Aerospace Technology Congress, 8-9 October 2019, Stockholm, Sweden
Swedish Society of Aeronautics and Astronautics (FTF)

Airframe Sealing Automation Using Snake-robot

Luís Gonzaga Trabasso[*1,2], Doglas Negri[*2], Walter Antonio Kapp[*2], Magnus Engström[*3], Guilherme Sartori Natal[*1] and Kerstin Johansen[*4]

*1 Department of Mechanical Engineering, Aeronautics Institute of Technology, SJCampos, SP/Brazil
E-mail: gonzaga@ita.br, guilesn@gmail.com
*2 SENAI Institute of Innovation, Joinville, SC/Brazil
E-mail: luis.gonzaga@sc.senai.br, doglas.negri@sc.senai.br, walter.kapp@sc.senai.br
*3 SAAB AB, Aeronautics, Linköping, Sweden
E-mail: magnus.engstrom@saabgroup.com
*4 Department of Management and Engineering, Linköping University, Linköping, Sweden
E-mail: kerstin.johansen@liu.se

Abstract

The work described herein have had the objective of carrying out a technical-economic feasibility study of a snake-robot for sealant application in an aircraft fuselage. The execution of the project brought knowledge related to the design and development of robots, constrained Cartesian trajectory generation, optimization methods for inverse kinematics computation, collaborative robotics and 3D simulations besides experiments related to aircraft components and application of sealants in the fuselage.

Concerning the technical feasibility of the project, the solution proposed proved to be an adequate solution for the proposed task. The sealant application trajectories in fillets (union of two plates of the fuselage) and fasteners were tested, being all well executed by the robot in a virtual environment, without collisions and without relevant complications. All the trajectories used for the evaluation were taken from the test framework provided by Saab and respecting the process specifications described for this task.

Economical evaluation of ROM (Rough Order of Magnitude) gives a result showing fairly good pay-back time although the potentials would be even better with a lower cost.

A list of commercial components for the manufacturing of the snake robot was specified, such as: motors, reducers, vision system, components for programming and controlling the robot (drivers, software and robot interface with the user) etc. It is important to mention that, despite the design of the robot drivers, the CAD model developed here needs further detailing for fabrication purposes. Structural analysis must be carried out in the structure and, after any changes are made, evaluations and corrections must be done. A study about the dimension and mass reductions of an existing sealant dispenser was carried out, where the cartridge and the sealant application valve were chosen to be coupled directly to the robotic arm.

In general, all activities were executed according to the proposed schedule. The project presented a very high potential and could be applied not only to the aeronautical industry (which was the main objective of the present work), but also to other fields of the industry. Therefore, a continuation of this project is highly recommended, such that the proposed snake robot can be manufactured and physically tested.

Keywords: SNAKE-Robot, Aircraft Industry, Sealing Automation, Collaborative Robotics.

1 Introduction

In the aircraft industry normally most part of the airframe must be sealed. For example, the internal structure of the wing defines the tank and needs to be sealed to prevent any leakage. Due to the weight optimization, there is no internal tank. The fuel is in direct contact with the inner side of the wing skin.

To seal a wing a small and thin worker have to go inside of the airframe through inspection openings. For a big commercial aircraft, this task is easy, but for an executive jet, or combat jet the sealing task could be very hard, or even impossible, putting constrains to the designer when optimizing the structure for aerodynamics and loads.

DOI
10.3384/ecp19162022

The internal fuselage structure is also sealed in order to provide a pressurized cabin. In addition sealing is often used for corrosion protection, especially when mixing airframe part materials and in areas with high level of moisture.

The standard anthropomorphic robots cannot fit to this task due the lack of degrees of mobility necessary to avoid collisions in the restricted workspace. This article has proposed a study of a robot design similar to a snake morphology.

The kinematic chain of a snake robot has several redundant degrees of freedom. This makes it possible to control a path and a shape of a robot, making it able to insert the robot and work inside the airframe, avoiding collisions.

The kinematic solver and the trajectory planner give some new ideas of how to realize a feasible robot design.

2 Project Objective

2.1 General Objective

The objective of this project was a feasibility study of a concept of a virtual snake-robot for automatic application of sealant in aeronautical structures.

The developed robot has been tested (virtually) through the generation of trajectories, kinematics solution strategies, accessibility simulations and the execution of sealing tasks. Subsequently it has been evaluated for technical and economic feasibility for implementation at Saab Aeronautics, a Swedish industrial partner in this project.

2.2 Specific Objectives and Established Goals

Specific Objectives	Established Goals
Definition of an application and requirements for automated sealing that can be used for additional evaluation of the snake robot and the sealing agent.	Define a representative application for sealing, which includes detailed requirements regarding product aspects: sealing; and of production: interface man / machine. Delivery includes CAD application models that will be used for further system evaluation.
Design the conceptual 3D model of the snake robot based on industrial components, including the kinematic model.	Design a low-TRL snake robot for an industrial application of sealant in aircraft fuselage using CAD modeling, CAE and simulation tools.
Study to reduce mass and dimensions of an existing sealant application tool (used in larger robots), to be coupled to the snake robot.	Considering the sealant applicator developed by ITA, assess possibilities of reduction of dimensions and mass of the device.
Technical and economic criteria used for evaluating the feasibility of the proposed solution. Assess the potential for establishing an international network among robot integrators Swedish, Brazilians and component suppliers.	Considering the constraints of the process described above, assess the performance of the robot from the point of view of implementing the task of sealing. Still, evaluate the manufacturing costs of the robot, in order to observe the economic viability of the project.

Table 1: Specific objectives

3 Scope of the project execution

3.1 Definition of an application and requirements

The application is defined by Saab and consists of a virtual DMU – Digital Mock-Up, of a section of an aircraft fuselage with the overall dimension 1 m³. The DMU contains the main requirements for the robot geometrical accessibility, quality requirements of sealing of fillets and rivets including process cycle times, see Fig. 1.

3.2 Automated sealing process

The sealing processes that are of interest for automation are the fillet sealing within the structure (not the aerodynamic sealing) and the coating of the fasteners (photos below).

The ASASR project uses the virtual test structure (DMU) to simulate the sealing process using the snake robot and the end effector. Specific parts of interest in the test are displayed on the following pages.

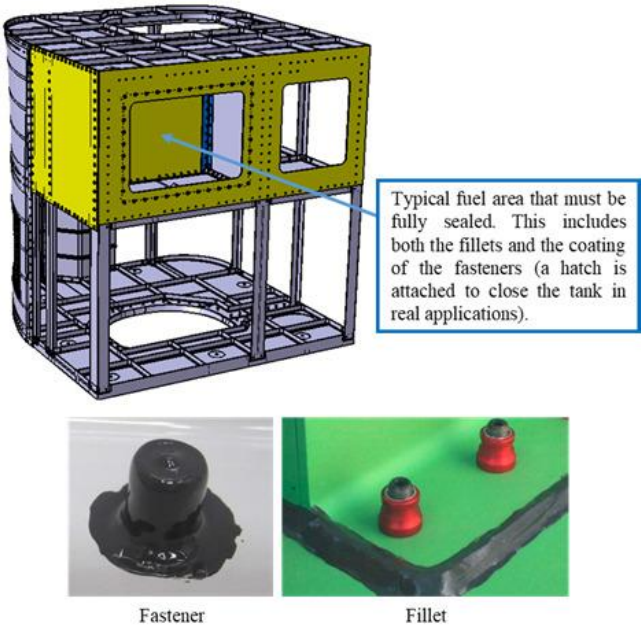

Typical fuel area that must be fully sealed. This includes both the fillets and the coating of the fasteners (a hatch is attached to close the tank in real applications).

Fig. 1: Test airframe and project requirements

3.3 Product requirements

As described above, the ASASR project focuses on the fillet sealing and the coating of fasteners of the airframe. The PR1776 B2 (PPG) sealant was chosen for this project. It is important to mention that, when mixing the two materials that form the sealant, air is introduced to the mixture, which will result in quality problems (with a risk of rework). In this way, it is preferable to use the existing cartridge system for the sealing end-effector.

3.4 Production requirements

The following production requirements are defined for the use of a snake robot with sealant application system for fastener coating and fillet sealing:

- The application cycle time, including the positioning between two neighboring rivets, must be less than 10 s/rivet;

- The fillet sealing must reach a minimum speed of 50 mm/s;

- The installation should not require physical safety barrier and must still conform to the guidelines of the relevant European collaborative robots machines;

- The reach position is found up to 1 m, from the robot's entry into the aircraft structure;

- The error of product position, both for rivets and for fillets has a maximum tolerance of +/- 2 mm. Therefore, it is necessary to use a calibration system for accurately positioning the robot

- Snake robot programming should be done offline, using commercially available software, such as CATIA V5 Robotics, with appropriate software.

4 Snake robot project

The kinematic chain model, which is the basis for the development of this project, is a highly flexible concept when it comes to performing tasks in confined environments (limited accessibility and restricted workspaces). This high flexibility is guaranteed mainly by the hyper redundant robot. A first analysis is related to kinematics modeling of this robot, where the models of direct and reverse kinematics are considered. Then there are studies to generate the path to be followed by the snake robot while performing tasks in a 3D environment. Finally, the dimensioning of the elements and the CAD modeling of snake robot.

4.1 Kinematic model of the snake robot

For robotic chains of any kind, the creation of kinematic models is of paramount importance to the knowledge of the dynamic behavior of the robot, or for the characterization of the stresses suffered by certain joints. Skeleton models, associated with the Denavit-Hartenberg parameters [1] (serial robots) for the translations and rotations of coordinate systems of each joint, are commonly used in these robotic cinematic models, as showed in Fig. 2. In this study, this application added to some techniques of numerical optimization is used in the models of direct and inverse kinematics of the robot-snake. Following are the kinematic models employed for this solution.

4.1.1 Direct kinematics

Initially, considering the process requirements presented by company Saab, it was proposed a distribution of joints that made up the kinematic chain of the snake robot. This kinematic chain represented schematically in Fig. 2, consists of a pedestal (J_1), which provides the robot rotation and inversion complete. Followed by five joints snake type (J_2-J_6). These snake joints have the particularity of maintaining the alignment of the links, however, they rotate the axis in 90° joints in consecutive joints. Finally, like joints 7 and 8, they are part of the semi-spherical robot of the snake that assists in the execution of the sealing tasks.

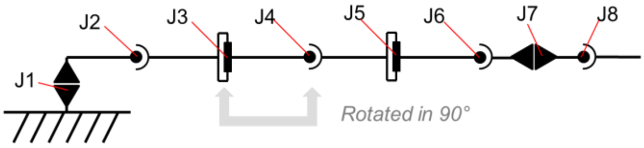


Fig. 2: Kinematic Model

Then, an 8-degree freedom skeleton model for the direct kinematics of the snake robot is proposed and developed in numerical calculations that solve in Scilab 6.0.1 (Fig. 3). The main objective of this model is the understanding of the distribution of joints and their axes of local coordinate referred to in the workspace of the robot.

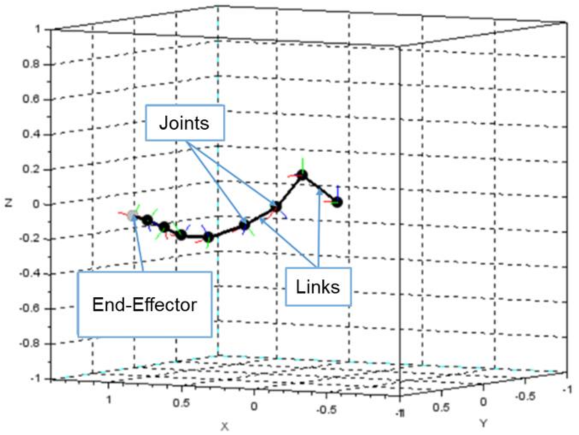


Fig. 3: Skeleton model for direct kinematics solution

Table 2 presents the D-H parameters used in the design of the kinematic model.

The same skeletal model using Denavit-Hartenberg parameters, as well as optimization techniques (PSO - Particle Swarm Optimization), is used for the solution of the inverse kinematics of the robot.

4.1.2 Inverse kinematics

Inverse kinematics of a robot, nothing more, than from a three-dimensional cartesian coordinate (X, Y, Z), directed (V_X, V_Y, V_Z) or not, the effector, through algebraic solutions (numerical or analytical), a set of angular amplitudes joints (in this case, $\theta_1, \ldots, \theta_8$) [2], which meets this solution.

*1 – All inout information of angles must be in radians [rad].
*2 – All input information of length must be in meters [m].

Link	θ^{*1}	d^{*2}	a^{*2}	α^{*1}
# 1	θ_1	0,022	0,277	$\pi/2$
# 2	θ_2	0	0,249	$\pi/2$
# 3	θ_3	0	0,212	$\pi/2$
# 4	θ_4	0	0,196	$\pi/2$
# 5	θ_5	0	0,196	$\pi/2$
# 6	θ_6; $-\pi/2$; 0	0; 0; 0,060	0; 0; 0	0; $-\pi/2$; 0
# 7	θ_7; 0; $\pi/2$; 0	0; 0; 0; 0	0; 0; 0; 0,165	0;$\pi/2$; 0; 0
# 8	θ_8	0	0,084	0

Table 2: Denavit-Hartenberg parameters of snake robot

In this study, an iterative numerical solver was proposed in Scilab 6.0.1 to solve the trajectories (oriented three-dimensional Cartesian coordinate sequence), which uses concepts of the optimization method PSO - Particle Swarm Optimization in its solution. This solver receives the cartesian trajectories oriented in the form of a ".txt" file, performs the solution, and returns the angular amplitude of each of the 8 joints $\theta_1, \ldots, \theta_8$ in a file of the same format for each discrete point of the trajectory, as can be seen in Fig. 4.

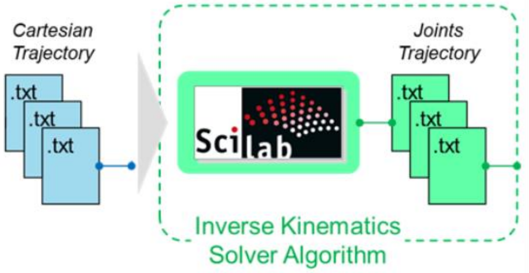


Fig. 4: Inputs and outputs of the inverse kinematics solver

As it is a redundant robot, that is, it has more degrees of freedom (number of θ = 8) than is necessary to execute a given task (X, Y, Z) and oriented (V_X, V_Y, V_Z), a strategy of divide the solution of the inverse kinematics of the robot into two parts is adopted. This strategy consists of, starting from the proposed chain, to create two virtual robots. The first robot consists of the three initial joints of the snake robot, making it possible to position the joint 4 in a known Cartesian coordinate (X_4, Y_4, Z_4) within a working space. For a better positional control of the robot, this known coordinate is strategically positioned in the trajectory of the robot, as can be seen in Fig. 5. After the inverse kinematics proposed for the first robot, find X_4, Y_4 and Z_4 through the PSO, the solution of the second virtual robot containing 5 degrees of freedom is performed. In this case, besides the positioning of the process effector (X_{EE}, Y_{EE}, Z_{EE}), the orientation of the vector V_X must be guaranteed (Fig. 5). For this, two (α, β) of the three Euler angles are required for the orientation to be maintained. These are calculated from the V_X vector itself, such as:

$$V_X = [a \quad b \quad c],$$

$$\beta = sin^{-1}(-c),$$

$$\alpha = cos^{-1}(b, a).$$

Thus, the effector orientation matrix of the process can be calculated as:

$$EE_{orientação} = \begin{bmatrix} cos\,\alpha * cos\,\beta & -\,sin\,\alpha & cos\,\alpha * sin\,\beta \\ sin\,\alpha * cos\,\beta & cos\,\alpha & sin\,\alpha * sin\,\beta \\ -\,sin\,\beta & 0 & cos\,\beta \end{bmatrix}$$

With the remaining vectors of orientation V_Y and V_Z being the result of this matrix of rotation. With this, again the optimization algorithm can perform the search of the angles θ_4, ..., θ_8 that satisfy the considered tolerance.

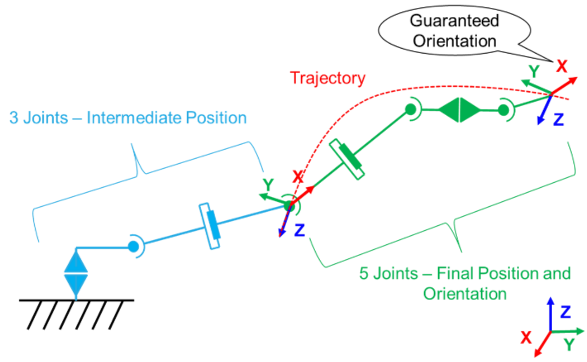

Guaranteed Orientation
Trajectory
3 Joints – Intermediate Position
Y
X
Z
Y
X
Z
5 Joints – Final Position and Orientation
Z
X
Y

Fig. 5: Proposed of strategy to solve the inverse kinematics.

The way the optimization algorithm finds a solution that satisfies the required tolerance is identical for the two steps of the solution strategy (intermediate position and final position and orientation). However, the difference between the two solutions is the objective function to be optimized. For the intermediate positioning, the objective function is defined as:

Minimize $f(\theta_1, \theta_2, \theta_3)$

$$f = \sqrt{(X - \overline{X})^2 + (Y - \overline{Y})^2 + (Z - \overline{Z})^2}$$

where *(X, Y, Z) the desired Cartesian coordinate,*

$(\overline{X}, \overline{Y}, \overline{Z})$ *the optimized Cartesian coordinates.*

and for the positioning and orientation of the effector of process defined as:

Minimize $f(\theta_4, \theta_5, \theta_6, \theta_7, \theta_8)$

$$f = \sqrt{\begin{matrix} \left(\sum V_x - \sum \overline{V_x}\right)^2 + (X - \overline{X})^2 + \\ (Y - \overline{Y})^2 + (Z - \overline{Z})^2 \end{matrix}}$$

where V_x *is the desired orientation vector,*

(X, Y, Z) the desired Cartesian coordinate,

$\overline{V_x}$ *is the optimized orientation vector*

$(\overline{X}, \overline{Y}, \overline{Z})$ *the optimized Cartesian coordinates.*

An important definition is the working limits of each of the snake robot's joints. These limits are used in the optimization algorithm to restrict the search field of the PSO particles. The limits for each of the joints are shown in Table 3.

θ_1	θ_2	θ_3	θ_4
±180°	-90°~+135°	±120°	±120°
θ_5	θ_6	θ_7	θ_8
±120°	±120°	±180°	±90°

Table 3: Angle limits for the joints

In this way, the optimization algorithm seeks to minimize the objective functions. The PSO, an optimization algorithm originally developed by Eberchart and Kennedy [3], consists of, from an initial population of random particles whose members interact locally with each other and are governed by global rules, perform the minimization of any previously defined objective function. The present work uses the fully connected topology (g_{best}), which means that a particle takes the entire population as its topological neighbors.

Initially, each particle has a random position and velocity. The particles interact with each other, informing the best position. With the data, velocities and positions are adjusted for each particle i. The velocity of a particle for the next increment (k + 1) is given as:

$$\boldsymbol{v_{k+1}^i = \omega v_k^i + C_1 R_1 (p_{best}^i - x_k^i) + C_2 R_2 (g_{best}^i - x_k^i);}$$

$$\boldsymbol{i = 1, 2, \dots, N}$$

where, N is the total number of particles, R_1 and R_2 are random values from 0 to 1, p_{best} is the best position for that particle and g_{best} is the best overall position. ω, C_1 and C_2 are the chosen parameters: the first is the inertial component of the particle, and the latter are terms of "reliability" among the particles of the group [4]. The position in the next iteration can be determined as:

$$x_{k+1}^i = x_k^i + v_{k+1}^i$$

where, x_k^i is the position of the particle in the current iteration. Fig. 6 shows the position and velocity of a particle in one step and how the best overall position and the best particle position will influence the next particle position.

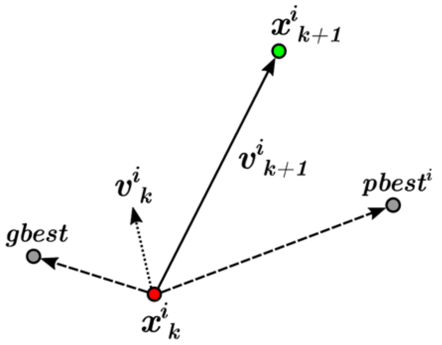


Fig. 6: Present and future position of the particle and speed

The entire optimization process requires a stop condition. For this problem, the set of stop tolerance is 1E-4, or even the maximum number of iterations is reached 150. This procedure is triggered for each discrete coordinate that makes up a path to be followed by the robot.

4.2 Generation of trajectories

A great challenge for the snake robot (hyper redundant serial manipulator) is to develop its interface to the programmer to be able to make use of all the geometric performance that this mechanical system model can offer.

In this feasibility study on the use of a snake robot for the application of aircraft fuselage seals, a versatile graphical interface was adopted to simulate a simplified CAD/CAM system for specific off-line programming application of the virtual robot prototype. This section describes the routines developed with the Rhinoceros 5 software and its Grasshopper 0.9.0076 plug-in for the generation of Cartesian trajectories in different types of tasks. Fig. 7 shows the user input workflow to view the programmed path.

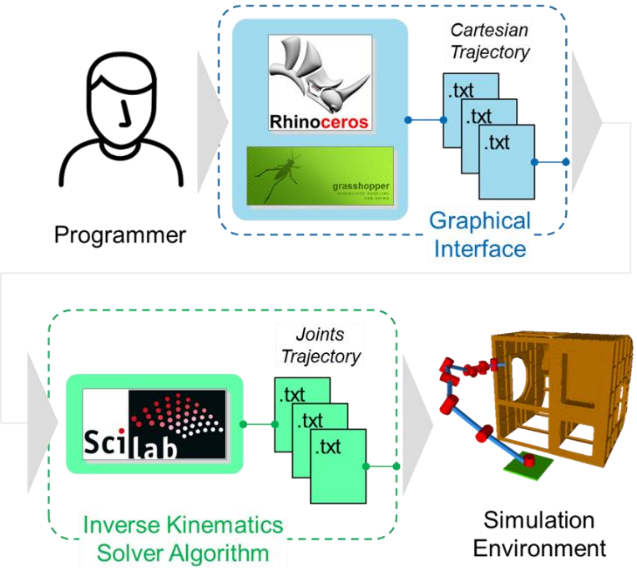


Fig. 7: Overview of workflow for trajectory generation.

To generate a trajectory for the snake robot, steps are followed, consisting of a list of coordinates of sequential points and their respective orientations of the effector; therefore, the programmer must first use the Rhinoceros 5 CAD module to: (i) draw free curves or (ii) extract curves from an existing CAD model (eg, fuselage). In both cases, the curves serve as the basis for the complete definition of the trajectory. Different systematic algorithms developed with Grasshopper especially for this project, segment the curves selected by the programmer and also, in most cases, give orientations to the effector at each segmentation point, generating a text file (Cartesian trajectory). In this step, the text files of the Cartesian trajectory are given as input in Scilab 6.0.1, which implements the inverse kinematics solution algorithm implemented. This procedure is described in Section 4.1, along with a detailed explanation on the kinematic modeling of the snake robot mechanism. The last stage of the trajectory generation workflow is motion simulation. In a dedicated simulation environment generated in LabView 2017 with the help of the Robotics supplement. Joints trajectory text files are used to represent the synchronized angular position of each motor / joint for each trajectory segmentation point.

4.2.1 Types of trajectories

Two categories of main trajectories were designed for this feasibility study: chain insertion and tasks. The task execution trajectories for the application of sealing in aeronautical structures comprise three subcategories: general trajectory, fillet sealing and rivet covers, the latter being divided into two subclasses as to the geometric strategy appropriate for the application (propeller and cone).

Fig. 8: Types of trajectories for the snake robot.schematically shows the types of trajectory categories for the snake robot and its subdivisions. The following sections briefly describe when and how the programmer should apply each trajectory to make use of the mechanical flexibility offered by the snake robot.

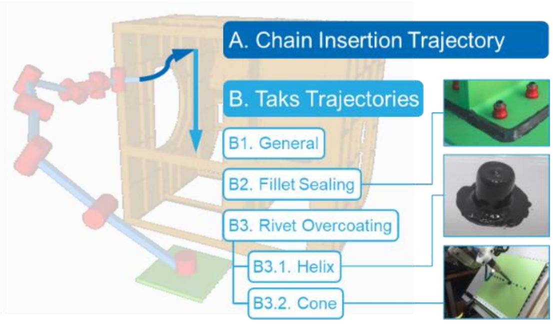


Fig. 8: Types of trajectories for the snake robot.

4.2.1.1 Chain insertion path

This type of trajectory is used for the insertion of the kinematic chain within a restricted working space (obstacles). The objective is to position the final effector ("snake head") as close as possible to the region where a specific task is to be conducted, with a certain amount of control over the path through which the entire kinematic chain ("snake body") moves along this trajectory. This is obtained by controlling the spatial trajectory of the fourth joint of the robot (J4) and the effector (EE) separately, but synchronously. In this sense, the simplest way for the programmer to generate an insertion trajectory is to draw a curve in space to serve as the basis for the movement of the entire snake robot. This curve is called the complete path and is the first input of the algorithm (Fig. 9). Throughout the insertion, the fourth joint and the effector follow different sections of this curve.

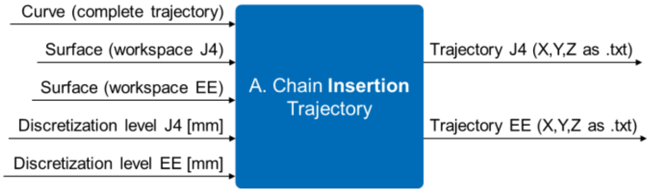

Curve (complete trajectory)
Surface (workspace J4)
Surface (workspace EE)
Discretization level J4 [mm]
Discretization level EE [mm]
A. Chain Insertion Trajectory
Trajectory J4 (X,Y,Z as .txt)
Trajectory EE (X,Y,Z as .txt)

Fig. 9: Inputs and outputs of the algorithm for the generation of insertion trajectory of kinematic chain

The algorithm basically divides the entire trajectory with the use of the surfaces that define the boundaries of the workspace for the fourth joint and the final effector. These surfaces were previously generated and are saved as global variables in the Grasshopper application. The last entries required in this case are the desired discretization levels for each segment of the trajectory. All this information is processed and two text files containing each trajectory (fourth joint and effector) are created. It is important to note that both trajectories have the same number of points. The inverse kinematic solver will have a pair of points (one from each trajectory) to find the joint locations values. Note that output text files don't contain orientation information (such as vectors or Euler angles). The tangent vector of the curve is calculated in Scilab and the solver will align the X axis of the robot to this vector. Fig. 10 shows an example of the use of this type of trajectory.

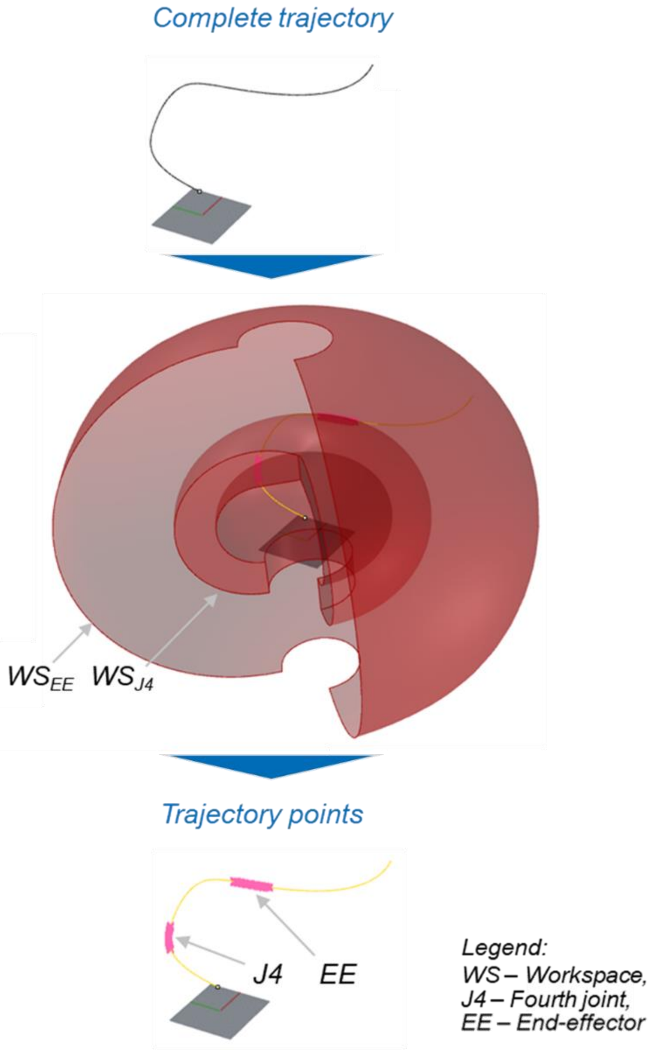


Fig. 10: Example of the use of the chain insertion trajectory

4.2.1.2 Task Execution Trajectory – General

General trajectories are the most flexible type of programming that can be applied to the execution of a task in this feasibility study. In this case, as in all paths of the task, it is assumed that the snake robot is already positioned with its effector near the area where the fence task will be conducted. The first three joints are kept fixed (i.e., the center of the fourth joint does not move in space) and the remaining five joints are responsible for positioning and orientation of the effector center and the X-axis, respectively.

For this type of trajectory, the algorithm takes an entire path given as input by the user, as well as its level of discretization and returns a list of segmentation points and tangent vectors (V_X), with respect to which the orientation of the final effector will be solved. As shown in Fig. 11, the last input required for the definition of the remaining vector (V_Y and V_Z) is the reference rotation angle (α).

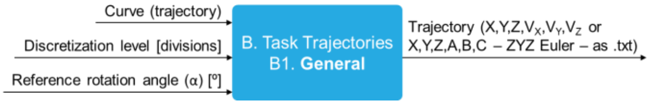

Curve (trajectory)
Discretization level [divisions]
Reference rotation angle (α) [º]
B. Task Trajectories
B1. General
Trajectory (X,Y,Z,V$_X$,V$_Y$,V$_Z$ or
X,Y,Z,A,B,C – ZYZ Euler – as .txt)

Fig. 11: Inputs and outputs of the algorithm for generating a history of general tasks.

The reference rotation angle is used to control the angular position of the orthogonal pair V_Y-V_Z in the normal plane at each segmentation point. This is the last parameter needed to fully define the trajectory in terms of position and orientation. The cross product V_X X $\hat{u}_Z$, where $\hat{u}_Z$ represents the overall Z direction, generates an auxiliary vector (V_{Z_AUX}) which is normal for V_X. This result is then rotated around V_X from an angle α to create V_Z. Finally, the orientation vector of the last trajectory is given by $V_Y = V_Z$ X V_X. The influence of the reference rotation angle can be seen in Fig. 12.

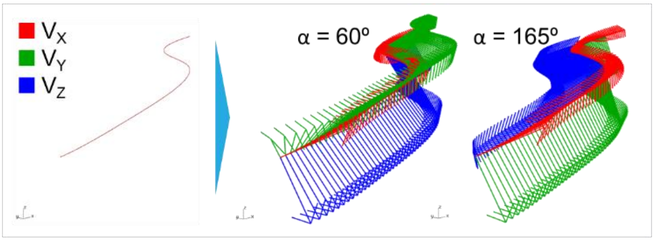


Fig. 12: Example of using the trajectory of general tasks and influence of the angle of rotation

4.2.1.3 Task execution path – Fillet sealing

This type of path is used to facilitate the scheduling of tasks in which the Effector of the robot should follow a corner (for example, fillet from the union of two aluminum plates) with a given orientation to the homogeneous application of the sealant. Unlike the general trajectory, there is no need to use the global Z direction to generate of an auxiliary vector. Once the plane curve representing the bottom of the fillet and its discretization level are given, the orientation of the final effector is totally defined by three other entries (see Fig. 13).

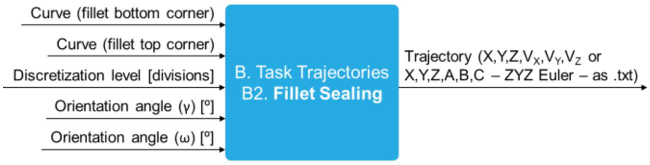

Curve (fillet bottom corner)
Curve (fillet top corner)
Discretization level [divisions]
Orientation angle (γ) [º]
Orientation angle (ω) [º]
B. Task Trajectories
B2. Fillet Sealing
Trajectory (X,Y,Z,V$_X$,V$_Y$,V$_Z$ or
X,Y,Z,A,B,C – ZYZ Euler – as .txt)

Fig. 13: Inputs and outputs of the algorithm for generation of fillet in sealing task trajectories.

First, the upper fillet curve, which consists of the lower curve folded in an orthogonal direction, guides the definition of local coordinate systems for each segmentation point. The tangent vectors of the lower curve, together with the vectors that join the segmentation points of each inserted curve, form the basis for these coordinate systems. Finally, the two orientation angles in relation to the previously formed local coordinate systems, namely the y-lead angle and the ω-tilt angle, define the inclination of the sealing applicator to the surface of the fillet. Fig. 14 shows the influence of each of these angles on the effector orientation along the programmed path.

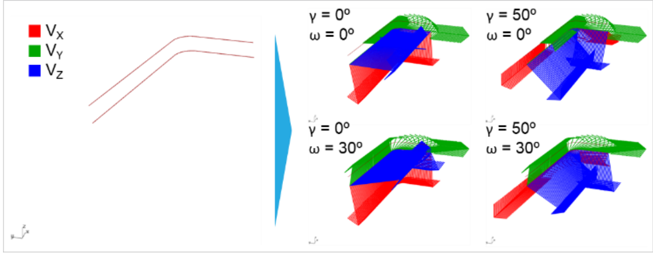


Fig. 14: Example of using of sealing trajectory of fillets and influence of rotation angle of reference

4.2.1.4 Task execution path – Rivet coating – Helical

Although this was not the main focus of this feasibility study, the SENAI Institute for Innovation in Manufacturing and Laser Processing Systems studied, as requested by Saab, to what extent the snake robot would also be able to conduct the auxiliary tools (such as sealant covers). The inputs required to generate the overlap helical rivet trajectory, are listed in *Fig. 15*.

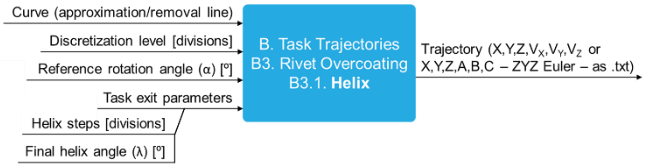

Curve (approximation/removal line)
Discretization level [divisions]
Reference rotation angle (α) [º]
Task exit parameters
Helix steps [divisions]
Final helix angle (λ) [º]
B. Task Trajectories
B3. Rivet Overcoating
B3.1. Helix
Trajectory (X,Y,Z,V$_X$,V$_Y$,V$_Z$ or
X,Y,Z,A,B,C – ZYZ Euler – as .txt)

Fig. 15: Inputs and outputs of the algorithm for generating coating for a helical trajectory of rivets

In this type of task, the effector must approach the rivet location orthogonally in a linear trajectory along which its orientation is kept constant. The trajectory of exit / removal is also a line with the same direction and level of discretization as the approach trajectory. The main difference here is the helical behavior of the final effector orientation along the tool removal. The propeller pitch parameters and the final helix angle (λ) determine how the vectors V_Y and V_Z should rotate to the application flow of the sealant to be cut and the surface finish within the acceptable sealing patterns. The final angle of the propeller provides the angular displacement between the set of vectors V_Y-V_Z for the first segment of the approach path and for the last segmentation point of the trajectory of removal. The steps of the propeller, on the other hand, define the extent to which reorientation / transition regime should last (from the beginning of the trajectory of removal). Fig. 16 shows the influence of each of these parameters on final effector orientation throughout the propeller overcoating task.

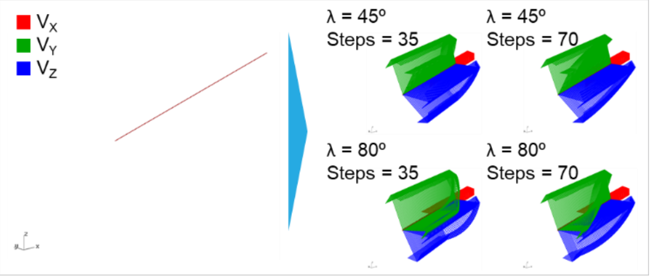


Fig. 16: Example of the use of helical trajectory for rivet and the influence of the parameters of the task's output

It is important to note that, as in the general path of the task, the general Z direction and a reference rotation angle (α) are necessary to define the approach orientation of the effector in the helical rivet liner. In addition, in this case of trajectory, 6-DOF are necessary to solve the inverse kinematics, since each parameter of orientation of the final effector must be combined so that the trajectory can be totally followed. The tests were performed with the freezing of the first two joints of the robot, but the stability of the solution found by the optimization algorithm still needs to be improved. If necessary and approved by all partners, the scope of a future project will meet this challenge.

4.2.1.5 Task execution path – Rivet coating – Cone

This is the latest trajectory concept designed for the feasibility study and provides a way for the application of sealant around a rivet, driven without any additional tools, but with the extruder's own nozzle. Similar to the concept of fillet-sealing trajectory, the trajectory of the rivet cone also needs two base curves (bottom edge and top row) for the creation of auxiliary coordinate systems, in relation to which the orientation angles of the sealing application process will be configured. The main input parameters that differ for these types of trajectory are the task output parameters present in the conical edge overlap (Fig. 17).

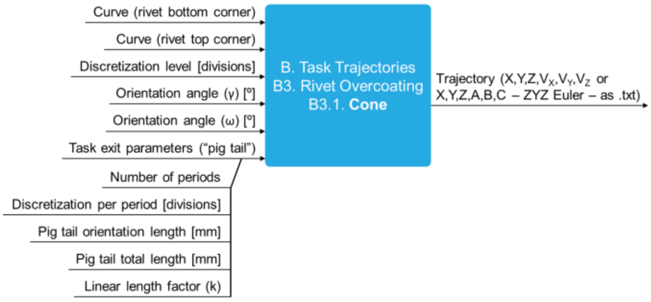

Curve (rivet bottom corner)
Curve (rivet top corner)
Discretization level [divisions]
Orientation angle (γ) [º]
Orientation angle (ω) [º]
Task exit parameters ("pig tail")
Number of periods
Discretization per period [divisions]
Pig tail orientation length [mm]
Pig tail total length [mm]
Linear length factor (k)
B. Task Trajectories
B3. Rivet Overcoating
B3.1. Cone
Trajectory (X,Y,Z,V_X,V_Y,V_Z or
X,Y,Z,A,B,C – ZYZ Euler – as .txt)

Fig. 17: Inputs and outputs of the algorithm for trajectory generation facing cone in rivets

An exit path in the form of a "pigtail" was defined at project meetings as ideal for cutting the flow of sealant in this type of application. Thereafter, after the sealant feed is interrupted, the "pigtail" trajectory takes care of the remaining sealant volume in the extruder and accommodates it so that the finish of the rivet seal is not compromised. There are a large number of parameters used for the complete definition of this output trajectory, but the influence of the main parameters can be seen in Fig. 18. The number of periods determines how many complete revolutions the effector center will be triggered until it reaches a linear output trajectory. The total length of the "pigtail" defines how many millimeters of trajectory will be required for the number of defined periods to complete. In other words, it has influence on the "density" of the conical spiral. Finally, the linear length factor (k) indirectly defines how long the last (linear) part of the trajectory should be. The higher this factor, the smaller the linear part of the output path. If k = 1, the linear part of the output trajectory will have exactly the same length as the "pigtail". If k = 2, the length of the linear part will be reduced by half.

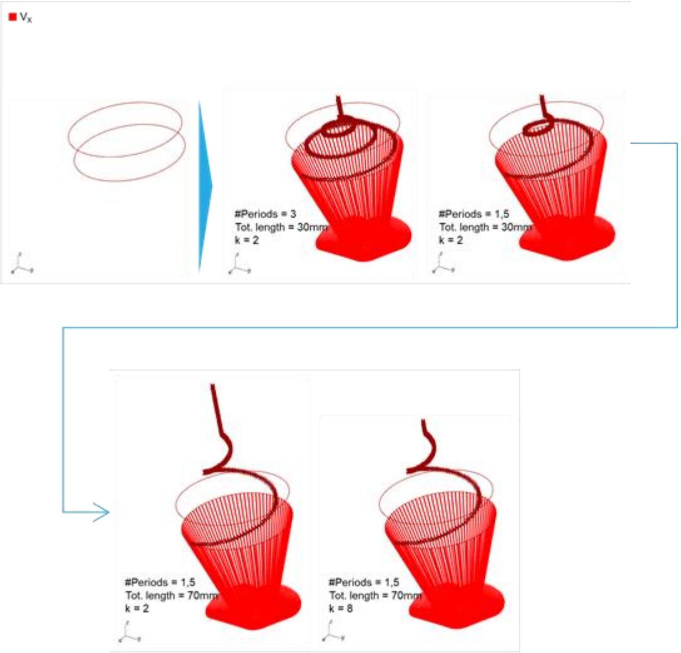


Fig. 18: Example of using cone trajectory for rivet and the influence of the parameters of the task's output.

4.3 Dimensioning and CAD model of the snake robot

For the snake robot design, some operational specifications are necessary. These specifications were basically taken from the process requirements defined by Saab (Section 3).

These specifications are used to dimensioning actuation drives for robot joints. Thus, a selection of some commercial components can be performed to compose this study. For gearboxes of main torque, the KHK Gears and Harmonic Drive® Gearboxes suppliers have been selected.

In the same way, from efforts to drive de gearboxes, the motors and control drivers of the joints of the snake robot have been defined. Available components from the Maxon Motor manufacturer have been selected.

The modeling of the snake robot was performed in the SolidWorks 2018 graphic environment. For this feasibility study, the dimensions related to the bearings and other structural components of the robot were not considered, since it was expected to evaluate its accessibility capacity and not its structural resistance. Fig. 19 show the CAD model of the proposed snake robot. A folder packed with the robot CAD file will be attached to this report.

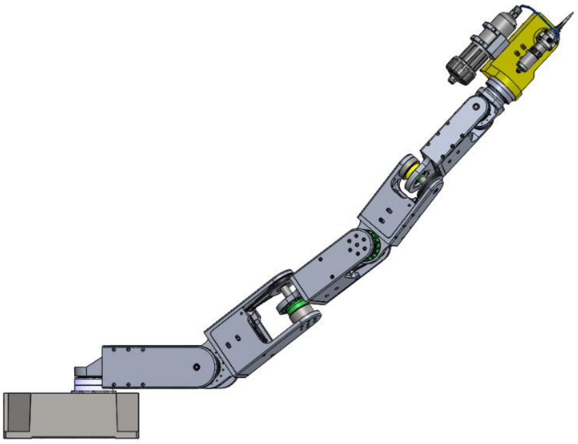


Fig. 19: Side view of the snake robot

4.4 Vision system

The proposed vision system for the project will serve to determine a local coordinate system in the sealing task workspace. The proposal is to use a measuring head with two integrated measuring systems: a LASER profilometer with three light sheets for 3D measurements and a 2D shape measuring camera on a surface.

The principle of measurement of the profilometer for 3D measurements is that of triangulation, with the measurement direction aligned with the axes of the LASERs and the camera positioned obliquely observing the line projected by the LASERs. The image shift of the line along the image at 2D sensor array lines determines the Z coordinate of the image on the LASERs axis.

The LASER profilometer with three light sheets, serves to accurately measure the normal vector of the surface to be machined. This could be done by using a LASER light sheet profilometer integrated into the robot movement, but which would carry the errors of the robot's movement. Like three sheets of light, the normal vector is measured with the robot in static pose, in a stable condition, and measurement is made only with recourse in the measurement head. Another function is to locate the robot at reference poles in the workspace and use local coordinates to minimize the errors of displacement of the kinematic chain.

Because of the complexity of the geometries found in the airframe, the measurement of the triple profilometer would require complex algorithms to identify the lines projected by each profilometer in the same image. To solve this problem, the LASERs are pulsed alternately, and the monochrome 3D camera captures images synchronized with each of them.

The camera that makes the 3D measurement is monochromatic and in its field of view there is a bandpass interference filter tuned to the LASER wavelength of the projectors. This feature, in addition the adjustment of the numerical aperture of the lens, set to the minimum value for the required depth of focus, causes the image captured by the camera to record only the line projected by the LASER without the sketch texture in the image, the 3D geometry processing is also analytical, with basic image processing. Due to the possibility of addressing the LASER, it is possible to activate only one LASER to record the shape of the sealant deposited on the joints and on the fasteners (rivets and screws).

In the head there is even a second camera, colored, unfiltered passing through its field of vision. There is also a set of white LED illuminators to produce a precise episcopic image with appropriate contrast. This image is used to locate reference points for the task on the surface to be machined. Fig. 20 shows in detail a measuring head.

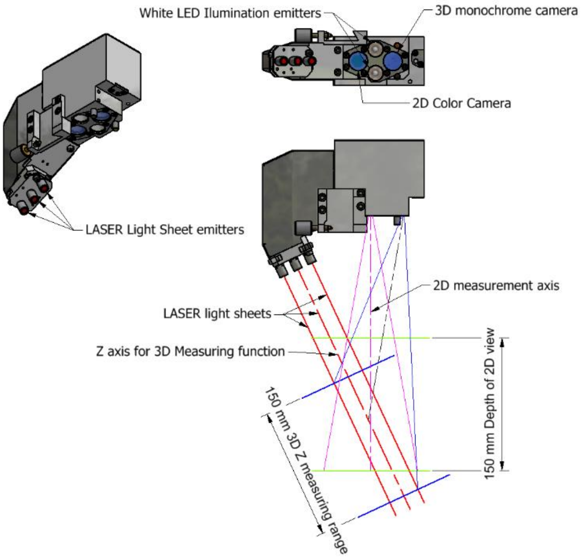
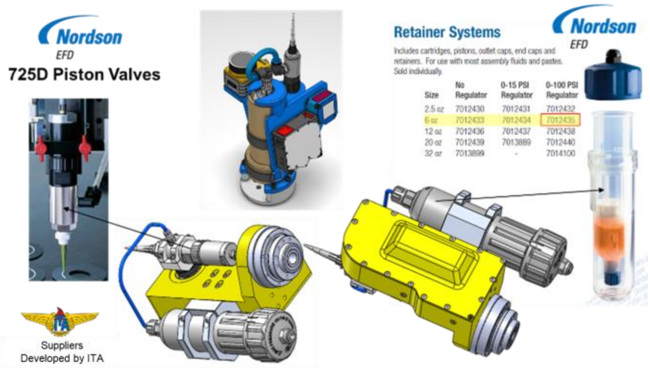

White LED Ilumination emitters
3D monochrome camera
2D Color Camera
LASER Light Sheet emitters
2D measurement axis
LASER light sheets
Z axis for 3D Measuring function
150 mm 3D Z measuring range
150 mm Depth of 2D view

Fig. 20: LASER head for 3D / 2D measurements

4.5 Study of weight reduction and dimensions of the sealant dispenser 2K

A study for the development of a sealant dispenser for airframe fuselage has already been developed by the project partner Aeronautics Institute of Technology - ITA, where size and mass reductions have made on this development, resulting in a complete new design of this dispenser has proposed, where the coupling of the reservoir cartridge and the seal dispensing valve to the robot structure was chosen. For this purpose, the same cartridge and valve used in the dispenser developed by ITA were used in this proposal (Fig. 21), but now the dispenser is integrated on link 7 and valve is on the flange after the Joint 8.



Includes cartridges, pistons, outlet caps, end caps and retainers. For use with most assembly fluids and pastes. Sold individually.

Size	No Regulator	0-15 PSI Regulator	0-100 PSI Regulator
2.5 oz	7012430	7012431	7012432
6 oz	7012433	7012434	7012435
12 oz	7012436	7012437	7012438
20 oz	7012439	7013889	7012440
32 oz	7013899	-	7014100

Fig. 21: Snake robot process effector

Thus, an even greater mass reduction was achieved for the sealant dispenser, reaching 2.7 kg considering the transmission components of joints 7 and 8, however, without considering the vision system in the dispenser developed by the ITA.

5 Conclusions

The present project, whose objective was to carry out a technical feasibility study on the use of a snake robot for the application of sealant in aircraft fuselage, had its scope and schedule fully executed. The execution of the project brought knowledge related to the design and development of robots, collaborative robotics, besides experiments related to aircraft components and application of sealant in fuselage.

In terms of the technical feasibility of the project, the solution proposed here, proved to be an excellent solution for the proposed operation. The trajectories of application of sealant in fillets (union of two plates of the fuselage) and fasteners were tested, being all very well executed by the robot in a virtual environment, without collision and without many complications. All the trajectories used for the evaluation were taken from the test framework provided by Saab, and respecting the process specifications described for this task. For the economic feasibility, although the estimated cost of the snake robot is higher when compared to the IIWA and the UR10e, the project is still feasible, since for the snake robot, after the complete development of the project, its cost of will be reduced.

A series of commercial components for the manufacture of the snake robot specified, from components for the manufacture and construction of the robot, such as: motors, reducers and vision system, to components for programming and control like drivers, software and robot interface with the user. It is important to point out that, despite the design of the robot drives, the CAD model developed here is not suitable for fabrication, structural analysis must be carried out in the structure and, after these changes have been made, evaluations and corrections are necessary robot drives must be performed. A study of the reduction of dimensions and mass in the sealant dispenser was carried out, where the coupling of the cartridge and sealant application valve was chosen directly to the robotic arm, causing an initial mass reduction of the applicator of 1.1 kg.

References

[1] L-W. TSAI, Robot Analysis: The Mechanics of Serial and Parallel Manipulators. Institute for Systems Research University of Maryland. Joh Wiley & Sons, Inc, 1999. ISBN 0-471-32593-7;

[2] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics: Modeling, Planning and Control. Springer, 2009. ISBN 978-1-84628-641-4 / e-ISBN 978-1-84628-642-1

[3] Eberchart, R. and Kennedy, J. (1995). Particle swarm optimization. IEEE International Conference on Neural Networks.

[4] Perez, R. E. and Behdinan, K. (2006). Particle swarm approach for structural design optimization. Computers and Structures 19-20: 1579-1588.