24th COBEM - 2017

24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-0994
DEVELOPMENT OF A NOVEL PERPENDICULARITY CORRECTION METHOD FOR ROBOTIC MANIPULATORS APPLIED TO AEROSPACE DOMAIN

Wesley R. de Oliveira[1]
Thiago Neves[2]
Guilherme B. Rodamillans[3]
Emília Villani[4]
Luís G. Trabasso[5]
ITA – Aeronautics Institute of Technology
Division of Mechanical Engineering (IEM) – Center of Competence in Manufacturing (CCM)
Praça Marechal Eduardo Gomes, 50 - Vila das Acácias, São José dos Campos, Brazil.
e-mails: wesleyro@ita.br[1]
thiago.nc2@ccm-ita.org.br[2]
guilhermebr@ccm-ita.org.br[3]
evillani@ita.br[4]
gonzaga@ita.br[5]

Abstract. *The aerospace industry has increased the accuracy requirements of the robotic manipulators systems from the traditional 0.2~0.5 mm, which have been usually enough to the automotive sector, to the challenging 0.05 mm (or better). Because of that, the technological research for new automation technologies based primordially on commercial off-the-shelf (COTS) elements has been considered an important way to reach higher accuracy of ordinary industrial manipulators. In this sense, the design of end-effectors that perform multiple operations allowing the improvement of the robot accuracy by means of perpendicularity measurement and correction modules retains the context of application of this paper. This work presents the development of a perpendicularity measurement module to be used in industrial robot end-effectors that require online angular corrections of the manipulator in relation to the product, as in the case of automated wing/fuselage assembly in the aeronautic industry. Besides the review of previous approaches, a novel approach for the measurement and correction based on distance laser sensors is detailed. The robustness of the presented solution is evinced by an analysis of the error propagation to define the threshold of the system and by the results of a real-time tracking application of a reference plane moved by another industrial robot.*

Keywords: *perpendicularity correction, laser sensors, manufacturing automation, robotics, aerospace industry.*

1. INTRODUCTION

Besides the traditional dominance of the automotive sector (Holland, 1999), the aerospace industry represents a new important stakeholder for the market of industrial manipulators. Its applications require new improvements on the robotic system as a whole by means of new automation technologies, which have been studied by some important companies like Airbus (Kleebaur, 2006), Boeing (Devlieg & Feikert, 2008) and Embraer (Rangel, 2010). Possible applications for industrial manipulators currently considered by this industry are wing, fuselage and subparts assembling (Eguti & Trabasso, 2014) (Mosqueira et. al., 2012). Usually, such processes work with geometrical and assembling tolerances that lead to robot positional accuracy requirements of 0.05 mm or better (Furtado et. al., 2009).

In order to reach this tight requirement on the robotic system and provide all kind of functions required by an aerospace manufacturing process, the design and integration of end-effectors that perform multiple operations have been the most assertive choice. Furthermore, although the most adopted approach to integrate such special end-effectors to the robotized process is the correction of the robot position/orientation by means of large volume metrological systems (e.g. laser tracker, photogrammetry, GPS indoor (Villani et. al., 2010)), there is still a need for integrated measurement modules to work within confined spaces and small areas. This is named herein the *local* correction performed by the robotic system upon some kind of interaction between the end-effector and the product.

A special end-effector function for the local correction emphasized in this work is the normalization of the robot in relation to the product, which is extremely important to perform other specific functions like drilling, fastener insertion and rivet squeezing. To reach this kind of local correction, some approaches are based on vision systems, which are usually good in position correction but either lack accuracy or are harder to calibrate when applied to local orientation correction (Furtado et. al., 2009). Cibiel and Prat (2006) and Devlieg and Feikert (2008) developed end-effectors with different perpendicularity measurement modules while studying the drilling process, yielding to a conclusion that the robot should be driven to touch the aircraft part. This touch procedure, called *clamp* in the technical literature, is a force-controlled operation that adapts and measures the normality angular deviation on an aeronautical surface. However, is worth to mention there are some confined situations where the clamp approach is not always feasible or should be avoided to prevent superficial damages on the product.

Previous works developed at the Center of Competence in Manufacturing (CCM), at the Aeronautics Institute of Technology (ITA), also addressed the normalization function issue. Furtado et. al. (2009) presented a system to measure small deviations of orientation of the robot in an edge to certify that the perpendicularity is correct to be used in an end-effector developed to automate a fuselage assembly process (Furtado et. al., 2014) (Eguti et. al., 2012). Their system was based on a clamp procedure by means of a mechanical patella with inductive sensors. They also calculated the accuracy requirement on the sensors considering the required accuracy of the system and validated the uncertainty of their module reaching 0.046º up to 0.180º (Furtado et. al., 2010).

Furtado et. al. (2012) compared two approaches for the integration of their correction method with the industrial robot: an *absolut* and a *null/incremental* algorithm. In the *absolut* method, the angular deviation is calculated once in Euler Angles (in the perpendicularity module coordinate system) so that the robot performs it at once. After that, the angular deviation is checked again and, if it does not reach the predefined accuracy, the process is repeated until then. In the *null* approach, the correction is based on a series of repetitive steps wherein the angular deviation is calculated and a predefined incremental/decremental angular step is sent to the robot. Both approaches can be thought as open loop procedures, since they do not consider the most updated error between the current TCP Euler angles (θ_{Ry} and θ_{Rx}) and the target orientation (and the coupling between them) while sending the target step correction to the robotic system. Despite the fact their integration was based on an OPC network, which is an inherently slow communication protocol, the convergence of their approach, which took 3.5 s to 18 s, would be also influenced by that.

Considering the aforementioned background, this paper presents the development of a compact perpendicularity measurement module to be used in industrial robot end-effectors that is based on distance laser sensors and thus performs the local normalization of the end-effector in relation to the product without prior contact. The models for the angular deviation measurements and the correction algorithm developed at LabView® are presented. Since the system is based on a real-time architecture provided by the proprietary software KUKA® Robot Sensor Interface, the results of the current proposal are presented for both common static correction and tracking applications.

2. PERPENDICULARITY CORRECTION SYSTEM

The development of the perpendicularity method described in this paper is motivated by the AME-ASA Project (*Automação da Montagem Estrutural de Asas de Aeronaves– Automation of the Structural Assembly of Aircraft Wings*), running in partnership between ITA and the Brazilian aeronautic industry with the support of FINEP (a Brazilian governmental agency). This project is aimed at developing automation technologies for a robotized wing assembly process, from which can be highlighted a robot end-effector called MINT (*Montagem de Estruturas Internas* – Assembly of Internal Parts).

The MINT end-effector was designed to drill, insert and squeeze a rivet between structural parts inside the confined spaces of a wing. Considering previous approaches from Eguti & Trabasso (2014) and Furtado et. al. (2009), starting from the automated process requirements, an integrated design approach (DIP) was used to get the concept, objective tree, functional diagram and the morphologic chart until the detailed design of the end-effector and its different modules. The MINT end-effector should be able to locally normalize itself in relation to the product surface with a perpendicularity accuracy requirement of ±1,0º. Villani et. al. (2010) showed the robot can reach until ±0.05º in terms of actuation, but not in terms of measurement. This means the robot should be closed-looped with an additional measuring system to ensure the target accuracy. As in Furtado et. al. (2009), the perpendicularity measurement module performs this function in MINT.

2.1 Perpendicularity Measurement Module

Figure 1 presents the solution from Furtado et. al. (2009; 2012), which consists of a mechanical patella with linear inductive sensors that, upon a pressure on the aeronautical surface, allowed the angular deviation calculation. Figure 1 points out some important aspects of the patella solution: the inductive sensors are supposed to be in an orthogonal coordinate system arrangement so that linear distance measurements from the sensors are supposed to be directly expressed in the patella's XYZ coordinate system; and the distance measures (*b*) are supposed to be fully oriented in the Z-axis direction, which is a well-know approximation for small angle deviations so that this kind of measurement system is specified for such a condition. This work proposes a non-touching perpendicularity measurement module based on distance laser sensors as depicted in Fig. 2.

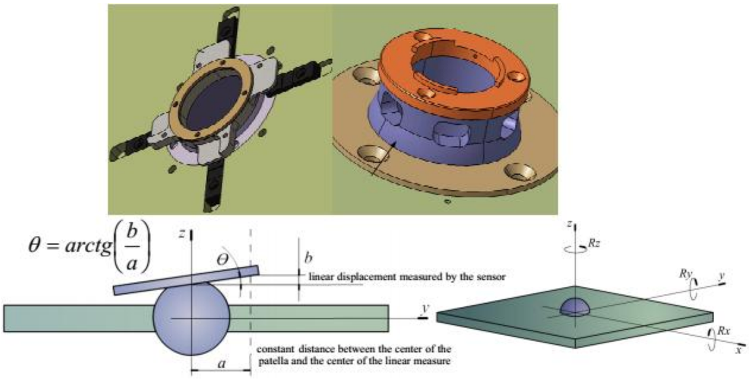

$\theta = arctg\left(\frac{b}{a}\right)$
z
θ
b
linear displacement measured by the sensor
y
constant distance between the center of the patella and the center of the linear measure
a
z
Rz
Ry
y
Rx
x

Figure 1. Mechanical patella for perpendicularity measurement. Source: Furtado et. al., 2009.

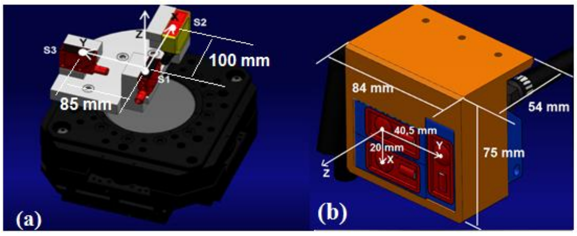


Figure 2. Prototype (a) and DFP (b) of the perpendicularity measurement module to be coupled to MINT.

Figure 2 (a) exhibits the functional prototype designed to verify the angular deviation calculation method and the correction algorithm. Figure 2 (b) exhibits the DFP (*Demonstrador Funcional Pleno - Full Functional Demonstrator*) which is a more compact version to be integrated in MINT. The sensors are fixed in an orthogonal-triangle form in order to resembling a coordinate frame that can be referred to the robot flange as well. Figure 3 shows the functional prototype installed on a robot KUKA KR210, evincing the basic principle of angular deviation calculation.

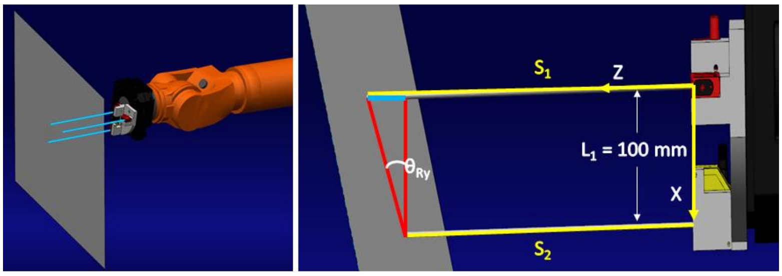


Figure 3. Basics of angular deviation calculation from the measures referred to the module coordinate system.

The most simple equation for the angular deviation calculation shall be derived as in Fig. 1.

$$\theta_{Rx} = tg^{-1}\left(\frac{s_3 - s_1}{L_2}\right) \quad (1.1a) \qquad \theta_{Ry} = tg^{-1}\left(\frac{s_2 - s_1}{L_1}\right) \quad (1.1b) \tag{1}$$

where, according to Fig. 2, L$_1$ is the distance between sensors S1 and S2 (100 mm); L$_2$ is the distance between sensors S1 and S3 (85 mm); and s_1, s_2 and s_3 are the measures from sensor S1, S2 and S3, respectively. This quite basic equation can be used to get the approximate required minimum accuracy of the sensors, as in Furtado et. al. (2009), expressed in Eq. 2 for a target accuracy 10 times better than the specification of ±1.0°, which yields to ±0.1°.

$$|\Delta s_3| + |\Delta s_1| \ \ and \ \ |\Delta s_2| + |\Delta s_1| \leq \min\left(L_1 tg\left(0{,}1^o\right), L_2 tg\left(0{,}1^o\right)\right) \tag{2}$$

Equation 2 yields to the conclusion that the absolute errors from sensors S1 and S2 (or S3 and S1) should not sum up more than: 148.4 µm for the functional prototype (Fig. 2.a) and 34.9 µm for the DFP (Fig. 2.b). Table 1 presents the specifications of the sensors used in the tests with the functional prototype and specified for the final DFP.

Table 1. Characteristics of the laser sensors used in the perpendicularity measurement modules.

Functional Prototype (Fig. 2.a)		**DFP (Fig. 2.b)**	
Model	Accuracy	Model	Accuracy
01 Banner LED-Sensor Q50AVU	3.5 mm	Wenglor CP08MHT80	10.0 µm
02 Leuze Laser ODSL 8W4	1.5 mm		

Although used just for the purpose of verifying the correction method presented in this paper, Tab. 1 evinces the sensors used in the functional prototype are far beyond the minimum required accuracy (their accuracy sum up from 3 mm up to 5 mm). However, in the practical terms of control, upon a calibration procedure the systematic/trueness portion of that was fairly extracted, so that the measured repeatability/precision lied between 0.254 mm for S1 and S3 and 0.938 mm for S2. Thus, considering the application of Eq. 2, the functional prototype should not be thought to reach an accuracy better than ±0.5º for θ_{Rx} and ±0.8º for θ_{Ry}. The final version of the module (DFP) is specified with a suitable laser sensor (Tab. 1 shows it would sum up 20 µm, therefore better than the specified 34.9 µm).

At this point, it is worth mentioning the model in Eq. 1 undergoes some drawbacks because, besides the small angle deviations condition, it assumes the orthogonal triangle arrangement, which is not feasible with an indefinite precision for a real mechanical assembly. This is precisely the idea behind the need for the calibration and validation procedure discussed in Furtado et. al. (2010). Furthermore, Eq. 2 comes from the assumption there is no error on the dimensions L_1 and L_2. From Bentley (2004), a more rigorous error analysis can be derived from Eq. 1 as in Eq. 3.

$$\Delta\theta_{Ry} = \Delta s_2 \frac{L_1}{L_1^{\;2} + (s_2 - s_1)^2} + \Delta s_1 \frac{L_1}{L_1^{\;2} + (s_2 - s_1)^2} + \Delta L_1 \frac{|s_1 - s_2|}{L_1^{\;2} + (s_2 - s_1)^2} \tag{3}$$

Equation 3 (exhibited just for $\Delta\theta_{Ry}$) shows that the accuracy on the angular deviation calculation is a function of the difference between the distance measures itself. Another important practical drawback with a correction method based on Eq. 1 is that the given angular setpoint is supposed to be implemented directly in XYZ coordinate system of the module (Fig. 1 or Fig. 2), implying a dependency of such an approach on the robot tool calibration, which could be another source of error. In the following, an alternative approach for the angular deviation calculation will be presented.

2.2 Homogenous Transform for Angular Deviation

In the following mathematical treatment, the arrangement of the sensors within the perpendicularity module is supposed to be in the same orthogonal form. As it shall be demonstrated, this would not yield to a loss of generality. Figure 4 shows a vectorial approach for the concept illustrated in Fig. 1 and Fig. 2. The following symbols are used: ${}^S\vec{v}_A$, ${}^S\vec{v}_B$ and ${}^S\vec{v}_C$ are the vectors that points to the laser projection A, B and C, respectively from the sensors S1, S2 and S3, referred to the module coordinate system XYZ. The vectors ${}^S\vec{v}_{AB}$ and ${}^S\vec{v}_{AC}$ are located on the target plane. The term *coordinate system* is shorten hereinafter by C.S.; ${}^O\overline{\overline{R_F}}$ is the rotation matrix describing the orientation of the C.S. *F* in the C.S. *O*; and ${}^O\overline{\overline{H_F}}$ is the homogeneous matrix describing the orientation and position of the C.S. *F* in *O*.

The main goal of the vectorial approach is to identify a target C.S. onto the target plane for which the Z-axis should point along the normal direction of this plane. Such a target C.S. is obtained from the vectors depicted in Fig. 4.

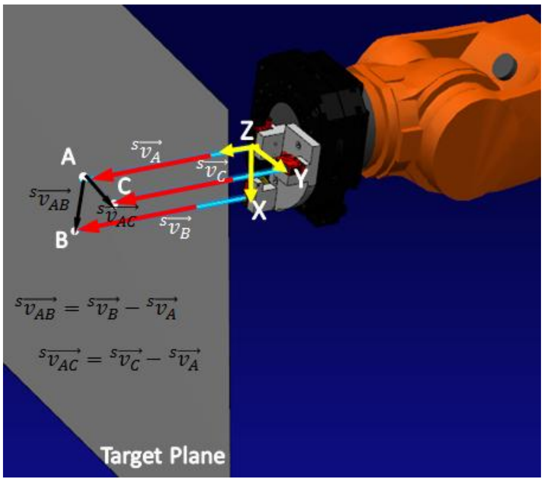

A
C
B
Z
Y
X
${}^S\vec{v}_A$
${}^S\vec{v}_C$
${}^S\vec{v}_{AB}$
${}^S\vec{v}_{AC}$
${}^S\vec{v}_B$
${}^S\vec{v}_{AB} = {}^S\vec{v}_B - {}^S\vec{v}_A$
${}^S\vec{v}_{AC} = {}^S\vec{v}_C - {}^S\vec{v}_A$
Target Plane

Figure 4. Vectorial approach for the angular deviation calculation referred to the sensor coordinate system.

To build the target coordinate system, point A is defined hereinafter as its origin. The versor of vector ${}^{S}\vec{v}_{AB}$ (${}^{S}\hat{u}_{X}$) shall be defined as its X-axis versor. The versor of the vectorial product of the vectors ${}^{S}\vec{v}_{AB}$ and ${}^{S}\vec{v}_{AC}$ shall be defined as its Z-axis versor (${}^{S}\hat{w}_{Z}$) and the remaining Y-axis versor (${}^{S}\hat{v}_{Y}$) should be determined by the right-hand rule. The set of expressions in Eq. 4 summarize the mathematical procedure to build up the target C.S.

$$
\begin{array}{l}
{}^{S}\vec{v}_{A} = \begin{bmatrix} 0 & 0 & s_1 \end{bmatrix}^{T} \quad (4.1a) \qquad {}^{S}\vec{v}_{B} = \begin{bmatrix} 0 & L_1 & s_1 \end{bmatrix}^{T} \quad (4.1b) \qquad {}^{S}\vec{v}_{C} = \begin{bmatrix} L_2 & 0 & s_2 \end{bmatrix}^{T} \quad (4.1c) \\
{}^{S}\hat{u}_{X} = \dfrac{{}^{S}\vec{v}_{AB}}{\left|{}^{S}\vec{v}_{AB}\right|} \quad (4.2a) \qquad {}^{S}\hat{w}_{Z} = \dfrac{{}^{S}\vec{v}_{AB}\times{}^{S}\vec{v}_{AC}}{\left|{}^{S}\vec{v}_{AB}\times{}^{S}\vec{v}_{AC}\right|} \quad (4.2b) \qquad {}^{S}\hat{v}_{Y} = \hat{w}_{Z}\times{}^{S}\hat{u}_{X} \quad (4.2c) \\
{}^{S}\overline{\overline{R}}_{Plane} = \begin{bmatrix} {}^{S}\hat{u}_{X} & {}^{S}\hat{v}_{Y} & {}^{S}\hat{w}_{Z} \end{bmatrix} \quad (4.3) \qquad {}^{S}\overline{\overline{H}}_{Plane} = \begin{bmatrix} {}^{S}\overline{\overline{R}}_{Plane} & {}^{S}\vec{v}_{A} \\ 0\ \ 0\ \ 0 & 1 \end{bmatrix} \quad (4.4)
\end{array}
\tag{4}
$$

Equation 4.3 shows the resultant rotation matrix (${}^{S}\overline{\overline{R}}_{Plane}$) describing the orientation of the target C.S. (on the plane) in relation to the perpendicularity module C. S. Equation 4.4 shows the resultant homogeneous matrix (${}^{S}\overline{\overline{H}}_{Plane}$) that includes also the position of the target C.S. (on the plane) in relation to the perpendicularity module C. S.. The goal is to bring the robot aligned to the plane so that ${}^{S}\overline{\overline{R}}_{Plane}$ approaches to an identity matrix without changing ${}^{S}\vec{v}_{A}$. To get this, the target C.S. should be described in the robot flange C.S. (F), by means of Eq. 5.

$$
{}^{F}\overline{\overline{H}}_{Plane} = {}^{F}\overline{\overline{H}}_{S} * {}^{S}\overline{\overline{H}}_{Plane} \tag{5}
$$

In order to accomplish this simple transformation, the position and orientation of the perpendicularity module in relation to the robot flange *F* C. S. (${}^{F}\overline{\overline{H}}_{S}$) should be determined from a CAD reference model or by means of a calibration procedure, which is a well-known optimization problem in robotics and has been reported in many sources. The possibility of calibration without the need of another metrological system is an advantage of this approach, because if the arrangement of the sensors in the perpendicularity module is not in the orthogonal form, each sensor S1, S2 and S3 could be independently described in the *F* C.S. by homogenous matrixes ${}^{F}\overline{\overline{H}}_{S1}$, ${}^{F}\overline{\overline{H}}_{S2}$ and ${}^{F}\overline{\overline{H}}_{S3}$ gathered, respectively, from a calibration. Thus, the points A, B and C could be mapped directly in the *F* C. S. (Eq. 6).

$$
\begin{array}{l}
\begin{bmatrix} {}^{F}\vec{v}_{A}{}^{T} & 1 \end{bmatrix}^{T} = {}^{F}\overline{\overline{H}}_{S1} * \begin{bmatrix} 0 & 0 & s_1 & 1 \end{bmatrix}^{T} \\
\begin{bmatrix} {}^{F}\vec{v}_{B}{}^{T} & 1 \end{bmatrix}^{T} = {}^{F}\overline{\overline{H}}_{S1} * \begin{bmatrix} 0 & 0 & s_2 & 1 \end{bmatrix}^{T} \\
\begin{bmatrix} {}^{F}\vec{v}_{C}{}^{T} & 1 \end{bmatrix}^{T} = {}^{F}\overline{\overline{H}}_{S1} * \begin{bmatrix} 0 & 0 & s_3 & 1 \end{bmatrix}^{T}
\end{array}
\tag{6}
$$

Using the vectors ${}^{F}\vec{v}_{A}, {}^{F}\vec{v}_{B}$ and ${}^{F}\vec{v}_{C}$ in Eq. 4, the matrixes ${}^{F}\overline{\overline{R}}_{Plane}$ and ${}^{F}\overline{\overline{H}}_{Plane}$ could be derived directly from the distance measures. Besides this aspect, the nearness of this vectorial approach to the robot's Cartesian command loop language favour the integration and correction method, discussed in the following.

2.3 Perpendicularity Correction Method

Figure 5 presents the integration and correction approach proposed in this work. It can be considered as a closed control loop approach, external to the internal robot control loop.

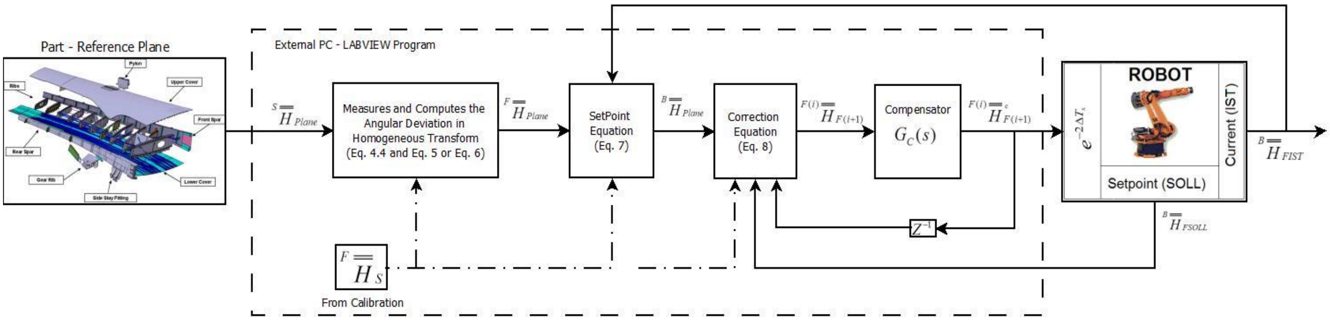

Part - Reference Plane
Pylon
Ribs
Upper Cover
Front Spar
Rear Spar
Gear Rib
Lower Cover
Side Stay Fitting
External PC - LABVIEW Program
${}^{S}\overline{\overline{H}}_{Plane}$
Measures and Computes the Angular Deviation in Homogeneous Transform (Eq. 4.4 and Eq. 5 or Eq. 6)
${}^{F}\overline{\overline{H}}_{Plane}$
SetPoint Equation (Eq. 7)
${}^{B}\overline{\overline{H}}_{Plane}$
Correction Equation (Eq. 8)
${}^{F(i)}\overline{\overline{H}}_{F(i+1)}$
Compensator
$G_C(s)$
${}^{F(i)}\overline{\overline{H}}^{c}_{F(i+1)}$
Z^{-1}
${}^{F}\overline{\overline{H}}_{S}$
From Calibration
$e^{-2\Delta T_s}$
ROBOT
Setpoint (SOLL)
Current (IST)
${}^{B}\overline{\overline{H}}_{FIST}$
${}^{B}\overline{\overline{H}}_{FSOLL}$

Figure 5. Closed control loop correction method.

Starting from the reference plane on the aeronautical part over which the robot should be normalized with, Fig. 5 shows that the control algorithm starts with the computation of the homogeneous matrix that describes the target C.S. on the surface in the robot flange coordinate system (${}^{F}\overline{\overline{H}}_{Plane}$). Then, this is fed to the setpoint equation (Eq. 7).

$$ {}^{B}\overline{\overline{H}}_{Plane} = {}^{B}\overline{\overline{H}}_{FIST}(i) \quad * \quad {}^{F}\overline{\overline{H}}_{Plane} \tag{7} $$

where: ${}^{B}\overline{\overline{H}}_{FIST}(i)$ is the current (ith step time) position/orientation of the robot flange (FIST) given in an homogeneous transform format in relation to the robot base (B) (this is usually read every 12 ms in a short Cartesian notation *X*, *Y* and *Z* for the position and in Euler *ZYX* angles *A*, *B* and *C* for the orientation from the robot controller).

At this point, is important to mention that the robot controller can feed the external client (External PC in Fig. 5) with two position/orientation feedback: (1) the last position/orientation setpoint received from the external client itself, identified by the acronym FSOLL, that is fed to the internal robot control loop with a transport delay of 24 ms (expressed by the $e^{-2\Delta Ts}$ in the robot block in Fig. 5); and (2) the current (ith step time) position/orientation of the robot flange, identified by the acronym FIST, that is gathered from the most updated robot joint encoders measures. Besides that, the robot can be optionally commanded in incremental (used in this work) or absolute values in relation to the base or the robot flange itself. In this correction approach, the setpoint shall be determined from the FIST data and the correction from the FSOLL data. The setpoint from Eq.7 is fed to the main correction equation (Eq. 8).

$$ {}^{F(i)}\overline{\overline{H}}_{F(i+1)}(i) = {}^{F}\overline{\overline{H}}_{S} * \left[{}^{B}\overline{\overline{H}}_{FSOLL}(i) * {}^{F(i-1)}\overline{\overline{H}}_{F(i)}(i-1) * {}^{F}\overline{\overline{H}}_{S} \right]^{-1} * {}^{B}\overline{\overline{H}}_{Plane} \tag{8} $$

Where: ${}^{F(i)}\overline{\overline{H}}_{F(i+1)}(i)$ is the incremental correction calculated in the current ith time step to be performed in the next time i+1th time step in relation to the robot flange. Equation 8 also states that the calculated incremental correction in each time step makes up for the last incremental command sent to the robot, to account for the internal transport delay of the robot. Although showed for the general homogenous transform, the algorithm is not supposed to implement a position modification so that just the rotation matrix part should change every time step. This notation however evinces the angular deviation calculations are in fact performed in the perpendicularity module coordinate system, but implemented by the robot in a conveniently chosen coordinate system: the robot flange coordinate system.

Finally, Fig. 5 also shows a compensator block ($G_C(s)$) which can be implemented over the correction yielded by Eq. 8 to reach a specific desired dynamic behavior of the whole system. Its structure is therefore dependent on the application and on the limits of actuation of the robot. Since it is not the focus of this work, it is adjust as a unit gain controller (no change on dymanics) with some saturation that preserves the operational increment limits of the robot.

3. EXPERIMENTAL APPROACH

This section describes the resources and procedures employed in the verification of the perpendicularity correction method presented in this work. The sensors used in the functional prototype are described in section 2.1. The integration was based on the KUKA proprietary application Robot Sensor Interface (RSI), which allows the implementation of a real-time correction system with a heartbeat of 12 ms.

3.1 Experimental Setup and Pocedure

Figure 6 shows the experimental test-bench used for verification of the proposed perpendicularity correction method. The functional perpendicularity prototype can be seen installed on the KUKA KR210-L180 robot (right side).

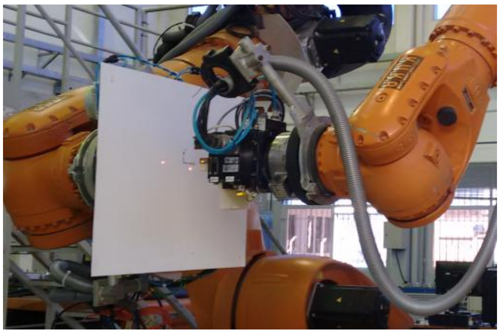


Figure 6. Experimental test-bench based on the robots KR500-L480 and KR210-L180.

Figure 6 shows that the experimental test-bench was based additionally on a second robot KUKA KR500-L480 (left side), which was used as a driver to introduce a regular and known angular deviation on a target plane, which is the object over which the KUKA KR210-L180 should be normalized with by means of the functional perpendicularity module prototype. Since the KR500- L480 robotic system is able reach until ±0.05º in terms of orientation actuation (Villani et. al., 2010), this is a fairly procedure to perform the verification of the perpendicularity correction method, which, as discussed in section 2.1, shouldn't provide a precision better than ±1.0º with the sensors selected initially.

However, to reasonably compare the input versus the output data, the object/tool calibration is performed on the KR500- L480 robotic system and a common external coordinate system (the *BASE* system) is defined for the measurement of the target plane orientation on both robotic systems KR500-L480 and KR210-L180. To do this, pre-calibration procedures of the tool center point (TCP) and of the external *BASE* are performed on both robots using respectively the XYZ 4-Point and the 3-Point methods (KUKA, 2002).

Figure 7 exhibits the Cartesian Euler convention for the representation of orientation used by the KUKA robot systems, being represented with sequential rotations around the Z-axis, Y-axis and X-axis by the A, B and C angles.

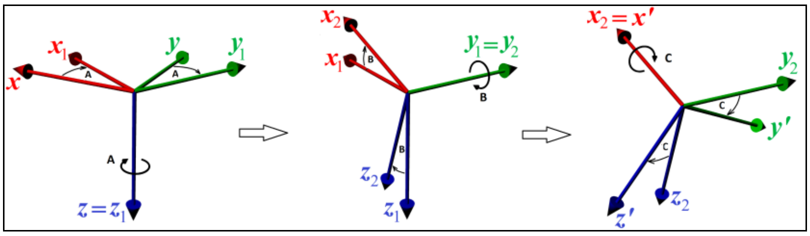


Figure 7. Cartesian euler angles convention (ABC represents a sequential rotation around ZYX).

As depicted previously, the TCP of the perpendicularity module *C.S.* is adjusted so that the z-axis points out from the module directed to the surface over which it should be normalized with. On the KUKA KR500-L480 the TCP is adjusted in the same but with the Z-axis pointing into the robot flange so that the Z-axis on the target plane gets aligned to the Z-axis of the perpendicularity module's TCP when they are parallel to each other (situation depicted in Fig. 6).

In order to evaluate the performance of the system on different operating points, the temporal angular deviation pattern depicted in Fig. 8 was applied to the KUKA KR500-L480. Figure 8 shows the angular orientation inputs referred to the *origin*, which stands for the initial target plane orientation depicted in Fig. 6. Every 5 s angles B and C were changed in steps of 1° in the upward and downward direction, each of them individually and simultaneously. In this manner, the target plane gets until 5° deviated from the original orientation around the two possible directions. These patterns were tried with angle A set in 0° (no rotation around Z), -15° and 15°. Holding each step along 5s allowed the evaluation of the static error and convergence of the method. Besides that, the step transitions evinced the tracking characteristics of the system. The tolerance used in the experiment was 0.5° for θ_{Rx} and 1.5° for θ_{Ry}, which can be also thought as a dead zone for the system's correction capability.

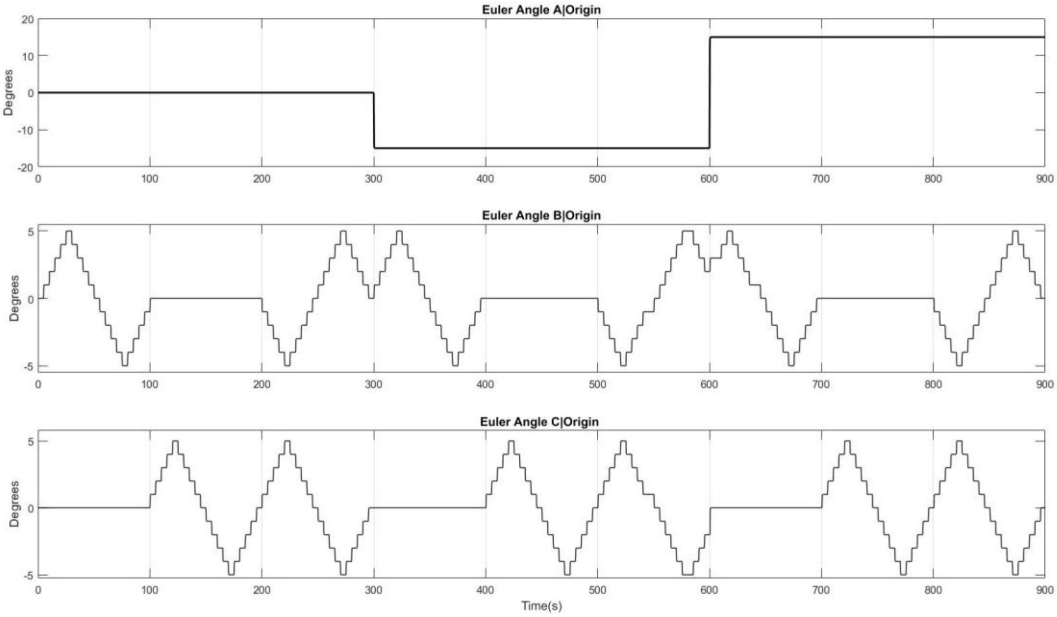


Figure 8. Angular pattern inputted on the target plane by means of the KUKA KR500-L480.

4. RESULTS AND DISCUSSION

Figure 9 presents the angular deviation in ABC Euler angles extracted from the resultant homogeneous matrix determined by means of Eq. 4 of the perpendicularity correction method of this paper.

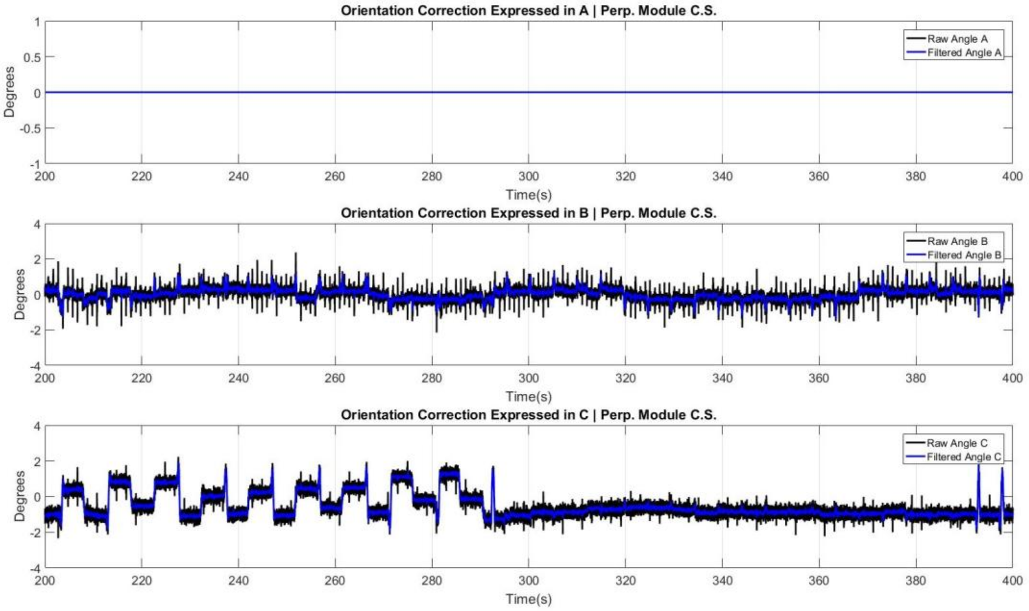


Figure 9. Experimental Data: Angular deviation calculated in relation to the TCP of the perpendicularity module.

Figure 10 presents the orientation (measured in ABC Euler angles) of the TCP located on the target plane (as stated in section 3.1) and the orientation of the perpendicularity module TCP, both measured in relation to the common external *BASE* C.S.. These signals were measured every 12 ms.

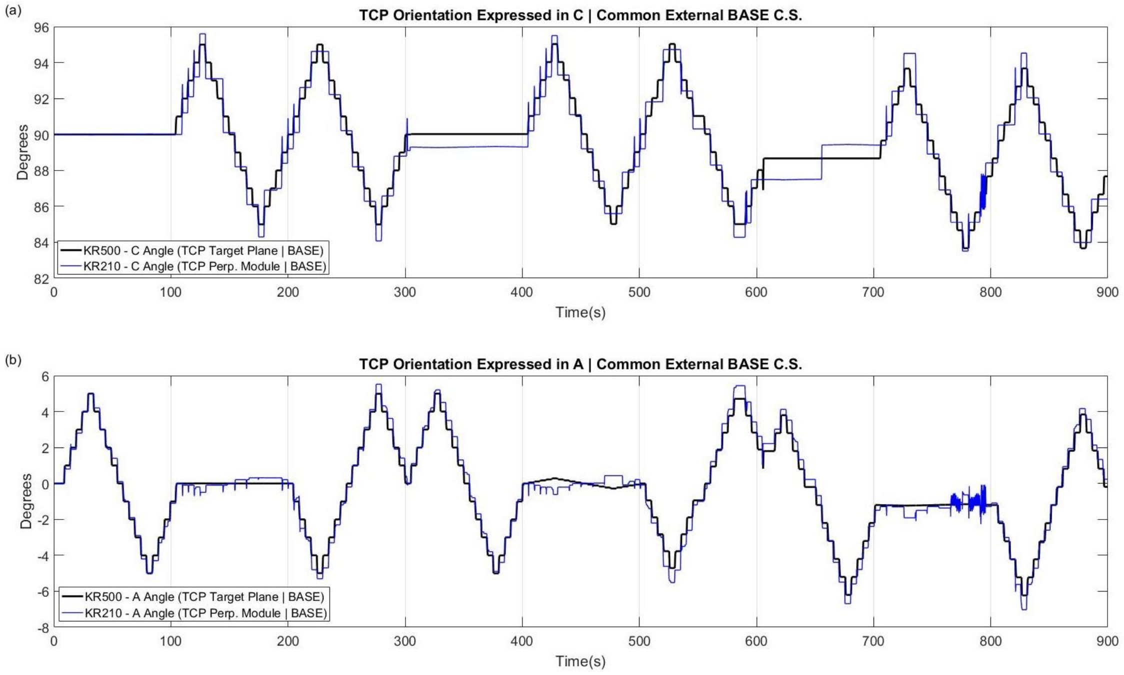
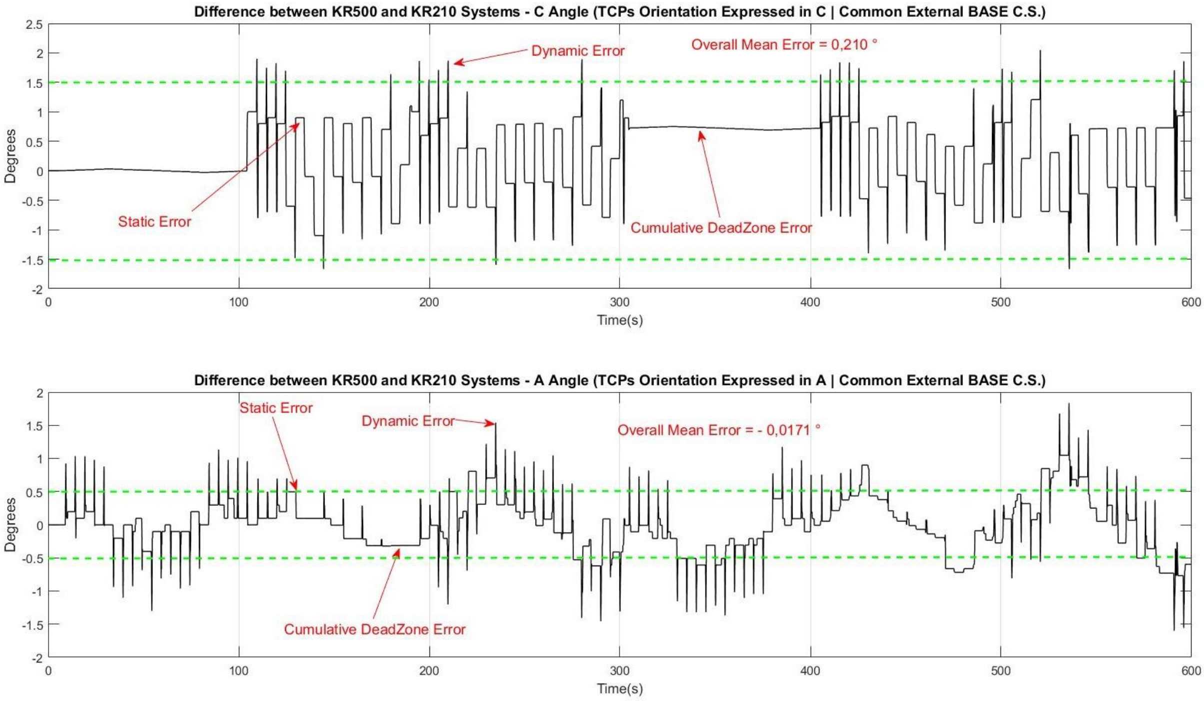


Figure 10. Experimental Data: (a) inputted angular deviation of the target plane (measured in the KUKA KR500-L480 system in relation to the BASE C.S.) and (b) response of the perpendicularity correction system (measured in the KUKA KR500-L480 system in relation to the BASE C.S.).

Figure 9 presents two versions of the signals: the angular deviation calculated directly from the raw distance sensors data (black lines); and the same angular deviation signals determined from a filtered data set by a Kalman Filter estimator (blue lines). Although out of the scope of this paper, the filtering was performed due to the poor quality of the sensors used. As discussed in section 2.1, from Fig. 9 becomes clear that the variance of the angular deviation gathered from the raw distance measures reaches 1.2º with the system in steady condition. Otherwise, the filtering yielded to a variance up to 0.20°. So, the filtered version shall be used in the next, but still keeping the same tolerance zone of 0.5° for θ_{Rx} and 1.5° for θ_{Ry} for the closed loop system.

Figure 10 shows that, within the presented real-time architecture, the system was able to converge in each step with a rise time of 280 ms. In order to have a detailed view on that, Fig. 11 shows a zoomed view, in the time interval [0, 600] s, of the difference between the curves in Fig. 10.


Difference between KR500 and KR210 Systems - C Angle (TCPs Orientation Expressed in C | Common External BASE C.S.)

Difference between KR500 and KR210 Systems - A Angle (TCPs Orientation Expressed in A | Common External BASE C.S.)

Figure 11. Difference between the inputted reference orientation and the final system orientation.

Figure 11 evinces two kinds of difference/error between the two orientation signals (the reference inputted on the target plane by KR500-L480 and the response measured from KR210-L180): a static error, linked to the static condition of each steady step value; and a dynamic error, linked to the step transitions. The former is due to the tolerance (dead zone) used for each orientation degree of freedom. The latter is due to the absence of a predefined controller to regulate the system behavior along each step transition.

Although there is a noticed coupling between the orientation angles when they are measured in the external *BASE* C.S., which, in a final point view, limits the direct comparison between the predefined tolerances (putted in the perpendicularity module’s TCP) and the errors, Fig. 11 indicates the static errors along the experiment are mostly lower than the tolerance zones (indicated in Fig. 11 by green-dotted lines, 0.5° for θ_{Rx} and 1.5° for θ_{Ry}). In some intervals (e.g. 400 s and 500 s) the static errors reached values beyond the tolerance zone, which is due to the coupling between ABC angles themselves and due to other sources of error (e. g. error from the *BASE* calibration in each robotic system). Figure 11 also evinces the dynamic errors reached higher values since they are linked to the difference between the systems while they are moving (therefore surpassing the tolerances values by definition).

In spite of that, the overall mean error between the signals, along the experiment, reached 0.210° for the orientation degree of freedom more influenced by the S1-S2 pair of sensors (angle C) and -0.0171° for the orientation degree of freedom more influenced by the S1-S3 pair of sensors (angle A), which can be considered, for all intents and purposes, a good result.

5. CONCLUSION

This paper presented a novel perpendicularity correction method for robotic manipulators whose importance was highlighted, though not limited, to the automation of the aerospace industry. Summarizing, the following points can be emphasized for the current proposition: a non-touching perpendicularity measurement module that performs local normalization of an end-effector in relation to the product in confined situations; a vectorial formulation for the angular deviation calculation; and a real-time closed loop integration method which yielded to a dynamic control system able to perform both quasi-static and tracking corrections.

Comparatively to previous works, this paper showed the proposed perpendicularity correction system integrated with the KR210-L180 robot was able to reach the predefined tolerance. An important aspect in this sense is that the system's accuracy can be reasonably adjusted with a suitable distance laser sensor (Eq. 2 and Eq. 3). Using the closed loop real-time architecture, the system was able to converge in 280 ms, which is 12,5 times better than the proposition of Furtado et. al. (2012). This is due mainly to the communication protocol, which was not real-time in their application.

The system presented a fast reaction, which yielded to an overshoot (in Fig. 10). This is due to the fact no predefined controller was added to change the dynamics, which was out of the scope of this paper and shall be discussed in an upcoming version of this work.

6. REFERENCES

Bentley, J. P. (2005) Principles of measurement systems.4th Edition. London. Pearson Education, England, 2005. ISBN 0 130 43028 5.

Cibiel, C.; Prat, P. (2006). “Automation for the Assembly of the Bottom Wing Panels on Stringers for the A320”, SAE International, 2006-01-3143.

Devlieg, R.; Feikert, E. (2008). One-up assembly with robots. In: SAE WORLD CONGRESS, 2008, Detroit 2008-01-2297. Proceedings of the SAE WORLD CONGRESS 2008, Washington, DC: SAE, 2008.

Doebelin, E..(2004). Measurement systems: application & design. 5. ed. Nova York: Mcgraw-Hill. 1078 p.. ISBN-10: 0072990724.

Eguti, C. C. A.; Trabasso, L. G.; Villani, E.; Coracini, G. K. & Furtado, L. F. F. (2012). Development of a robotic end-effector of drilling and fasteners inserter for aircraft structures. SAE International Technical Paper 2012-01-1858, 2012. DOI:10.4271/2012-01-1858.

Eguti, C. C. A. & Trabasso, L. G. (2014). Design of a robotic orbital driller for assembling aircraft structures. In: Journal of Mechatronics, vol. 24, Issue 5, pp. 533-545. Elsevier, 2014. DOI: 10.1016/j.mechatronics.2014.06.007.

Furtado, L. F. F.; Villani, E. & Sutério, R. (2009). A perpendicularity measurement system for industrial robots. IN: Proceedings of the 20th International Congress of Mechanical Engineering. ABCM, Gramado-Brazil, November 15-20, 2009.

Furtado, L. F. F.; Villani, E.; Sutério, R. (2010). Verificação metrológica de um sistema de medição de perpendicularidade para manipuladores robóticos industriais. In: 18th CBA - Congresso Brasileiro de Automática 2010. Proceedings of the 18th CBA - Congresso Brasileiro de Automática, Bonito, 2010.

Furtado, L. F. F.; Coracini, G. K.; Villani, E. & Trabasso, L. G. (2012). Comparative study between two methods for perpendicularity corrections in robotic manipulators. In: Proceedings of ABCM Symposium Series in Mechatronics – Section VII: Robotics, Vol. 5 pp. 1194-1200. ABCM, 2012.

Furtado, L. F. F.; Villani, E.; Trabasso, L. G. & Silva, C. E. O. (2014). DTW: a design method for designing robot end-effectors. In: Journal of the Brazilian Society of Mechanical Science and Egineering, vol. 36, pp. 871-885. DOI 10.1007/s40430-013-0109-8.

Holland S. W. & Nof S. Y. (1999). Handbook of industrial robotics: emerging trends and industry needs, 2nd ed. John Wiley & Sons, New York, 1378 p. (ISBN 0-471-17783-0).

Kleebaur, R. (2006). Where precision counts above all else. The High Flyer: European Aeronautic Defence And Space Company, Munick, n. 2, p.12-14, 2006. Newsletter For Engineering Students.

KUKA Roboter GmbH. (2002) KUKA Manual: Software KR C2 Expeet Programming – KUKA System Sofwtare 8.3. Version 00, Issued: 26 Sep 2003.

Maropoulos, P. G.; Muelaner, J. E.; Summers, M. D. & Martin, O. C. (2013) A new paradigm in Large Scale Assmebly – Research Priorities in Measurement Assisted Assembly. In: The International Journal of Advanced Manufacturing Technology, vol. 70, issue 1-4, pp. 621-633. Springer, 2013. DOI: 10.1007/s00170-013-5283-4.

Mosqueira, G.; Apetz, J.; Santos, K.M.; Villani, E.; Suterio, R.; Trabasso, L.G. (2012). Analysis of the indoor GPS system as a feedback for the robotic alignment of fuselages using laser radar measurements as comparison. In: Journal of Robotics and Computer-Integrated Manufacturing, vol. 28, pp. 700-709. Elsevier, 2012. DOI: 10.1016/j.rcim.2012.03.004.

Rangel, R. (2010). Robôs garantem precisão na montagem de aeronaves. Inovação em Pauta: FINEP, Rio de Janeiro, n. 8, p.27, jan. 2010.

Villani E.; Sutério, R.; Trabasso L.G.; Furtado L.F.F.; Alvarado B.H.L.; Amorim D.Y.K. (2010). Avaliação metrológica de um robô industrial para montagem estrutural de aeronaves. SBA Controle Automação, Vol. 21, Issue 6, pp. 634–646. DOI:10.1590/S0103-17592010000 600009.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.