SPIDE-TP Platform

This thesis work was performed in the collaboration with the industrial partner CETIM. This chapter briefly describes the laser assisted robotic tape winding platform named, *SPIDE-TP Platform*, installed at CETIM. The current methodology used for the process is described. Lastly, the need for increasing productivity is established and the objectives of this thesis work are determined.

2.1 About CETIM

CETIM was established in 1965 with the joint initiative of the French Government and the Federation of the Mechanical Engineering Industries (FIM) with the aim of pooling resources and technical skills in order to provide the mechanical industry companies with assistance to improve their productivity and guarantee quality[2]. It also establishes a relationship between the industries and various scientific research communities. With a staff of nearly 700, CETIM is now France's benchmark institute in the field of mechanical engineering. It holds the Carnot label awarded by the French Ministry of Research.

Created in 2009, **Technocampus Composites**[4] is a shared technology research platform, dedicated to the implementation of high performance composite materials. **Technocampus Composites** co-locates facilities and major stakeholders working on the development of innovative technologies in the implementation of composite materials. It also houses the Automated Tape Winding platform from CETIM, *SPIDE-TP Platform*, upon which this thesis work is based.

2.2 SPIDE-TP Platform Specifications

The SPIDE-TP Platform, as shown in the Fig. 2.1, is equipped with an industrial robot, an auxiliary linear axis and two different winding axes. The 6-axis *KUKA KR210 R3100 ultra* robot has a rated payload of *210kg* and a maximum reach of *3095mm*. The base of this robot is mounted on the *KUKA KL-2000* linear track which has a range of approximately *4500mm*.

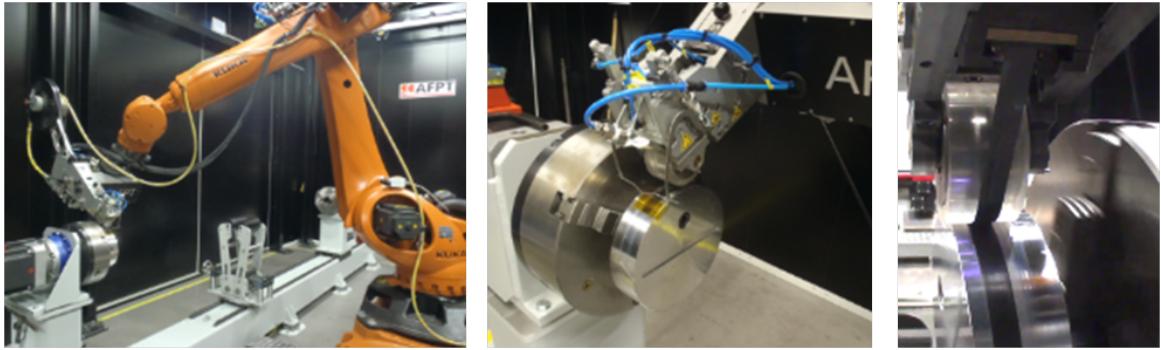


Figure 2.1: SPIDE-TP Platform, courtesy CETIM.

The two different winding axes, as shown in the Fig. 2.2, are used for rotating the mandrels for workpieces of different diameters according to the requirements. The large winding axis is used for large scale filament winding. Workpieces with diameters from $0.25m$ to $2.50m$ and lengths upto $5m$ can be fabricated on this [2].The medium winding axis can allow workpieces with diameters from $25mm$ to $500mm$ and lengths upto $3.5m$ and is used for medium and small scale filament winding applications [2].



Figure 2.2: Large Winding Axis, courtesy CETIM.

The robot flange is mounted with a technological tape winding head (tool) from AFPT [1] as shown in the Fig. 2.3. It is equipped with a material storage and a tensioning system to feed the polymer tape. A laser is used to perform the heating of the polymer above its melting temperature. The fibre optic laser power (maximum of $4kW$) is controlled via a thermal camera measuring the real time temperature of the polymer. The pressure necessary for welding is applied by a silicone compaction roller mounted on an air piston-cylinder.

The *SPIDE-TP Platform* aims to achieve winding speed of more than **$1m/s$** and the thermo-plastic throughput of more than **$1kg/min$** with reaching in-situ consolidation. It is designed to process all types of thermoplastic materials, ranging from Polypropylene (PP) to Polyether ether ketone (PEEK) and caters to various applications, the common ones being energy storage pressure vessels and cryogenic tanks.

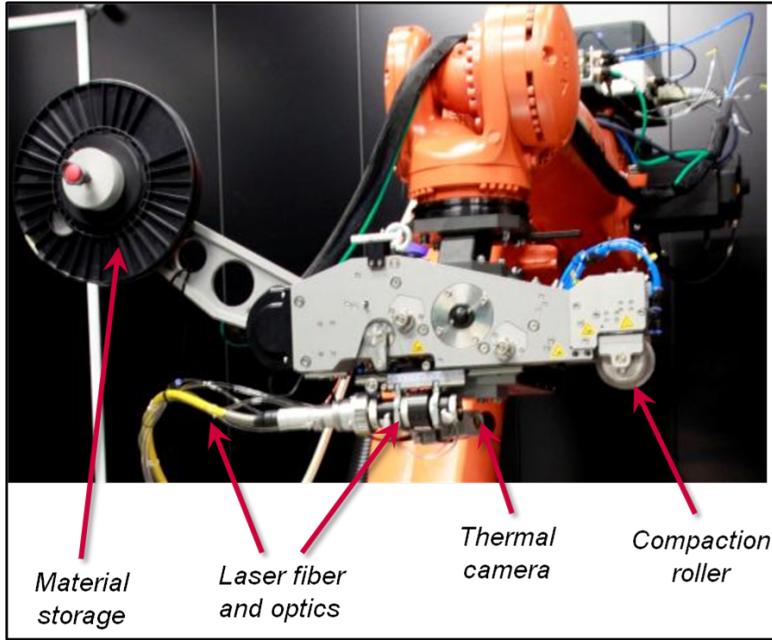


Figure 2.3: Winding Head Assembly, courtesy CETIM.

Current Methodology

Then current methodology implemented to generate motions for the robot and the winding axes is based on the software Composicad [3]. It is used to model the workpiece and generate the desired tracks for tape winding. Figure 2.4 shows a set of tracks generated on the workpiece surface in the software environment. The post processor of Composicad then, helps in building the task space trajectories and develops the corresponding robot program. The solutions obtained from this methodology are not optimal.

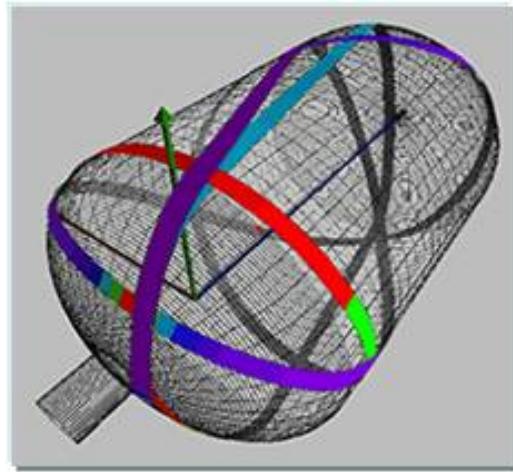


Figure 2.4: Tape winding tracks generated in Composicad [3].

External CAD files for the workpiece geometry are also supported in the software. As a part is being wound, the shape of the part is changing, due to the thickness of the composite. Composicad is capable of updating the mandrel shape online using several algorithms.

Once the tracks are generated and the robot program loaded onto the controller, the process

can then be started by loading the tape on the winding head and attaching it to the roller. Various production parameters such as pressure, tension, temperature, cooling of the roller, laser angle, etc. can be set as per the requirements through the interface of the winding head. These parameters are recorded during the fabrication process in order to analyse the performance of the process. The fabrication process is a two-step welding mechanism. First, the development of intimate contact and then interdiffusion of the polymer chains. The Figure 2.5 shows an illustration of the laser-assisted tape winding process. The finished workpiece is then removed and various tests are performed to check the weld strength and the other material properties.

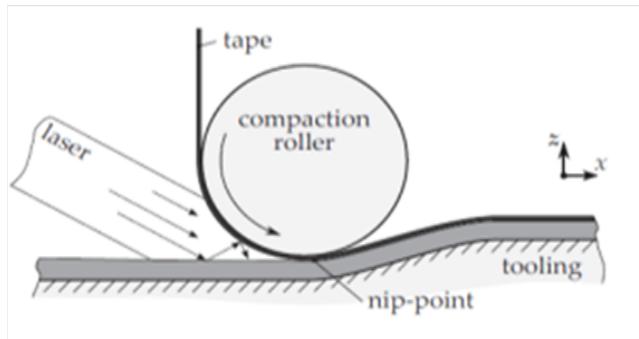


Figure 2.5: Illustration of laser-assisted tape winding [16, Grouve et. al., 2012].

Key Challenges

One of the technical challenges faced by CETIM in this process, is that the parameters are very sensitive to the tape material quality. Variations from ideal values causes surface roughness and deteriorates the strength and quality of the product. The trajectories generated from Composicad do not exploit the kinematic redundancy of the system to its full potential and also have discontinuities on domes. Another challenge is to optimize the process with a combination of speed, temperature, and/or pressure.

CETIM's approach to overcome these challenges, is to use different thermoplastic materials, to test next generation placement heads which can allow for thicker and wider tapes, and to optimize the overall process and increase the productivity with improved management of redundancy.

2.3 Thesis Objectives & Approach

The robotic system of the *SPIDE-TP Platform* is kinematically redundant. Currently, this redundancy is not taken into account as the robot motion is generated directly from the post processor of Composicad software and thus it is not optimal. Also to counter balance the high price of thermoplastic composite tapes (the raw material used for the tape winding), it is necessary to increase the productivity of the process. The general aim of this thesis work is to take into account this kinematic redundancy and the increase the average speed of production by implementing optimal trajectories for the system. For this purpose, the following objectives were proposed for this thesis work:

- Estimate the maximum production time of a given track, when using the current post-processor.
- Identify which axis is limiting the speed when running a given track.
- Estimate the gain of productivity that can be expected from an improved management of kinematic redundancy.
- Select tools to manage the kinematic redundancy and increase productivity (average speed).
- Apply the method on given test cases.

To achieve the above mentioned goals, the approach followed in this work can be briefly channeled into following three stages: (as shown in the Fig. 2.6)

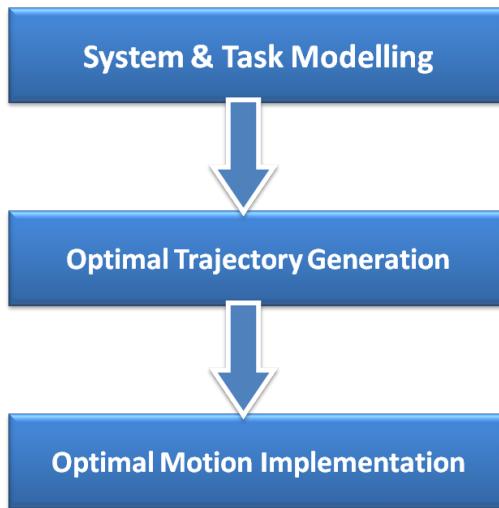


Figure 2.6: Outline of the approach for optimizing ATW.

- **System & Task Modelling:** This stage involves developing CAD model of the system as well as computing the kinematic model of the components. It also involves discretizing the considered placement track to generate the task locations. These models are developed in Chp. 3.
- **Optimal Trajectory Generation:** This stage involves generating the task graphs for all possible candidates considering the kinematic redundancy. Then running the collision detection and admissibility checks for the candidates and eventually the optimal motion planning for the robot and the positioner. The complete procedure for optimal trajectory generation is explained in Chp. 4.
- **Optimal Motion Implementation:** This final stage incorporates the simulation and analysis for the solutions obtained and the motion programming. Here different continuous path motion strategies such as Linear or Spline motions can be tested. Eventually, the results will be implemented on the SPIDE-TP platform at CETIM and the results will be analysed with respect to previous ones. Chapter 5 describes this stage in detail.

System & Task Modelling

This chapter describes the design of the CAD model for the ATW platform. The kinematic models for the robot, the positioner and the linear track are also computed and the considered placement track (task) is then discretized to generate the task model.

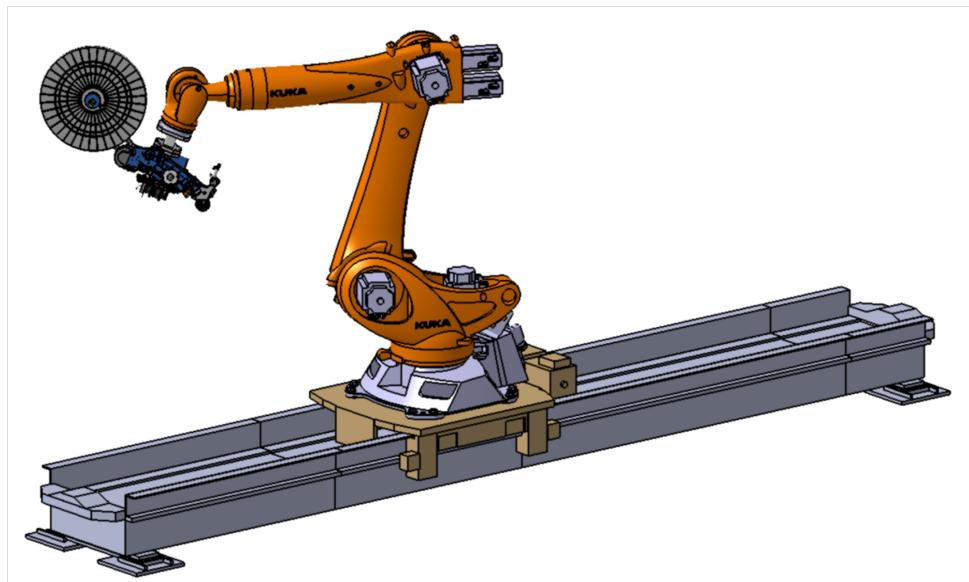
3.1 CAD Modelling

In order to facilitate the running and visualizing of the tape winding process, a CAD model for the industrial platform, previously shown in the Figs. 2.1 and 2.2, was developed on the CATIA V5 environment. Using the standard *step/iges* files available from the manufacturer's catalogues, the individual components of the workcell were recreated, as shown in the following figures.

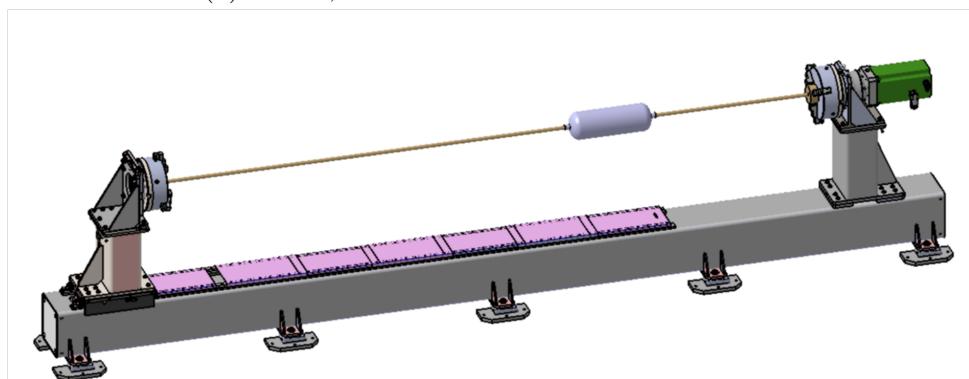
Figure 3.1a represents the CAD model of the robot with the tool on the linear track and Figure 3.1b depicts the CAD model of the winding axis and the considered workpiece liner. The components were assembled together to create the CAD model for the *SPIDE-TP Platform* as shown in the Fig. 3.2.

Using the **DMU Kinematics Workbench** on CATIA V5, all the actuated joints of the system were modelled and a mechanism was created. This mechanism can be simulated by manually jogging the commands for each axes, that is the 6 axes of the robot (A1,A2,A3,A4,A5,A6) as well as the two external axes, that is, the linear axis for the robot and the winding axis of the positioner (E1 and E2). This is represented in the following Fig. 3.2.

Another way to simulate the mechanism is 'Simulation by Laws'. In this method, the joint trajectories are input as a table from a **Microsoft Office Excel** file. The corresponding parameters of the joint commands are linked to this table by mathematical relations.



(a) Robot, linear track and tool CAD model.



(b) Positioner and workpiece CAD model.

Figure 3.1: CAD models of workcell components.

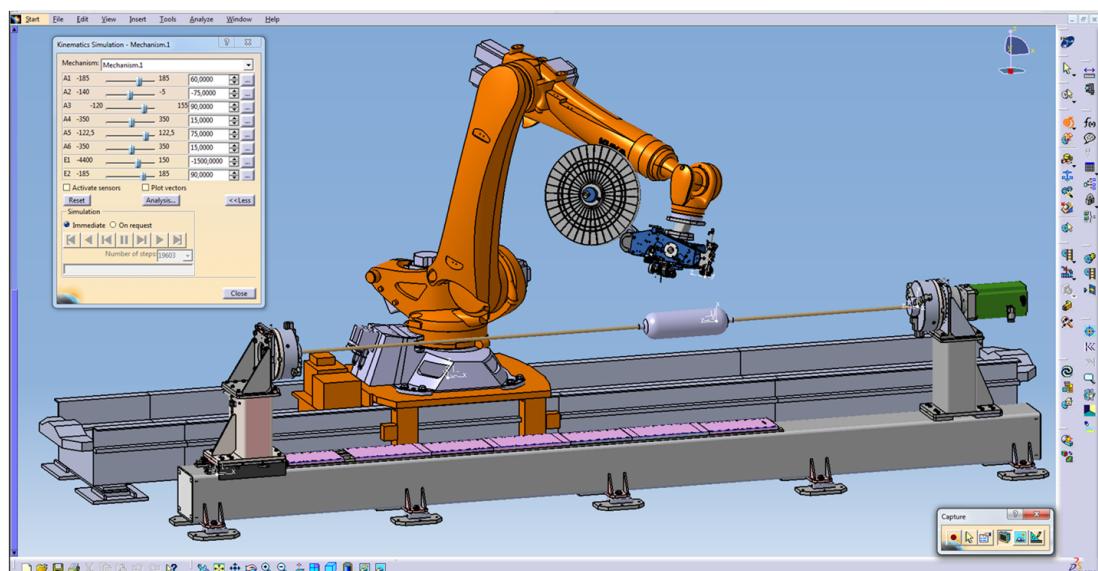


Figure 3.2: Manual Simulation in CATIA V5.

3.2 Kinematic Modelling

In order to compute the kinematic model of each of the component of the system, a standard Cartesian coordinate frame of reference is assigned to each of them, as shown in the Fig. 3.3, and they are referenced to a global frame. They are as follows:

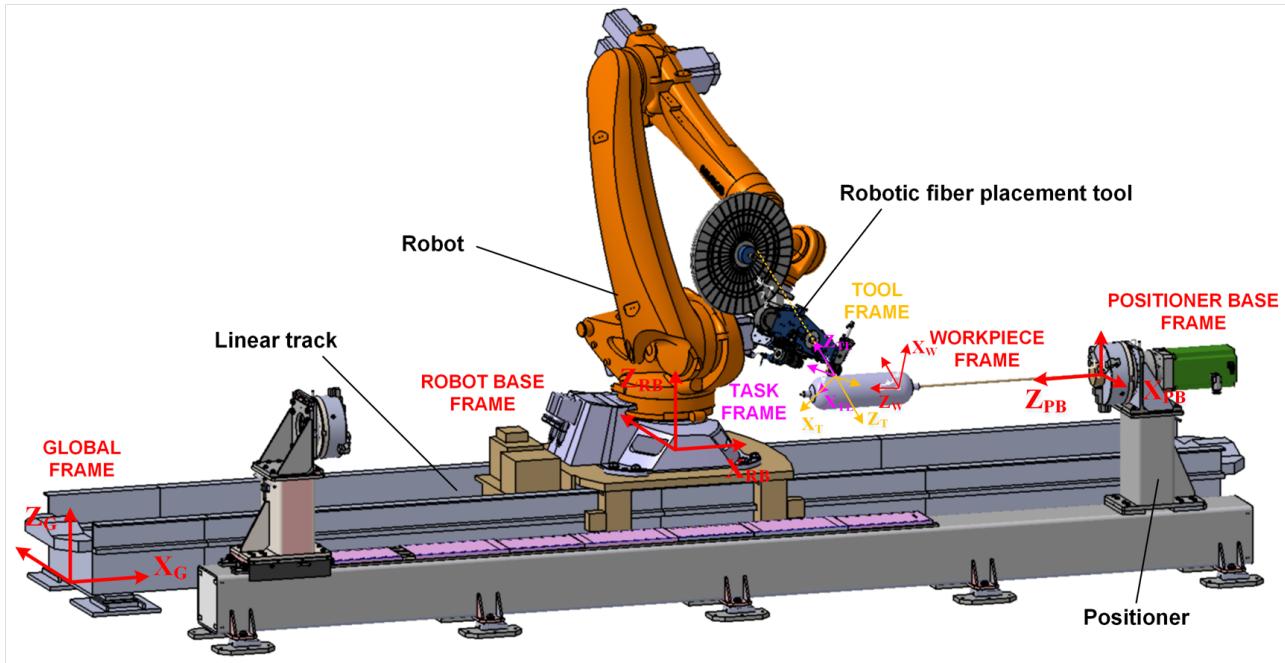


Figure 3.3: Typical Robotic Fibre Placement System.

- **Global Frame F_0** represents the global frame of the workcell with respect to which the other components are referenced. Its Z-axis is vertical and X-axis in the direction of the linear track.
- **Robot Base Frame F_{RB}** has its origin at the base of the robot and orientated similar to the global frame.
- **Positioner Base Frame F_{PB}** represents the base of the positioner. The origin is at the centre shaft with its Z-axis as the axis of rotation.
- **Workpiece Frame F_W** represents the workpiece, with its Z-axis along the length of the cylinder.
- **Tool Frame F_T** with the origin at TCP.
- **Task Frame F_{TL_i}** , the origin of which represents the discretized task location on the path with its X-axis along the placement direction and Z-axis normal to the workpiece surface pointing outside.
- **Robot Flange Frame F_{RF}** with origin at the centre of the robot flange, to which the tool can be mounted. (This is not shown in the figure).
- **Positioner Flange Frame F_{PF}** represents the flange of the positioner on which the workpiece can be mounted, with Z-axis along the mandrel. In this case, coinciding with the positioner base frame when in zero position.

- **Linear Track Frame** \mathbf{F}_L represents the linear track, its origin at the centre of the moving rail. The robot base frame and the linear track frame coincide in this case.

Robot Model

The robot model describes the mapping from the base of the robot to its flange. The following Equation (3.1) represents the robot model.

$$\mathbf{rob}(\vec{\mathbf{q}}_r) = {}^{RB} \mathbf{T}_1(q_1) \cdot {}^1 \mathbf{T}_2(q_2) \cdot {}^2 \mathbf{T}_3(q_3) \cdot {}^3 \mathbf{T}_4(q_4) \cdot {}^4 \mathbf{T}_5(q_5) \cdot {}^5 \mathbf{T}_{RF}(q_6) \quad (3.1)$$

where the matrix function $\vec{\mathbf{q}}_r$ is the vector of the robot joint coordinates and \mathbf{T} represents the homogeneous transformation matrix from left superscript frame to the right subscript frame.

There are several ways to compute the robot parameters and obtain the model. The standard convention being the use of Denavit-Hartenberg parameters as explained in [18, Khalil & Dombre, 2004]. However, different approaches can be followed in order to simplify the computations. One such method was developed in this work and is presented in detail in the Appendix A.

Winding Axis Model

Similarly, the winding axis or the positioner model, describes the relation between the frames \mathbf{F}_{PB} and \mathbf{F}_{PF} depending on the positioner joint coordinate q_p and is given by the Eqn. (3.2).

$$\mathbf{pos}(q_p) = \mathbf{Rot}_Z(q_p) \quad (3.2)$$

Linear Axis Model

The linear track allows only the linear motion along X-axis of \mathbf{F}_L and the corresponding kinematic model can be obtained as Eqn. 3.3. However, in the case of workpieces with shorter length, the robot position can be fixed on the linear axis.

$$\mathbf{lin}(q_l) = \mathbf{P}_X(q_l) \quad (3.3)$$

3.3 Winding Task Modelling

The workpiece considered within the framework of this thesis is shown in the Fig. 3.4 along with the corresponding winding path. The workpiece liner is 168 mm in diameter, 400 mm in cylindrical length and has elliptical domes on both ends of the cylinder. As mentioned in the Section 2.2 of the previous chapter, the winding track was generated from the Composicad

software. currently, with the robot and positioner motions generated from the software, the total travelling time for this track is approximately **14 sec**.

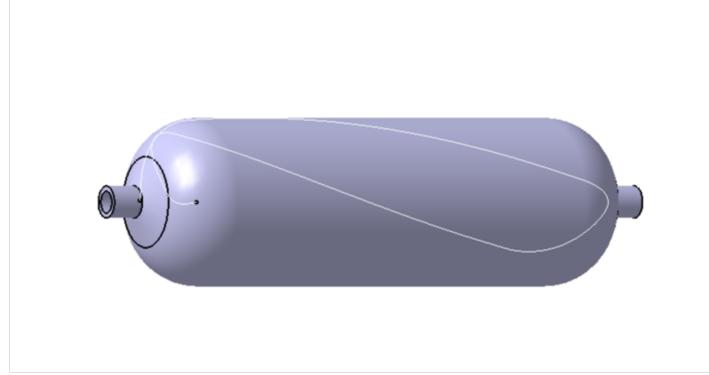


Figure 3.4: Workpiece liner CAD model.

The winding path is sampled to generate a set of n discrete task locations (position and orientations) relative to the workpiece frame \mathbf{F}_W . For each of these locations, the task location frame \mathbf{F}_{TL_i} is defined with $\mathbf{p}_i = (x_i, y_i, z_i)^T$ as the position vector of its origin and $\boldsymbol{\varphi}_i = (\alpha_i, \beta_i, \gamma_i)^T$ as the orientation vector based on the Z-Y-X Euler angle definition as explained in [18, Khalil & Dombre, 2004].

Thus the task can be represented as a sequence of 4×4 homogeneous transformation matrices, given by the Eqn. (3.4), and the robot tool (or the winding head) must visit these sequentially. Figure. 3.5 shows the winding path in blue with the green asterisk representing the start of the path. The red arrows in the figure represent the normal vectors to the surface at each of the task locations.

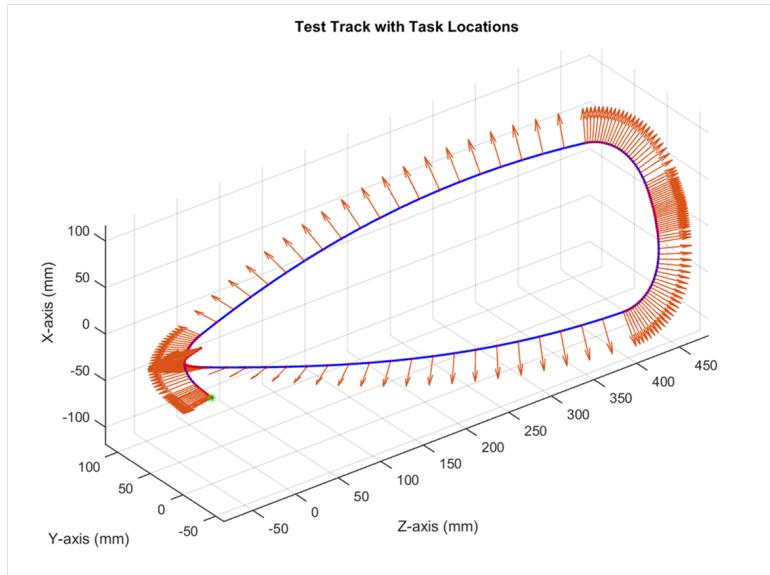


Figure 3.5: Test track representing the task locations.

$${}^W\mathbf{T}_{TL_i} = \begin{bmatrix} \vec{n}_i & \vec{s}_i & \vec{a}_i & \vec{p}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{Rot}(\vec{\varphi}_i) & \vec{p}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad |i = 1, 2, \dots, n \quad (3.4)$$

where \mathbf{n}_i , \mathbf{s}_i and \mathbf{a}_i are the unit vectors giving the directions of the X, Y and Z axes respectively.

3.4 Contributions

The contributions made within the framework of this master thesis, with respect to this Chp. 3, are as follows:

- The near accurate CAD model developed, is utilized for the collision detection to generate optimal solutions and to simulate the results.
- The kinematic models developed, are utilized in computing the solutions for the robot and the winding axis.

Optimal Trajectory Generation

This chapter describes the complete optimal trajectory generation technique using dynamic programming principle, as proposed in [13, Gao *et. al.*, 2017], which was adapted and complemented with the use of CAD model previously described for this thesis work. It mainly involves task graph generation, collision detection and path planning.

4.1 Problem Formulation

The task frames given by the Eqn. (3.4) can be expressed in the global coordinate frame \mathbf{F}_0 as functions of both the robot and the positioner kinematics.

$${}^0\mathbf{T}_{TL_i}(\vec{q}_r) = {}^0\mathbf{T}_{RB} \cdot \mathbf{rob}(\vec{q}_r) \cdot {}^{RF}\mathbf{T}_T \cdot {}^T\mathbf{T}_{TL_i} \quad (4.1)$$

$${}^0\mathbf{T}_{TL_i}(q_p) = {}^0\mathbf{T}_{PB} \cdot \mathbf{pos}(q_p) \cdot {}^{PF}\mathbf{T}_W \cdot {}^W\mathbf{T}_{TL_i} \quad (4.2)$$

Equating the above two Equations (4.1) and (4.2), we get the closed loop kinematics of the system dependent on \mathbf{q}_r and q_p .

$${}^0\mathbf{T}_{RB} \cdot \mathbf{rob}(\vec{q}_r) \cdot {}^{RF}\mathbf{T}_T \cdot {}^T\mathbf{T}_{TL_i} = {}^0\mathbf{T}_{PB} \cdot \mathbf{pos}(q_p) \cdot {}^{PF}\mathbf{T}_W \cdot {}^W\mathbf{T}_{TL_i} \quad (4.3)$$

The above Eqn. (4.3) gives six independent scalar equations for seven unknowns. Hence, this system is redundant and it does not provide a unique solution for robot and positioner coordinates for a given placement path. However, this issue can be solved by optimization.

Consider the time interval $t \in [0, T]$ and the time instances (t_1, t_2, \dots, t_n) corresponding to the instances when the robot end-effector visits the respective task locations with $t_1 = 0$ and $t_n = T$. The objective function is defined as the minimization of the total travelling time T over $\mathbf{q}_r(t)$ and $q_p(t)$ subject to the closed loop model given by the Eqn. (4.3).

$$T \rightarrow \min_{\mathbf{q}_r(t), q_p(t)} \quad (4.4)$$

Constraints

The robot/positioner joint angle, velocity and acceleration constraints are given by the Eqns. (4.5).

$$q_j^{min} \leq q_j(t_i) \leq q_j^{max} \quad (4.5a)$$

$$\dot{q}_j^{min} \leq \dot{q}_j(t_i) \leq \dot{q}_j^{max} \quad (4.5b)$$

$$\ddot{q}_j^{min} \leq \ddot{q}_j(t_i) \leq \ddot{q}_j^{max} \quad (4.5c)$$

where $j = 0, 1, \dots, 6$ is the common index for joint variables with $j = 0$ being q_p .

The collision constraints within the workcell are defined as a binary function as follows:

$$\mathbf{cols}(q_p(t), \mathbf{q}_r(t)) = 0; \quad \forall t \in [0, T] \quad (4.6)$$

These binary values are obtained from the collision detection done on the CAD model, as explained in Section 4.3.

4.2 Task Graph Generation

In order to solve this problem, a discrete optimization based approach is proposed. The positioner joint coordinate q_p is considered to be the redundant variable. This facilitates to compute the inverse kinematics for the robot for a given q_p , and get \mathbf{q}_r . The domain of the redundant variable is sampled as $q_p \in [q_p^{min}, q_p^{max}]$ with step Δq_p as given in Eqn. (4.7).

$$q_p^k = q_p^{min} + \Delta q_p \cdot k ; \quad k = 0, 1 \dots m \quad (4.7)$$

where $m = (q_p^{max} - q_p^{min}) / \Delta q_p$

Sequentially applying the equality constraint given by Eqn. (4.3) based on the kinematics of robot and positioner, a set of possible configurations for the robotic system is obtained. Let us define $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)^T$ corresponding to the manipulator posture configuration specifying shoulder, wrist and elbow respectively. Equation (4.8) defines the robot configuration to the given $q_p^{(k)}$.

$$\mathbf{q}_r^k(t_i) = \mathbf{rob}^{-1}(\mathbf{pos}(q_p^{(k)}(t_i)), \boldsymbol{\mu}); \quad k = 0, 1, \dots, m; \quad i = 1, 2, \dots, n \quad (4.8)$$

where t_i is the unknown time instant corresponding to the task location TL_i and the functions $\mathbf{pos}(.)$ and $\mathbf{rob}^{-1}(.)$ denote the direct positioner kinematics and the inverse robot kinematics respectively. Therefore, for each task location, a number of location cells can be generated as $\mathbf{L}_c^{(k,i)} = (\mathbf{q}_r^{(k)}(t_i), q_p^{(k)}(t_i))$.

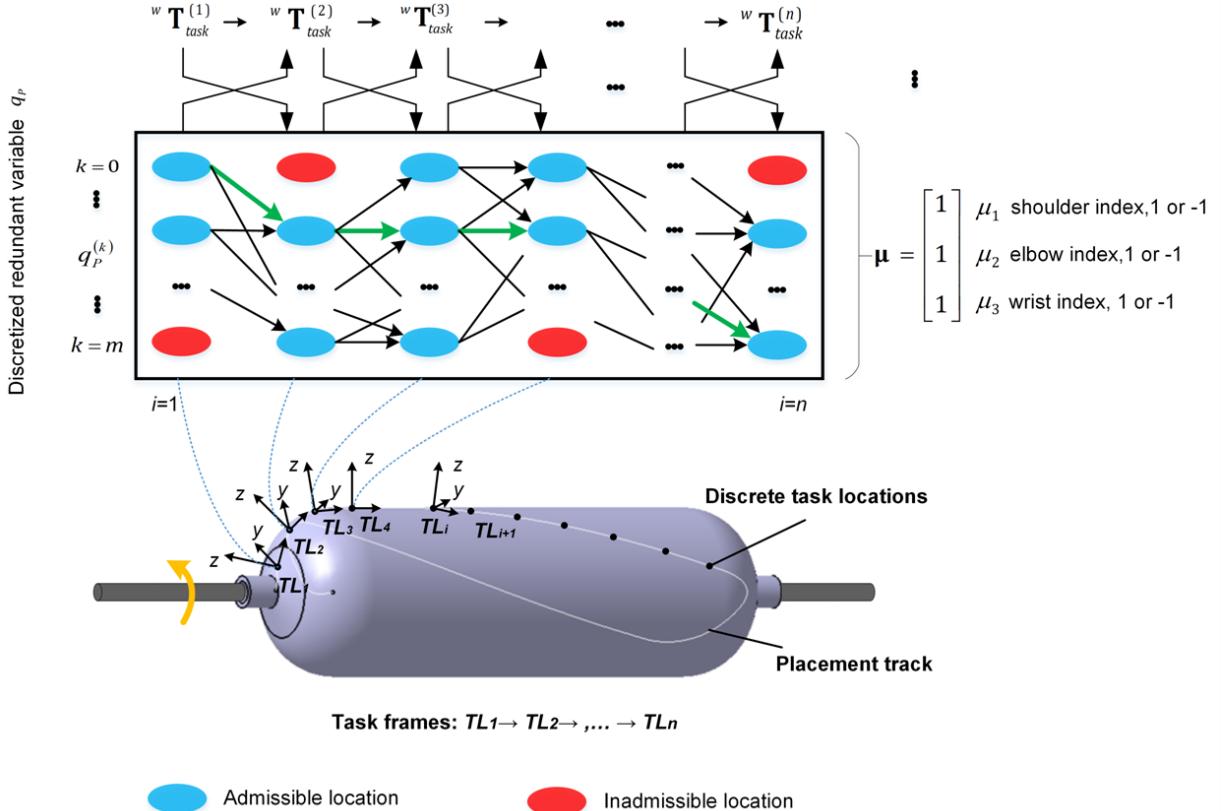


Figure 4.1: Graph based representation of discrete search space.

The sequence of task locations as described earlier in Eqn. (3.4) is converted into a directed graph-based discrete search space as shown in the Fig. 4.1. This task graph is computed with the help of a MATLAB program. The locations in red represent the inadmissible cells due to the violation of actuator joint limits Eqn. (4.5), collision constraints Eqn. (4.6) or simply to avoid the near singular postures. The collision-free candidates are obtained by running this task graph in the CAD model.

Due to time-irreversibility, the allowable connections are limited to the configuration states $\mathbf{L}_c^{(k_i,i)} \Rightarrow \mathbf{L}_c^{(k_{i+1},i+1)}$. The distance between the subsequent nodes is given as Eqn. (4.9).

$$dist(\mathbf{L}_c^{(k_i,i)}, \mathbf{L}_c^{(k_{i+1},i+1)}) = \max_{j=0,\dots,6} \left(\frac{|q_{j,i}^{(k_i)} - q_{j,i+1}^{(k_{i+1})}|}{\dot{q}_j^{max}} \right) \quad (4.9)$$

Using this discrete search space the problem becomes of finding the shortest path and the objective function to be minimized, as given in the Eqn. (4.4), can then be defined as follows:

$$T = \sum_{j=1}^{n-1} dist(\mathbf{L}_c^{(k_i,i)}, \mathbf{L}_c^{(k_{i+1},i+1)}) \quad (4.10)$$

This however, takes into account only the velocity constraints given in Eqn. (4.5) and the acceleration limits can be examined, based on the second order approximations of $\mathbf{q}_r(t)$ and $q_p(t)$, as follows:

$$\frac{2|\Delta t_i(q_{j,i+1}^{(k_{i+1})} - q_{j,i}^{(k_i)}) - \Delta t_{i+1}(q_{j,i}^{(k_i)} - q_{j,i-1}^{(k_{i-1})})|}{\Delta t_{i+1}\Delta t_i(\Delta t_{i+1} + \Delta t_i)} \leq \ddot{q}_j^{max} \quad (4.11)$$

Here, $\Delta t_{i+1} = dist(\mathbf{L}_c^{(k_i,i)}, \mathbf{L}_c^{(k_{i+1},i+1)})$ and $\Delta t_i = dist(\mathbf{L}_c^{(k_i,i)}, \mathbf{L}_c^{(k_{i-1},i-1)})$.

4.3 Collision Detection

The developed CAD model was used for running intensive collisions checks for all the solution candidates. The task graph, representing the entire search space generated for the optimization problem, are input to the mechanism and simulated by law. Interferences between the different components of the workcell have been defined; i.e., the interference between the tool and all other components except the workpiece and the interference between the robot links and other components of the workcell. The collisions (if any) are highlighted during the execution, and the results could also be saved in an Microsoft Office Excel file. This completes the **cols(.)** defined in the Eqn. (4.6).

Following Figure 4.2 shows different cases of collisions detected while running the task graphs. Figures 4.2a, 4.2e, and 4.2g represent the cases when the manipulator arm collides with the different parts fo the winding axis or the winding head. Figures 4.2b, 4.2c, 4.2d and 4.2h depict the cases in which the creel(fibre storage ring) of the winding head is in collision with the winding-axis or robot links. The Figure 4.2f represents a collision-free candidate.

Based on the different collisions detected during the trials and to avoid certain possible collisions, different modifications had to be done in the model. They are explained in the following subsection.

Practical Considerations

In order to study the reachability of the task locations more accurately, a separate simulation was created where the workpiece is kept stationary, and only the tool (without the actual robot) is made to follow the trajectory relative to the workpiece. Figure 4.3 shows this simulation where a detected collision is highlighted.

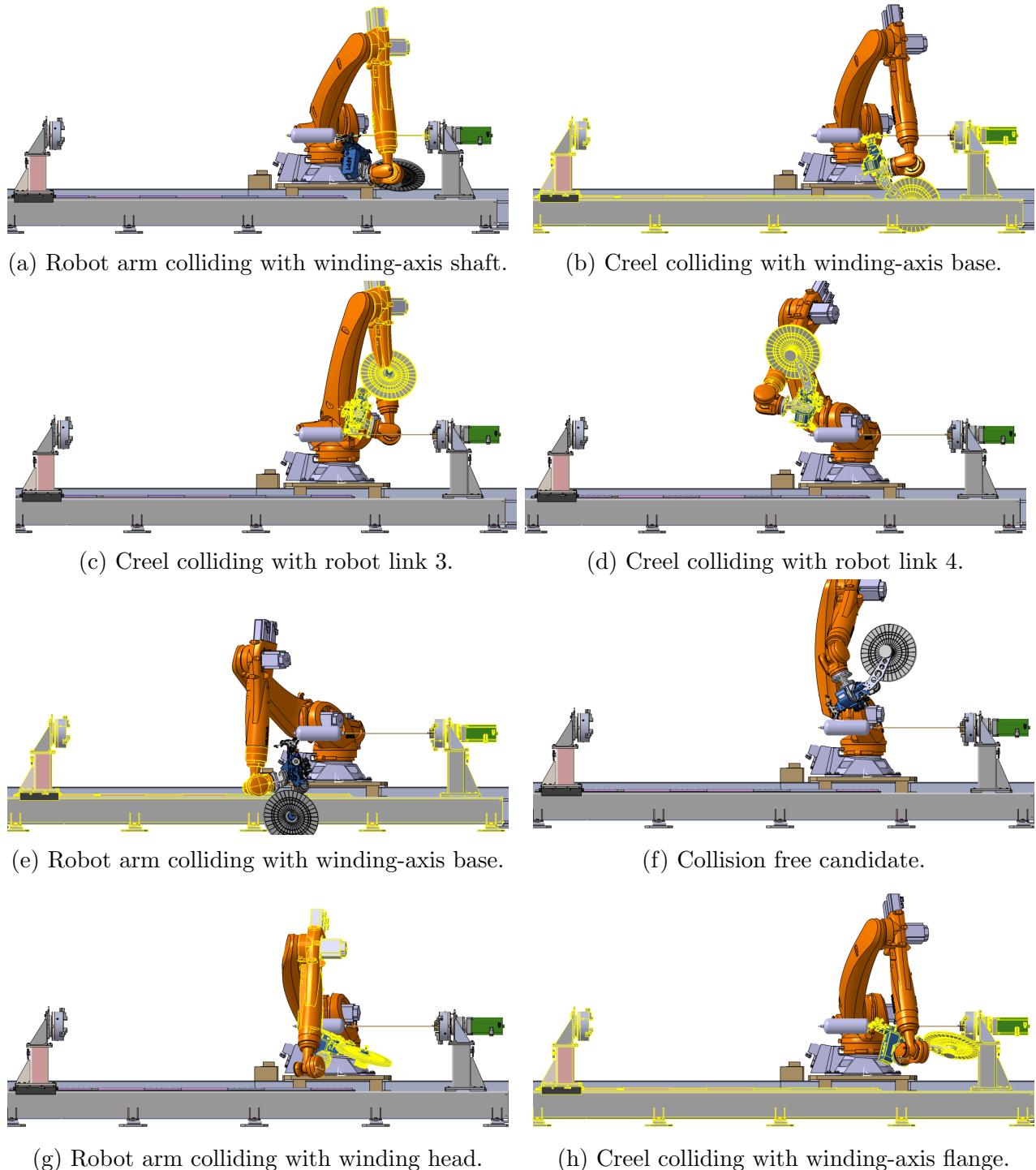


Figure 4.2: Different types of collisions detected.

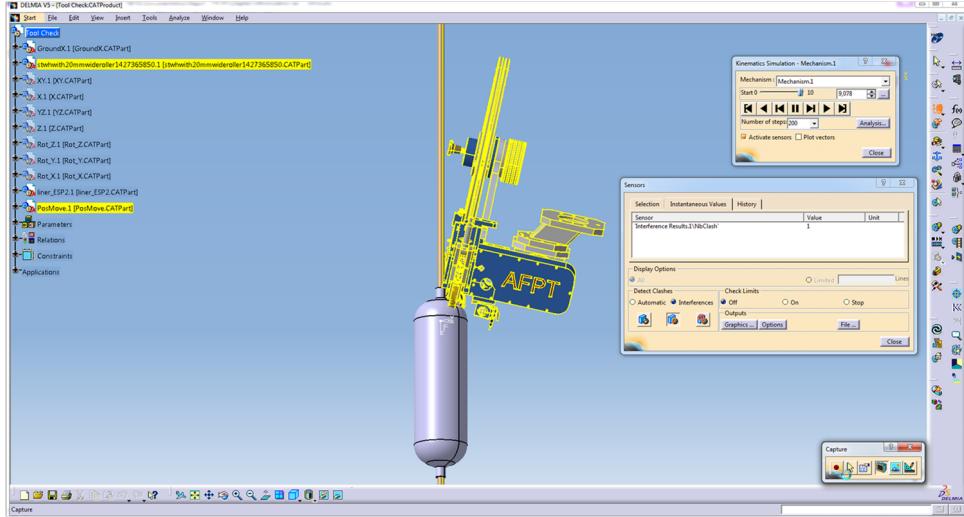


Figure 4.3: Collision detection, considering only tool and workpiece, in CATIA V5.

While running the simulations, the initial collisions detected were between the tool body, the laser in particular, with the workpiece surface. It was due to the fact that the orientation of the tool considered was incorrect. Also, the angle of incidence for the laser was not apt for the required process quality. This was then rectified, by updating the orientation of the TCP frame by taking into account the laser head angle of 26 degrees. The following Figure 4.4 shows this difference in the frames considered, with the frame in yellow being replaced by the one in blue.

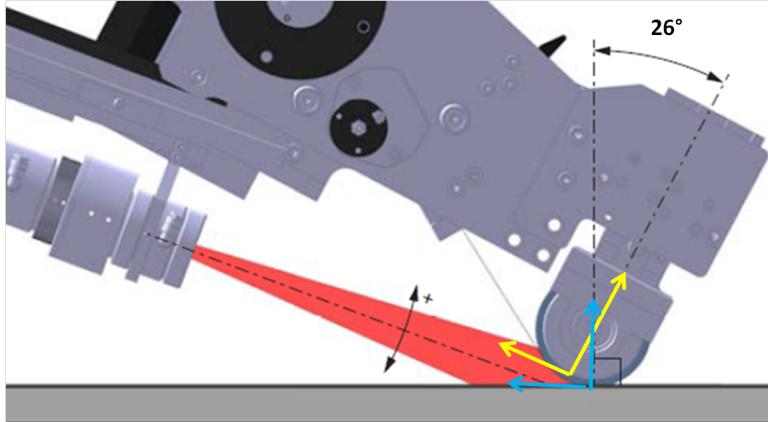


Figure 4.4: Adapting the TCP frame with laser head angle, courtesy CETIM.

Other frequent collisions detected were with the shaft on which the workpiece is mounted. When the tool is near the head of the domes on either ends it collides with the shaft. Also, when the tool reverses its travelling direction for the return phase of the track, the creel of the tool collides with the farther end of the shaft. In order to resolve these collisions the shaft geometry had to be modified. The diameter of the shaft was reduced, thus allowing to reach closer to the head of the dome of workpiece without collision as shown in the Fig. 4.5. Also the length of the shaft was reduced, close up to the end of the workpiece.

As a result of this modification in the shaft geometry; a hollow space is left between the workpiece liner and the shaft diameter. When the fibre placement process is running, the pressure from the compaction roller causes a slight movement of the workpiece liner within that

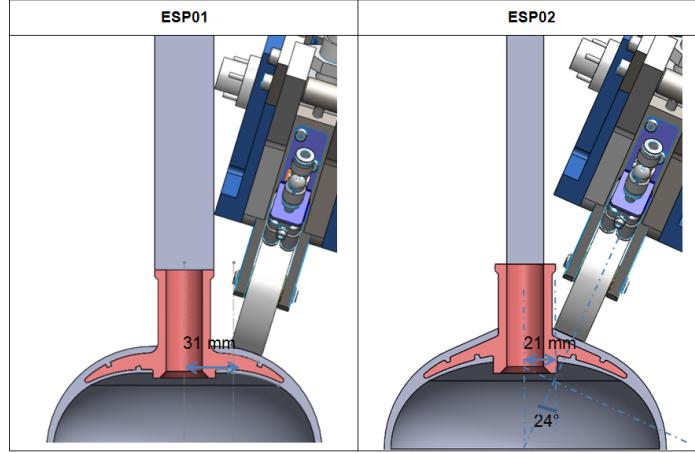


Figure 4.5: Modification of the shaft geometry, courtesy CETIM.

hollow space. Secondly, some parts of the placement tool have undergone deformation due to previous collisions in the workcell. This results in a difference between the actual placement tool and its original CAD model. To compensate for these factors, an offset was created for mounting the tool on the flange in the CAD model so as to maintain the same transformation matrix between the robot flange and the tool centre point. In practice, robust calibration methods need to be followed in order to eliminate the dimensional inaccuracy between the actual system and its 3D model.

4.4 Optimal Motion Planning

The path planning algorithm proposed, is based on the dynamic programming principle, which breaks down the full-size problem into a set of sub-problems [8, *Bertsekas et. al.*, 1995.]. To present this idea, let us consider the length of the shortest path from one initial node $\mathbf{L}_c^{k_1,1}, \forall k_1$ to the current node $\mathbf{L}_c^{k,i}$ be $d_{k,i}$. Thus, the shortest path for the successive layer is computed by finding the shortest path upto the current node and adding the distances to next layer. This is given in the Eqn. (4.12).

$$d_{k,i+1} = \min_{k'} \{ d_{k',i} + \text{dist}(\mathbf{L}_c^{(k,i+1)}, \mathbf{L}_c^{(k',i)}) \} \quad (4.12)$$

Thus the overall shortest path can be found by backtracking the indices from the final layer.

Algorithm 1: Path Planning (Dynamic Programming)

Data: Matrix of locations $\mathbf{L}(\mathbf{k}, \mathbf{i})$ of size $m \times n$
Result: Minimum path length D_{min} & Optimal path indices $k^0(i)$, $i = 1, 2, \dots, n$

```

 $\mathbf{D}(\mathbf{k}, 1) = 0 ;$ 
 $\mathbf{P}(\mathbf{k}, 1) = \text{null}, \forall k = 1, 2, \dots, m ;$ 
for  $i = 2$  to  $n$  do
    for  $k = 1$  to  $m$  do
        for  $j = 1$  to  $m$  do
            if  $\text{cols}(\mathbf{L}(j, i - 1)) = 0 \ \& \ \text{cols}(\mathbf{L}(k, i)) = 0$  or
                 $(\text{accl}(\mathbf{L}(j, i - 1), \mathbf{L}(k, i)) = 0)$  then
                     $| \quad r(j) = D(j, i - 1) + \text{dist}(\mathbf{L}(k, i), \mathbf{L}(j, i - 1)) ;$ 
                else
                     $| \quad r(j) = \text{inf} ;$ 
                end
            end
             $D(j, i) = \min(r) ;$ 
             $\mathbf{P}(j, i) = \arg \min(r) ;$ 
        end
         $D_{min} = \min(\mathbf{D}(\mathbf{k}, n)) ;$ 
         $k^0(n) = \arg \min(r) ;$ 
    end
    for  $i = n$  to  $2$  do
         $| \quad k^0(i - 1) = \mathbf{P}(k^0(i), i) ;$ 
    end

```

The following algorithm 1 summarizes the path planning algorithm as proposed in [12, *Gao et. al.*, 2017.]. Here $\mathbf{D}(\mathbf{k}, \mathbf{i})$ and $\mathbf{P}(\mathbf{k}, \mathbf{i})$ are the distance and pointer matrices.

1. Check for the collision $\text{cols}()$ and the acceleration $\text{accl}()$ constraints.
2. If they are satisfied, compute the all the paths for a robot joint upto the current layer and if not set it to infinity.
3. Repeat this for all robot joints.
4. Set the distance to the minimum of all the paths computed from the previous steps and store the corresponding pointers.
5. Repeat for all the configurations of the redundant variable.
6. Store the overall minimum path and the corresponding indices.
7. Loop again for all the task locations.
8. Retrieve the path by backtracking from the indices.

4.5 Contributions

The contributions made within the framework of this master thesis, with respect to this Chp. 4, are as follows:

- The use of CAD model on CATIA V5 complements the previously developed optimization algorithm with MATLAB.
- The interface between the two software and exchange of data at different stages of optimal motion generation is explained in the following Fig. 4.6. The exchange is through Microsoft Office Excel files.
- The collision detection performed on the CAD model is intensive and robust, as it takes into consideration the exact geometry of the workcell components and is capable of detecting all possible collisions, which may be missed by mathematical approximations.
- A time-optimal solution, using dynamic programming for the considered winding path was obtained.

Figure 4.6 shows the exchange of data between the system models on MATLAB and CATIA at different stages of the optimal motion generation stage.

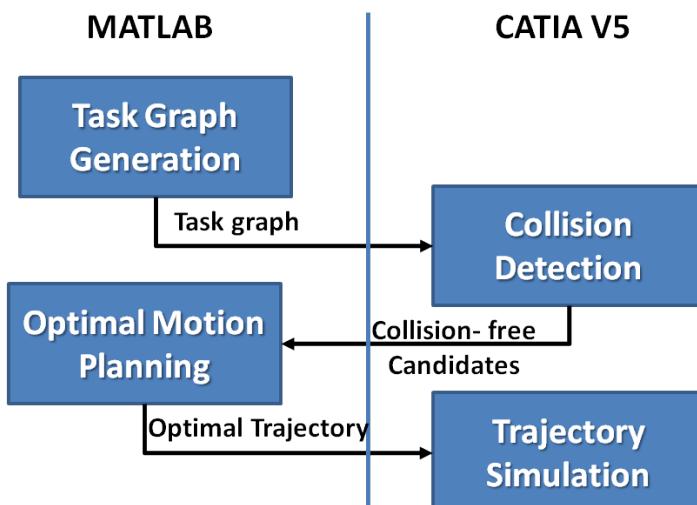


Figure 4.6: Data exchange through Microsoft Office Excel between MATLAB and CATIA V5.

Optimal Motion Implementation

In this chapter, various results obtained in the simulations are presented and analysed. Also, different robot motion programming strategies are discussed. Lastly, the results obtained from the implementations on the *SPIDE-TP Platform* are presented.

5.1 Simulation & Analysis

For the workpiece in consideration, which is shorter in length ($400mm$) compared to robot's horizontal reach; the robot location on the linear axis can be fixed to one coordinate during the task. To do this, the total travelling time for the task was computed by generating trajectories considering different positions on the linear axis. The results are presented in the following Table 5.1 and corresponding plot is shown in the Fig. 5.1. It was observed that the best travelling time for the task was obtained by fixing the robot location to **-3000mm** on the linear axis. However, in this case, there is no significant difference; and the robot location can be fixed by comparing the trajectory smoothness at different locations. For further simulation results the linear axis coordinate will be fixed to thus obtained value.

Linear Axis Position (mm)	Travelling Time (sec)
-2000	4.51
-2200	4.40
-2400	4.31
-2600	4.26
-2800	4.25
-3000	4.24
-3200	4.29
-3400	4.32
-3600	4.34
-3800	4.37
-4000	4.45

Table 5.1: Robot location on linear axis and corresponding travelling time.

The initial requirement was to achieve the maximum reduction in travelling time, irrespective of the variations in the actuator velocities throughout the path. To study this scenario,

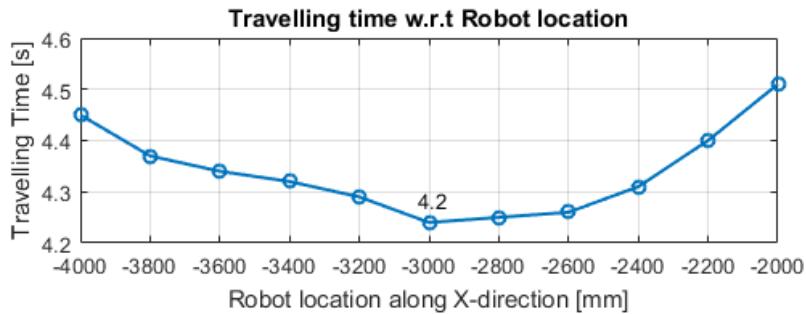


Figure 5.1: Robot location on linear axis w.r.t travelling time.

two cases of simulation results were obtained as shown in the following Table 5.2 with different discretizations of the winding path, as shown previously in Fig. 3.5. Here, the time-optimal trajectories had a travelling time of **3sec**, which is a reduction by **75%**.

No. of Task Locations	Travelling Time
127	3.0 sec
200	3.2 sec

Table 5.2: Results obtained with different number of task locations.

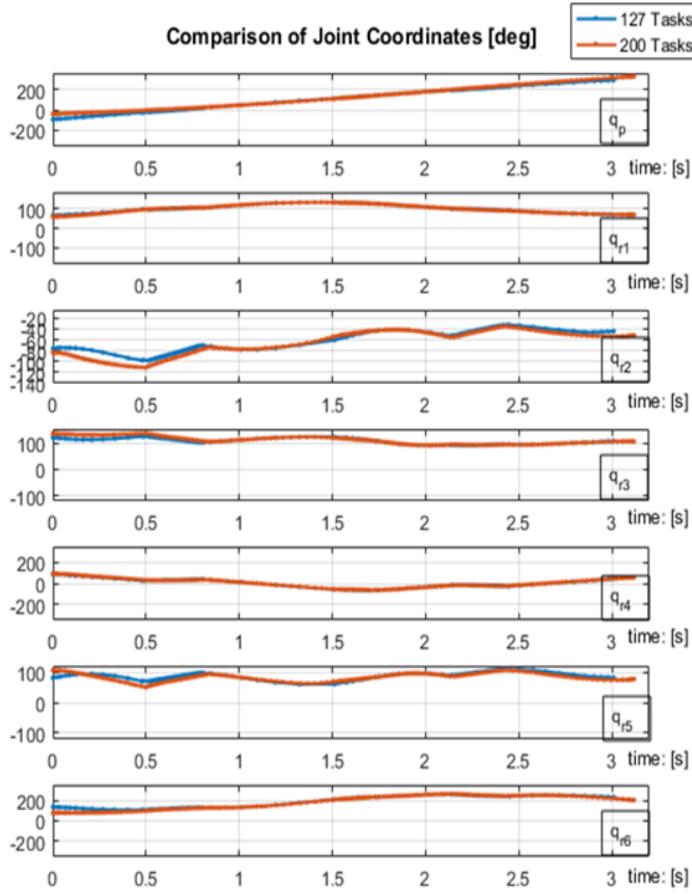
The Figures 5.2a and 5.2b represent the comparison of joint coordinates and velocities, respectively, with different number of task locations. Whereas, the TCP displacement and speed along the winding path comparisons for the same scenario are shown in the Figure 5.3.

The joint coordinates are more or less similar in both these cases, however it is observed from the velocities that the case with 200 task locations is smoother compared to that with 127, especially when there is a major change in the curvature of the surface, that is, near the dome parts of the workpiece. It could also be observed from the TCP speed plots, that the velocity variations of the TCP along the winding path are considerably high. This could not be feasible for practical implementations, as the industrial system imposes harder constraints on the velocity variations.

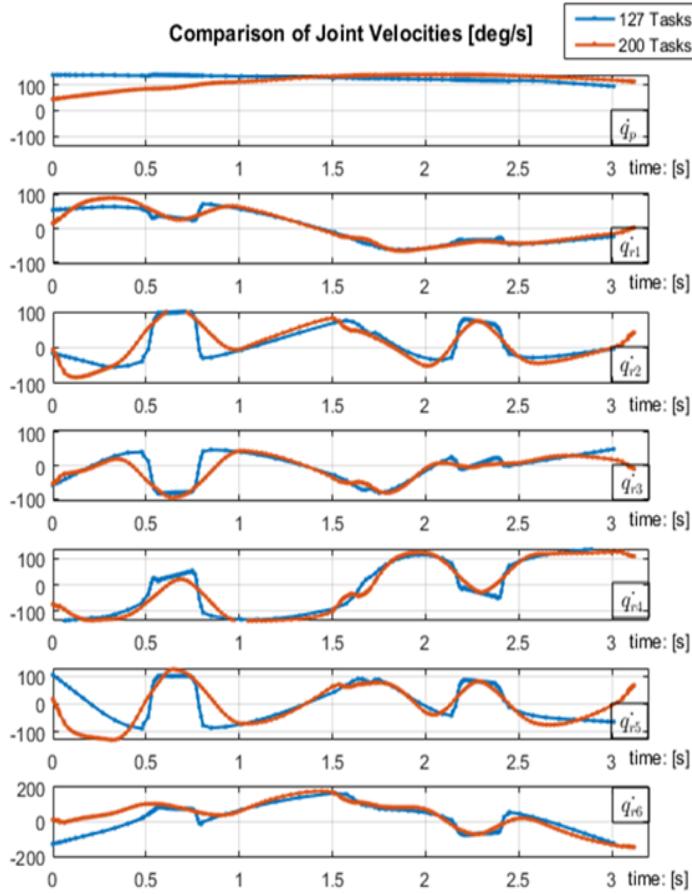
The industrial system imposes harder constraints on the variations in the velocity as this would affect the finish quality of the product. Due to this, it may not be possible to practically implement the above obtained results. Thus, along with time-optimal case, with maximum possible reduction time; an alternate objective was of constant speed was also tested in a different scenario. By keeping the speed of the TCP along the winding path constant, we restrict the variations in the actuator velocities within allowable. For this scenario, with winding path discretization into **127** task locations and the positioner coordinate by **2 deg**, the total travelling times obtained are as shown in the following Table 5.3.

Objective	Travelling Time
Constant Speed	8.4 sec
Time-optimal	4.7 sec

Table 5.3: Results obtained with different objectives.



(a) Comparison of joint coordinates.



(b) Comparison of joint velocities.

Figure 5.2: Results comparison with different number of task locations.

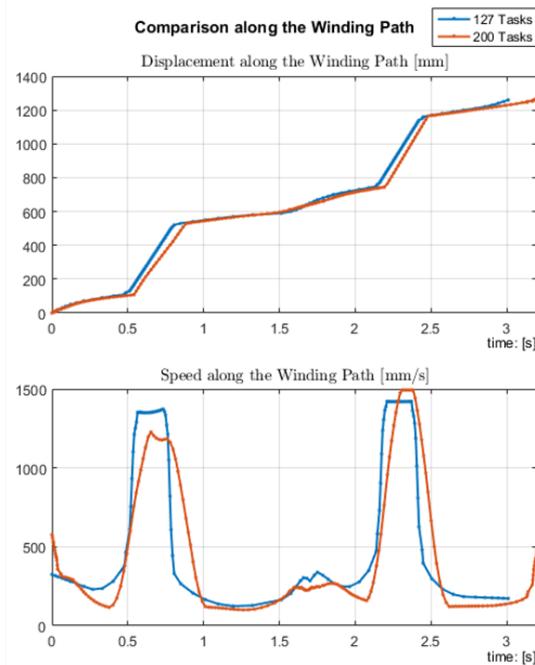


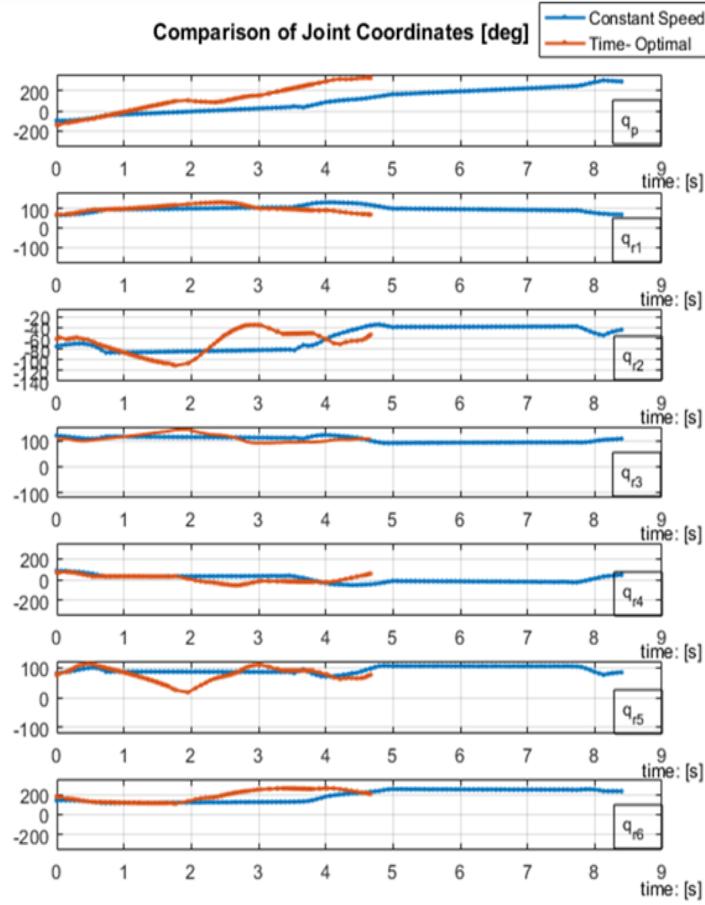
Figure 5.3: Comparison of TCP displacement and speed among different number of task locations.

Figures 5.4a and 5.4b compares the joint coordinates and velocities respectively, obtained with both objectives against the travelling time, whereas the Figure 5.5 compares the TCP displacement and speed. Here the TCP speed is kept constant to **150mm/s**. The constant speed case, reduces the processing time from **14 sec** to **8.4 sec**, thus showing a reduction by **40%**.

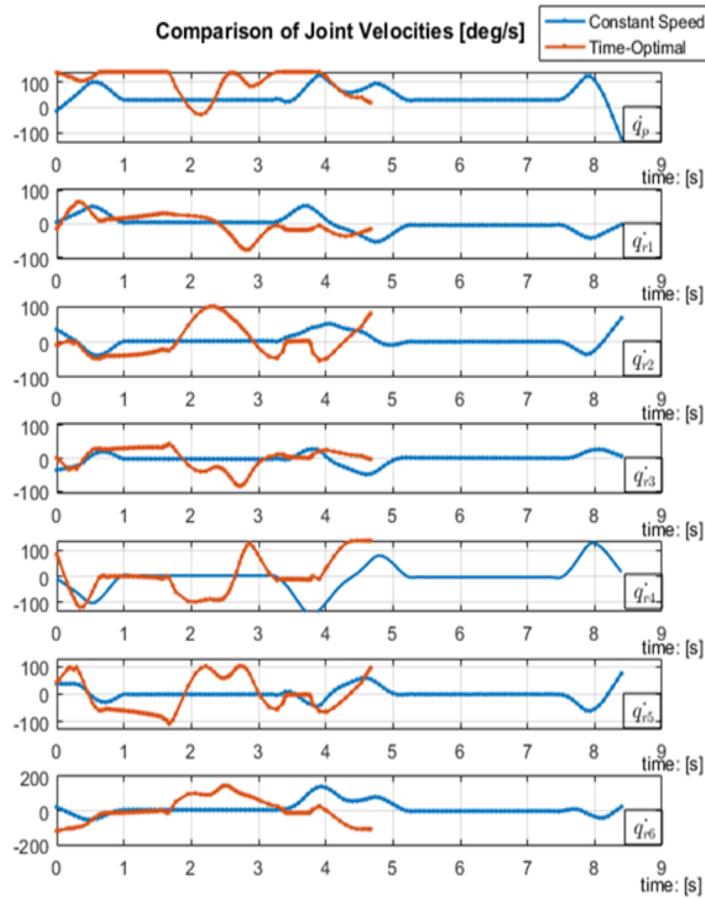
Based on the actuator velocity plots seen above, it could be said that the limiting axis for this process is the positioner axis, as it runs close to the limits for majority of the track. However, this is not conclusive enough, as there are different axes running close to the limits at different time of the track, and it is difficult to define one-particular limiting axis for the case. This depends on two factors, the distance to travel by each actuator between two task locations as well as their actuation limits.

In practice, the robot joints have additional limits, which are essential to avoid the interlocking of cables or over-bending of optical fibres resulting in abortion of the process. Moreover, high variations in the placement speed are not ideal for high quality finished product. These factors need to be considered and thus a sub-optimal solution has to be implemented.

Once the solution is ready, it has to be implemented on the industrial platform and to do so, the robot program needs to be developed. The different robot motion programming strategies are discussed and compared in the following Section. 5.2.



(a) Comparison of joint coordinates.



(b) Comparison of joint velocities.

Figure 5.4: Result comparison with different objectives.

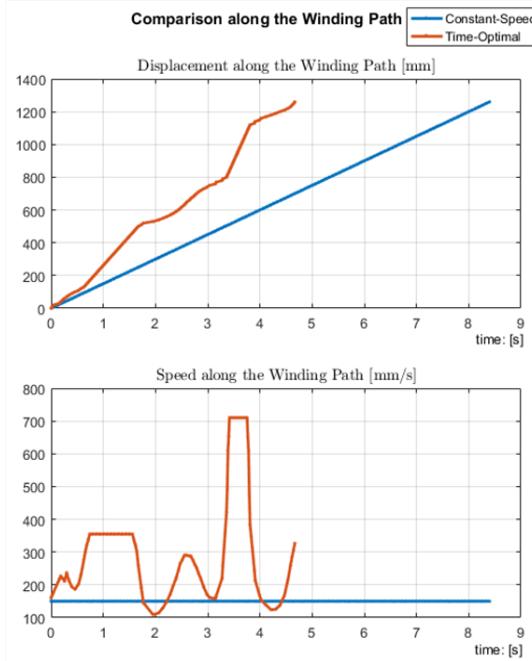


Figure 5.5: Comparison of TCP displacement and speed with different objectives.

5.2 Robot Motion Programming

Different robot controllers from different manufacturers provide various instructions to implement desired robot motions. They can be broadly classified into two types, point-to-point motions and continuous-path motions. The desired robot motions can be programmed using these individually or in several combinations. In the following sections, different motion instructions for KRL [5, KUKA Manual] are discussed.

5.2.1 Point-to-Point Motion

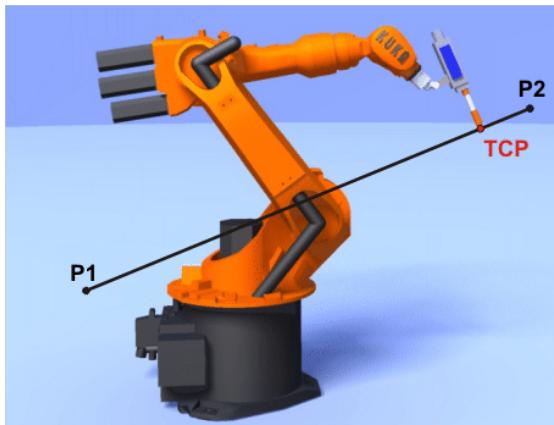
PTP is the quickest way to move the TCP from current position to the programmed one. The allowable velocity and acceleration limits for each joint have to be predefined. The controller generates the motion between the current position and the programmed position by interpolating in the joint space of the robot. The motion path in PTP is dependent on the robot's kinematic system and therefore, its geometry cannot be accurately predicted. It is mainly used for the applications involving pick up and place, palletizing, etc.

All the axes can be time-synchronized, that is they start and stop moving at the same time. This implies that only the axis with the longest trajectory, the so called leading axis, is actually moved with the programmed acceleration and velocity limits. All other axes move only with the velocity and acceleration rates necessary for them to reach the end point of the motion at the same moment, irrespective of the values programmed. The axes can also be programmed to be phase-synchronous, that is, all axes enter the acceleration, constant motion and deceleration phases simultaneously. Synchronous motion helps in eliminating the mechanical stresses on the robot since the motor and gear torques are reduced for all the axes except the leading one and it also generates smooth time optimal motion paths.

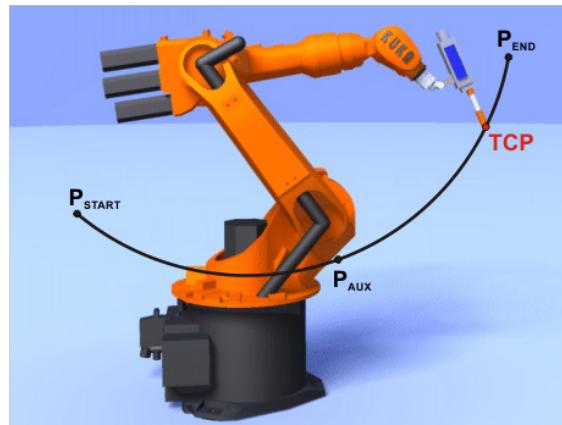
The optimal solutions obtained for the tape winding problem in consideration is in the form of discrete points and PTP motion could be implemented. However, the original problem being a continuous path, it is not advisable to use PTP. This may cause severe discontinuities in the solution as the interpolation between two points is done by the robot controller in the Joint Space.

5.2.2 Continuous-Path Motion

With continuous-path movements, the end effector describes a precise, geometrically defined path in space (straight line or circular arc), as shown in the Fig. 5.6. The controller generates the motion path by interpolating in the Cartesian space (task space). The TCP is moved precisely at the defined velocity and acceleration. The velocity and acceleration limits in this case refer to that of the TCP. However, the joint velocity and acceleration are also monitored for its limits, crossing of which results in braking of the motion. It is mainly used for the application involving continuous path movements like welding, laser cutting, tape winding, etc.



(a) Straight line motion with LIN.



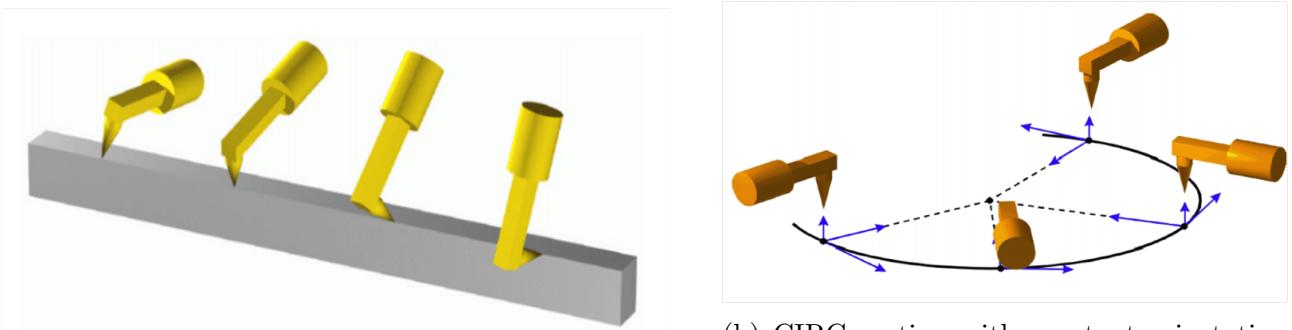
(b) Circular motion with CIRC.

Figure 5.6: Path realizations with different motion commands in KRL [5, KUKA Manual].

For LIN motion commands, the controller calculates a straight line from the current position to the programmed position defined in the Cartesian coordinates. To define a circular arc in space, at least three non-collinear points are needed. In the same way, for CIRC commands, an end position and an auxiliary position is defined. The controller generates a circular arc from the current position to the end position passing through the auxiliary position. In place of auxiliary position, a circular angle can also be defined to compute the arc.

The orientation of the tool can be controlled as desired. It can be kept constant with respect to the path, or with respect to the world frame or even varied and interpolated between the start and end positions. A LIN instruction with varied orientation is shown in the Fig. 5.7a whereas the Figure 5.7b shows a CIRC instruction with the orientation of the tool constant relative to the path.

Apart from these standard instructions, KRL also allows motions with approximate positioning. In order to increase the velocity, points for which an exact positioning is not necessary



(a) LIN motion with change of orientation.

(b) CIRC motion with constant orientation w.r.t path.

Figure 5.7: Orientation control with different motion commands in KRL [5, KUKA Manual].

can be approximated. Approximate positioning starts when the last (also the leading) axis falls below a specified angle to the approximate positioning point. This can be done within and between different motion instructions. This also helps in smooth transition between the motion commands. Figure. 5.8 shows the contour with approximation between LIN and CIRC commands.

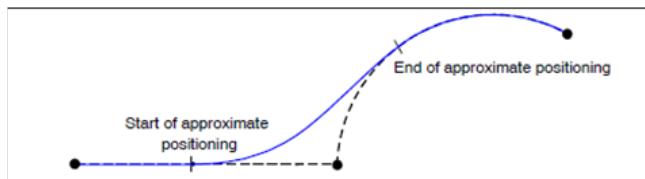


Figure 5.8: Approximate positioning contour with LIN-CIRC blocks [5, KUKA Manual].

The following algorithm 2 shows a sample syntax code for the LIN and CIRC motion commands used in KUKA Robot Language, in which the robot performs a linear motion from point $P1$ to $P2$ and a circular motion from $PSTART$ to $PEND$ through the auxiliary point $PAUX$.

In the more recent versions of the KUKA controller, spline motion commands (SPL) can be used to generate a spline motion for the defined positions. This is more suitable for particularly complexed, curved paths, such as in the case of tape winding. Here the robot path is defined by means of the points actually located on the path, this is not possible with the approximate positioning using the LIN or CIRC commands. SPL commands also allow the programmed velocity to be maintained.

Several different SPL commands can be grouped together to form an overall spline motion in a so-called SPLINE block. A spline block is planned and executed by the robot controller as a single motion block. The individual sub-blocks can also be programmed as SLIN and SCIRC commands which allow for partial linear or circular motion within the spline. Orientation control for SPL is similar to previous commands. The following Algorithm 3 shows a sample code to program spline motion block with the specified velocity control. This allows the user to adjust the travelling time. Thus using the SPLINE motion commands for the obtained solutions is most preferable.

Table 5.4 shows a brief comparison between the different strategies for robot motion programming.

Algorithm 2: LIN and CIRC commands

```
; ----- Initialization Section -----
DECL AXIS HOME ;
HOME = {AXIS: A1 0, A2 -120, A3 135, A4 0, A5 75, A6 90}
PTP HOME

; ----- Defining Points -----
P1 = {X 1030, Y 350, Z 1300, A 160, B 45, C 130}
P2 = {X 700, Y -300, Z 1000, A 23, B 230, C -90}
PSTART = {X 980, Y -238, Z 718, A 133, B 66, C 146}
PAUX = {X 925, Y -285, Z 718}
PEND = {X 867, Y -192, Z 718, A 155, B 75, C 160}
; ----- Main Section -----
$ORI_TYPE = #CONST
LIN P1
LIN P2
PTP PSTART
CIRC PAUX, PEND
; -----
PTP HOME
END
```

Algorithm 3: SPLINE block with SPL commands

```
; ----- Initialization Section -----
DECL AXIS HOME ;
HOME = {AXIS: A1 0, A2 -120, A3 135, A4 0, A5 75, A6 90}
PTP HOME

; ----- Defining Points -----
P1 = {X 1030, Y 350, Z 1300, A 160, B 45, C 130}
P2 = {X 700, Y -300, Z 1000, A 23, B 230, C -90}
P3 = {X 980, Y -238, Z 718, A 133, B 66, C 146}
P4 = {X 925, Y -285, Z 758, A 102, B 45, C 180}
P5 = {X 867, Y -192, Z 688, A 155, B 75, C 160}
; ----- Main Section -----
SPLINE WITH $VEL.CP = 0.75
SPL P1
SPL P2
SPL P3
SPL P4
SPL P5
ENDSPLINE
; -----
PTP HOME
END
```

	PTP	LIN/CIRC	SPLINE
Movement	Start-Stop	Continuous	Continuous
Positioning	Exact	Approximate	Exact
Time Setting	No	No	Yes

Table 5.4: Comparison of different robot motion programming strategies.

After developing the robot program, the final step involves executing the trajectories on the system. The following Figure. 5.9 depicts the *SPIDE-TP Platform* during one of the trial implementations.

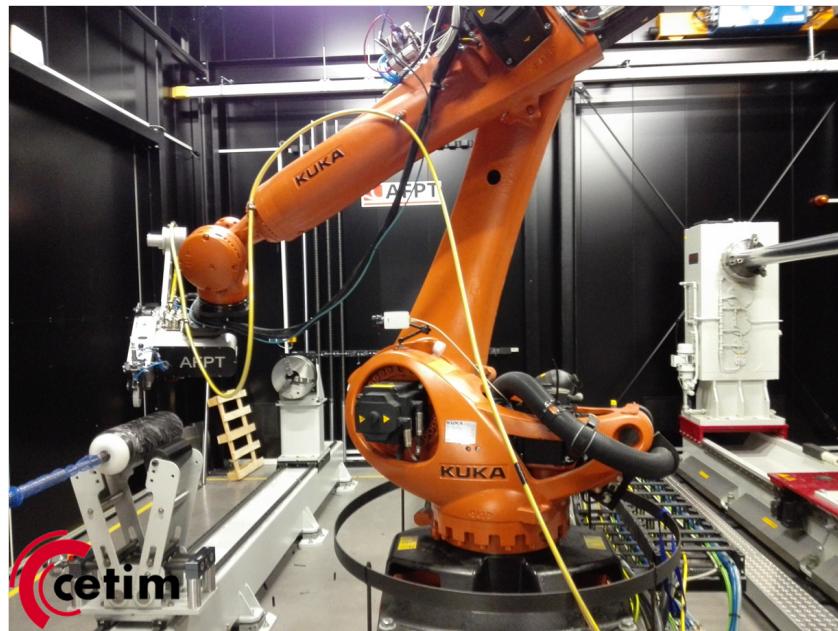


Figure 5.9: SPIDE-TP Platform during trial implementations, courtesy CETIM.

5.3 Contributions

The contributions made within the framework of this master thesis, with respect to this Chp. 5, are as follows:

- Time-optimal trajectories for the SPIDE-TP Platform for the considered workpiece were generated.
- Simulation results promise a reduction of the processing time to **40%** in constant speed case and up to **75%** in the time-optimal solution.
- The trajectories obtained are smoother and show less discontinuities, especially near the dome phase.
- Proposed methods are being implemented on the industrial platform.

Conclusion

The robotic system of the *SPIDE-TP Platform* is kinematically redundant. Currently, this redundancy is not taken into account as the robot motion is generated directly from the post processor of Composicad software and it is not optimal. Also to counter balance the high price of thermoplastic composite tapes (the raw material used for the tape winding), it is necessary to increase the productivity. The general objective of this thesis work was to take into account this kinematic redundancy and increase the average speed of production.

Figure 5.10 summarizes the comprehensive methodology followed in this thesis work in order to achieve the said goals. The CAD model of the system was developed on CATIA V5 to run collision checks and simulations. The time-optimal trajectories were generated by a graph based search using dynamic programming. Optimal solutions are being implemented on the industrial *SPIDE-TP Platform*.

Major contributions from this thesis work include the development of the CAD model for the industrial platform. The CAD model facilitates the computational algorithm in generating optimal solutions. The collision detection achieved from the CAD model is very intensive and robust. It also allows in simulating the solutions and visualizing the robot and positioner motions. The simulation results promise a reduction in the processing time up to **75%** (from previously **14 sec** to **3 sec**) and the trajectories are smoother with less discontinuities.

The comprehensive methodology proposed in this work will be presented and published at *The 3rd International Conference on Mechanical Engineering (ICOME), 2017*, to be held in Surabaya, Indonesia; on October 5th-6th, 2017.

Further work in this area corresponds to generating and implementing optimal solutions for winding paths with different geometries and complexities. For workpiece liners larger in lengths, the robot location on linear axis should not be fixed and the task graph should be computed with all 8 DOF. Another aspect to be considered is the calibration of the system to achieve a perfect dimensional accuracy in the CAD model with respect to the actual system. It could also be interesting to include the optimization of the workpiece placement and cell layout in order to further increase the productivity of the process.

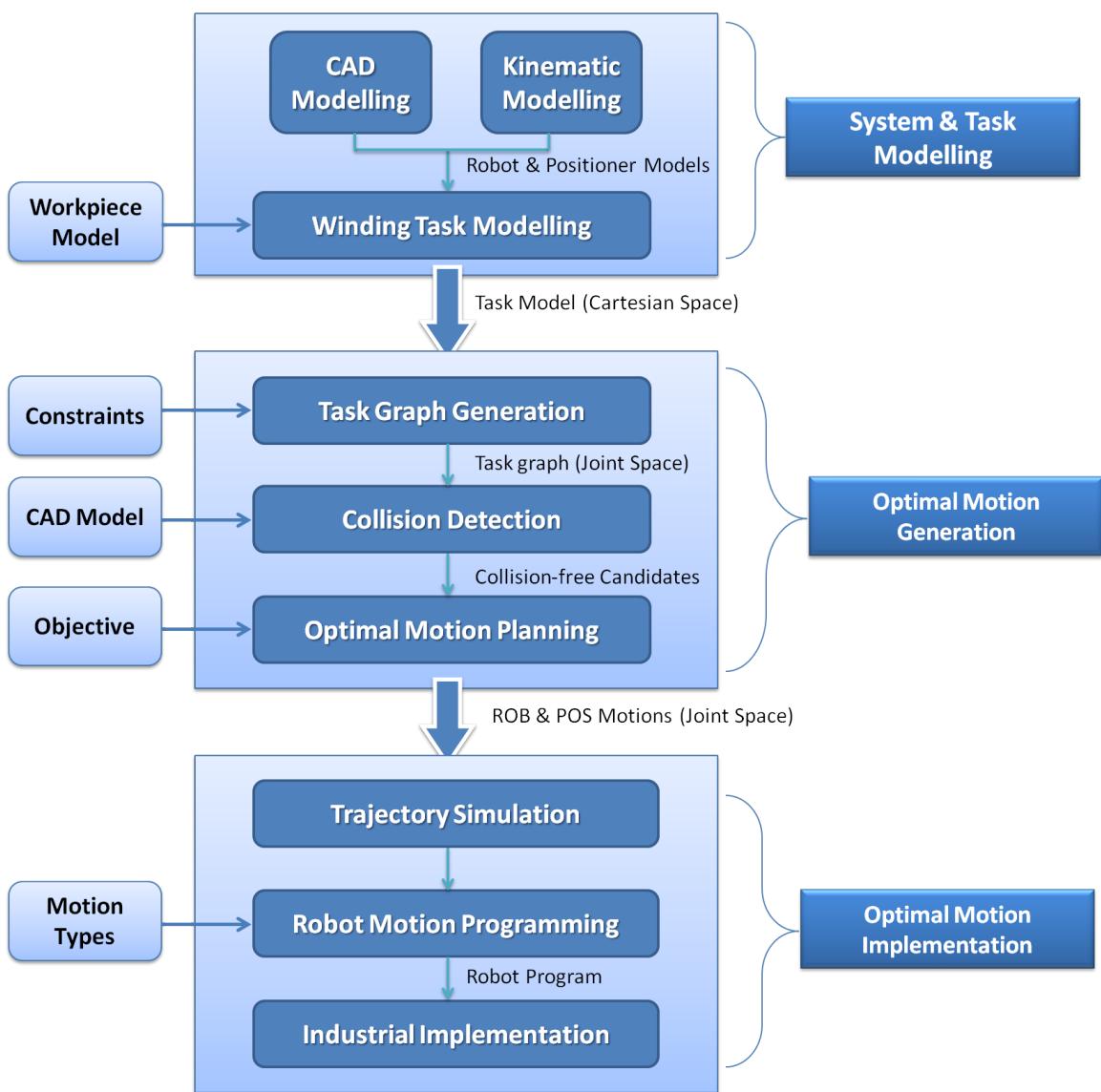


Figure 5.10: Comprehensive methodology followed for increasing the productivity of ATW.

APPENDIX A

Robot Parametrization & Geometric Modelling

This chapter describes the robot parametrization and the direct & inverse geometric modelling methodology that was implemented in this work.

A.1 Robot Parameters

An independent approach was implemented in order to determine the robot parameters. The direction for the rotation of the joint axes, as defined by the manufacturers, is shown in the Fig. A.1. Following the same convention, the coordinate frames for each of the robot joint was defined as shown in the Fig. A.2. In order to facilitate the computations, the 'y-axis' for each joint is considered to be in the same (or opposite, in case of joint 1) direction.

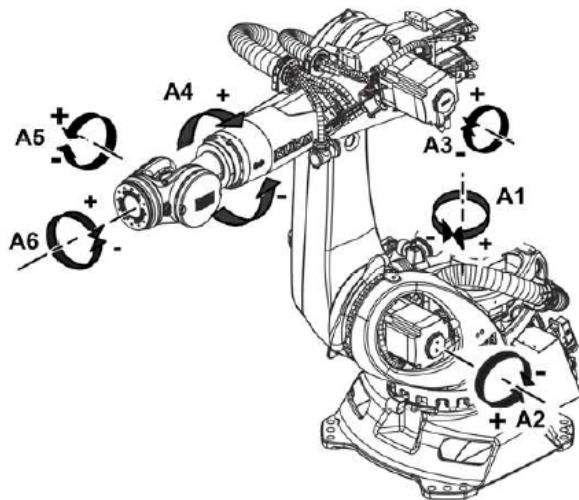


Figure A.1: Directions for the rotation of the joint axes [6, KUKA Specifications].

Figure A.2 also shows the major dimensions of the robot and the corresponding parameters were that determined are as shown in the Table A.1. The first three columns represent the position of the origin of the current joint frame with respect to the previous frame (expressed

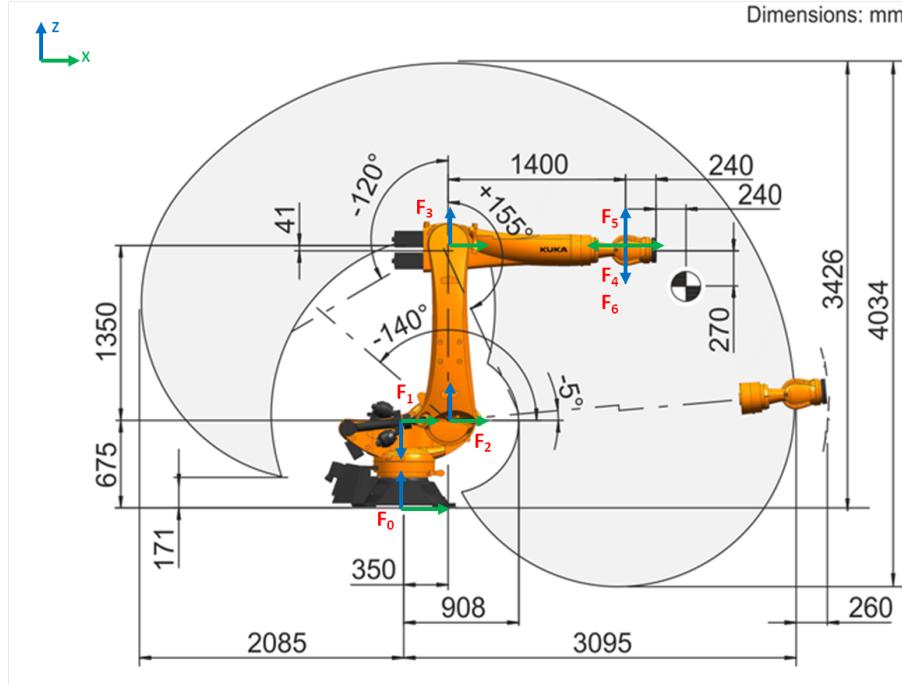


Figure A.2: KR210 R3100 ultra working envelope, side view, representing the joint frames [6, KUKA Specifications].

in *millimetres*) along the respective coordinate axes. The last three columns represent the orientation of the current frame with respect to the previous frame (expressed in *radians*); where ψ , θ and ϕ being the rotations around the X-axis, Y-axis and the Z-axis respectively. However, the order of rotation for the joints 1,4,6 and the joints 2,3,5 is different. This is explained in the following section.

Joint	x	y	z	ψ	θ	ϕ
1	0	0	675	π	0	q_1
2	350	0	0	$-\pi$	q_2	0
3	1350	0	0	0	q_3	0
4	1400	0	-41	0	$\frac{-\pi}{2}$	q_4
5	0	0	0	0	$\frac{\pi}{2} + q_5$	0
6	0	0	0	0	$\frac{-\pi}{2}$	q_6

Table A.1: Robot Parameters

A standard approach to determine the robot based on the Denavit-Hartenberg convention as explained in [18, Khalil & Dombre, 2004] could also be implemented.

A.2 Direct Geometric Model

The Direct Geometric Model (DGM) is a set of relations that define the location of the robot flange frame expressed in the robot base frame, as a function of the joint coordinates. For a 6-axis serial manipulator, the Eqn. A.1 gives the DGM as a serial multiplication of the homogeneous transformation matrices.

$$\boldsymbol{rob}(\vec{q}_r) = ^{RB}\boldsymbol{T}_{RF} = ^{RB}\boldsymbol{T}_1(q_1).{}^1\boldsymbol{T}_2(q_2).{}^2\boldsymbol{T}_3(q_3).{}^3\boldsymbol{T}_4(q_4).{}^4\boldsymbol{T}_5(q_5).{}^5\boldsymbol{T}_{RF}(q_6) \quad (\text{A.1})$$

The homogeneous transformation matrices (4x4) comprises of a rotation matrix (3x3) dependent on the orientation parameters of the robot and a translational vector (3x1) dependent on the position parameters, as given by the Eqn. A.2.

$${}^{j-1}\boldsymbol{T}_j = \begin{bmatrix} \boldsymbol{Rot}(\vec{\varphi}_j) & \vec{P}_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ where; } \boldsymbol{\varphi}_j = (\psi_j, \theta_j, \phi_j)^T \text{ and } \boldsymbol{P}_j = (x_j, y_j, z_j)^T \quad (\text{A.2})$$

The rotation matrix is formed by the multiplication of the rotation matrices around individual axes, in the defined order. For joints 1,4,6 the order of rotation is X-Y-Z whereas for the joints 2,3,5 it is Z-X-Y. This is because the axis of rotation for the joints 2,3,5 is defined to be the Y-axis as opposed to that of Z-axis for joints 1,4,6. The Eqns. A.3 & A.4 give the rotation matrices for both the cases.

$$\boldsymbol{Rot}(\vec{\varphi}_j) = \boldsymbol{Rot}_X(\psi). \boldsymbol{Rot}_Y(\theta). \boldsymbol{Rot}_Z(\phi) \quad \forall j = 1, 4, 6$$

$$= \begin{bmatrix} \cos\theta_j \cos\phi_j & -\cos\theta_j \sin\phi_j & \sin\theta_j \\ \sin\psi_j \sin\theta_j \cos\phi_j + \cos\psi_j \sin\phi_j & -\sin\psi_j \sin\theta_j \sin\phi_j + \cos\psi_j \cos\phi_j & -\sin\psi_j \cos\theta_j \\ -\cos\psi_j \sin\theta_j \cos\phi_j + \sin\psi_j \sin\phi_j & \cos\psi_j \sin\theta_j \sin\phi_j + \sin\psi_j \cos\phi_j & \cos\psi_j \cos\theta_j \end{bmatrix} \quad (\text{A.3})$$

$$\boldsymbol{Rot}(\vec{\varphi}_j) = \boldsymbol{Rot}_Z(\phi). \boldsymbol{Rot}_X(\psi). \boldsymbol{Rot}_Y(\theta) \quad \forall j = 2, 3, 5$$

$$= \begin{bmatrix} -\sin\phi_j \sin\psi_j \sin\theta_j + \cos\phi_j \cos\theta_j & -\cos\psi_j \sin\phi_j & \sin\phi_j \sin\psi_j \cos\theta_j + \cos\phi_j \sin\theta_j \\ \cos\phi_j \sin\psi_j \sin\theta_j + \sin\phi_j \cos\theta_j & \cos\psi_j \cos\phi_j & \cos\phi_j \sin\psi_j \cos\theta_j + \sin\phi_j \sin\theta_j \\ -\cos\psi_j \sin\theta_j & \sin\psi_j & \cos\psi_j \cos\theta_j \end{bmatrix} \quad (\text{A.4})$$

The corresponding transformation matrices of the Eqn. A.1 are given in the following Eqns. A.5,A.6,A.7.

$${}^{RB}\boldsymbol{T}_1 = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & 0 \\ -\sin q_1 & \cos q_1 & 0 & 0 \\ 0 & 0 & -1 & 675 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1\boldsymbol{T}_2 = \begin{bmatrix} \cos q_2 & 0 & \sin q_2 & 350 \\ 0 & -1 & 0 & 0 \\ \sin q_2 & 0 & -\cos q_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.5})$$

$${}^2\boldsymbol{T}_3 = \begin{bmatrix} \cos q_3 & 0 & \sin q_3 & 1350 \\ 0 & 1 & 0 & 0 \\ -\sin q_3 & 0 & \cos q_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^3\boldsymbol{T}_4 = \begin{bmatrix} 0 & 0 & -1 & 1400 \\ \sin q_4 & \cos q_4 & 0 & 0 \\ \cos q_4 & -\sin q_4 & 0 & -41 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.6})$$

$${}^4\boldsymbol{T}_5 = \begin{bmatrix} -\sin q_5 & 0 & \cos q_5 & 0 \\ 0 & 1 & 0 & 0 \\ -\cos q_5 & 0 & -\sin q_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^5\boldsymbol{T}_{RF} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ \sin q_6 & \cos q_6 & 0 & 0 \\ \cos q_6 & -\sin q_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.7})$$

A.3 Inverse Geometric Model

The inverse geometry computations can be divided into two parts, one with the position information of the robot flange that gives the solutions to the joints 1,2 and 3; and the orientation information used to compute the values for joints 4,5,6.

First, considering only the position information, we have the following equation:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = {}^{\textbf{RB}}\mathbf{T}_{\textbf{RF}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (\text{A.8})$$

Pre-multiplying the above equation with ${}^1\mathbf{T}_{\textbf{RB}}$, which was already known from the direct geometric model, we get

$${}^1\mathbf{T}_{\textbf{RB}} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = {}^1\mathbf{T}_{\textbf{RF}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (\text{A.9})$$

Extracting element (2,1), we get the q_1 values as follows:

$$\begin{aligned} -p_x \sin q_1 + p_y \cos q_1 &= 0 \\ q_1 &= \text{atan2}(-p_y, p_x) \\ \text{also; } q_1 &= \text{atan2}(p_y, -p_x) \end{aligned} \quad (\text{A.10})$$

Extracting the elements (1,1) & (3,1) from the Eqn. A.9, we get the following set of linear equations:

$$\begin{aligned} L2 \cos q_2 + L3' \cos(q_2 + q_3') &= p_x - L1 \\ L2 \sin q_2 + L3' \sin(q_2 + q_3') &= p_y \end{aligned} \quad (\text{A.11})$$

where $L3' = \sqrt{L3^2 + D3^2}$, $L1 = 350$, $L2 = 1350$, $L3 = 1400$, $D3 = 41$

First solving for q_3' , we get

$$\begin{aligned} \cos q_3' &= \frac{(p_x - L1)^2 + (p_y)^2 - L2^2 - L3'^2}{2L2L3'} \\ q_3' &= \text{atan2}(\pm \sqrt{1 - \cos^2 q_3'}, \cos q_3') \end{aligned} \quad (\text{A.12})$$

However, q_3' is the joint angle without considering the offset between joints 3 & 4. Thus the actual value can be obtained as follows:

$$q_3 = q_3' - \eta; \text{ where } \eta = \text{atan2}(D3, L3) \quad (\text{A.13})$$

Then, solving for the joint coordinate q_2 qs follows:

$$\begin{aligned} \sin q_2 &= \frac{B1p_y - B2(p_x - L1)}{B1^2 + B2^2} \\ \cos q_2 &= \frac{B1(p_x - L1) + B2p_y}{B1^2 + B2^2} \\ q_2 &= \text{atan2}(\sin q_2, \cos q_2) \end{aligned} \quad (\text{A.14})$$

where $B1 = L2 + L3' \cos q_3'$ and $B2 = L3' \sin q_3'$.

After getting the joint coordinates q_1, q_2, q_3 from Eqns. A.10,A.14,A.13 respectively; we can now solve for the joints 4,5,6 using only the orientation information of the robot flange. Thus, from the Eqn. A.1, we consider only the orientation matrices \mathbf{A} from the homogeneous transformations \mathbf{T} as follows:

$$\begin{aligned} {}^{RB}\mathbf{A}_{RF} &= {}^{RB}\mathbf{A}_3(q_1, q_2, q_3) \cdot {}^3\mathbf{A}_{RF}(q_4, q_5, q_6) \\ {}^3\mathbf{A}_{RB}(q_1, q_2, q_3) \cdot {}^{RB}\mathbf{A}_{RF} &= {}^3\mathbf{A}_4(q_4) \cdot {}^4\mathbf{A}_5(q_5) \cdot {}^5\mathbf{A}_{RF}(q_6) \end{aligned} \quad (\text{A.15})$$

The elements on the RHS are unknown; whereas the elements on the LHS are known and let us consider them to be as follows:

$${}^3\mathbf{A}_{RB} \cdot {}^{RB}\mathbf{A}_{RF} = \begin{bmatrix} F_x & G_x & H_x \\ F_y & G_y & H_y \\ F_z & G_z & H_z \end{bmatrix} \quad (\text{A.16})$$

On pre-multiplying the above equation with ${}^4\mathbf{A}_3$, we get:

$${}^4\mathbf{A}_3(q_4) \begin{bmatrix} F_x & G_x & H_x \\ F_y & G_y & H_y \\ F_z & G_z & H_z \end{bmatrix} = {}^4\mathbf{A}_5(q_5) \cdot {}^5\mathbf{A}_{RF}(q_6) \quad (\text{A.17})$$

From the Eqn. A.17, extracting element (2,3) gives:

$$\begin{aligned} H_y \cos q_4 - H_z \sin q_4 &= 0 \\ q_4 &= \text{atan2}(H_y, H_z) \\ \text{also; } q_4 &= \text{atan2}(H_y, H_z) + \pi \end{aligned} \quad (\text{A.18})$$

Now, extracting the elements (1,3) & (3,3) we get the following equations:

$$\begin{aligned} H_y \sin q_4 + H_z \cos q_4 &= \sin q_5 \\ -H_x &= \cos q_5 \\ q_5 &= \text{atan2}(H_x, H_y \sin q_4 + H_z \cos q_4) \end{aligned} \quad (\text{A.19})$$

Similarly, extracting elements (2,1) & (2,2) gives:

$$\begin{aligned} F_y \cos q_4 - F_z \sin q_4 &= \sin q_6 \\ G_y \cos q_4 - G_z \sin q_4 &= \cos q_6 \\ q_6 &= \text{atan2}(F_y \cos q_4 - F_z \sin q_4, G_y \cos q_4 - G_z \sin q_4) \end{aligned} \quad (\text{A.20})$$

Bibliography

- [1] “Advanced fibre placement technology (afpt).” [Online]. Available: <http://www.afpt.de/welcome/>
- [2] “Centre technique des industries mécaniques [cetim].” [Online]. Available: <http://www.cetim.fr/enr>
- [3] “Composicad.” [Online]. Available: <http://www.composicad.com/Help/Help.html>
- [4] “Le groupement technocampus.” [Online]. Available: <http://www.technocampus.fr/fr>
- [5] *KUKA System Software 8.2, Operating and Programming Instructions for System Integrators*. KUKA Roboter GmbH, 2011.
- [6] *KR QUANTEC ultra, With F and C Variants, Specification*. KUKA Roboter GmbH, 2016.
- [7] V. Benson, D. Gill, J. Nielsen, H. Mansouri, and N. Shepherd, “Multiple axes fiber placement machine,” Aug. 1 2000, uS Patent 6,096,164. [Online]. Available: <https://www.google.com/patents/US6096164>
- [8] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.
- [9] S. Caro, C. Dumas, S. Garnier, and B. Furet, “Workpiece placement optimization for machining operations with a kuka kr270-2 robot,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2921–2926.
- [10] P. Debout, H. Chanal, and E. Duc, “Tool path smoothing of a redundant machine: Application to automated fiber placement,” *Computer-Aided Design*, vol. 43, no. 2, pp. 122–132, 2011.
- [11] A. Dolgui and A. Pashkevich, “Enhanced optimisation techniques for off-line programming of laser cutting robots,” Research Report G2I-EMSE 2007-500-015, Ecole des Mines de Saint-Etienne, Tech. Rep., Nov. 2006.
- [12] J. Gao, A. Pashkevich, and S. Caro, “Manipulator motion planning in redundant robotic system for fiber placement process,” in *New Trends in Mechanism and Machine Science*. Springer, 2017, pp. 243–252.
- [13] ——, “Optimization of the robot and positioner motion in a redundant fiber placement workcell,” *Mechanism and Machine Theory*, vol. 114, pp. 170–189, 2017.
- [14] D. Gay, *Composite materials: design and applications*. CRC press, 2014.

- [15] R. Grone, L. Schnell, and W. Vearil, “Composite tape laying machine and method,” Dec. 10 1985, uS Patent 4,557,783. [Online]. Available: <https://www.google.com/patents/US4557783>
- [16] W. J. B. Grouve, *Weld strength of laser-assisted tape-placed thermoplastic composites*. University of Twente, 2012.
- [17] K. Kazerounian and A. Nedungadi, “Redundancy resolution of serial manipulators based on robot dynamics,” *Mechanism and machine theory*, vol. 23, no. 4, pp. 295–303, 1988.
- [18] W. Khalil and E. Dombre, *Modeling, identification and control of robots*. Butterworth-Heinemann, 2004.
- [19] C. Krombholz, D. Delisle, and M. Perner, “Advanced automated fibre placement,” *Advances in Manufacturing Technology XXVII*, pp. 411–416, 2013.
- [20] L. Li, D. Xu, X. Wang, and M. Tan, “A survey on path planning algorithms in robotic fibre placement,” in *Control and Decision Conference (CCDC), 2015 27th Chinese*. IEEE, 2015, pp. 4704–4709.
- [21] G. Marsh, “Automating aerospace composites production with fibre placement,” *REIN-FORCED plastics*, vol. 55, no. 3, pp. 32–37, 2011.
- [22] T. Martinec, J. Mlýnek, and M. Petruš, “Calculation of the robot trajectory for the optimum directional orientation of fibre placement in the manufacture of composite profile frames,” *Robotics and Computer-Integrated Manufacturing*, vol. 35, pp. 42–54, 2015.
- [23] J. Mlýnek and T. Martinec, “Mathematical model of composite manufacture and calculation of robot trajectory,” in *Mechatronics-Mechatronika (ME), 2014 16th International Conference on*. IEEE, 2014, pp. 345–351.
- [24] S. R. Munasinghe, M. Nakamura, S. Goto, and N. Kyura, “Trajectory planning for industrial robot manipulators considering assigned velocity and allowance under joint acceleration limit,” *Intl. Journal of Control, Automation and Systems*, vol. 1, no. 1, pp. 68–75, 2003.
- [25] A. Pashkevich and A. Dolgui, “Manipulator motion planning for high-speed robotic laser cutting,” *International Journal of Production Research*, vol. 47, no. 20, pp. 5691–5715, 2009.
- [26] S. T. Peters, *Handbook of composites*. Springer Science & Business Media, 2013.
- [27] L. Sabourin, K. Subrin, R. Cousturier, G. Gogu, and Y. Mezouar, “Redundancy-based optimization approach to optimize robotic cell behaviour: application to robotic machining,” *Industrial Robot: An International Journal*, vol. 42, no. 2, pp. 167–178, 2015.
- [28] Y. Shen, K. Hüper *et al.*, “Optimal joint trajectory planning for manipulator robot performing constrained motion tasks,” in *Australasian Conf. on Robotics and Automation*, 2004.
- [29] B. Shirinzadeh, G. Alici, C. W. Foong, and G. Cassidy, “Fabrication process of open surfaces by robotic fibre placement,” *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 1, pp. 17–28, 2004.
- [30] B. Shirinzadeh, C. Wei Foong, and B. Hui Tan, “Robotic fibre placement process planning and control,” *Assembly Automation*, vol. 20, no. 4, pp. 313–320, 2000.

- [31] J. Sloan, “Atl & afp: Defining the megatrends in composite aerostructures,” *HIGH PERFORMANCE COMPOSITES*, vol. 16, no. 4, p. 68, 2008.
- [32] K. Subrin, L. Sabourin, F. Stephan, G. Gogu, M. Alric, and Y. Mezouar, “Analysis of the human arm gesture for optimizing cutting process in ham deboning with a redundant robotic cell,” *Industrial Robot: An International Journal*, vol. 41, no. 2, pp. 190–202, 2014.