International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 11, November-2020

Derivative Analysis Of Compensated Machine Coordinates Vectors

Samoil Samak
Institute for Advanced Composites and Robotics
IACR
Prilep, North Macedonia

Igor Dimovski
Institute for Advanced Composites and Robotics
Prilep, North Macedonia

Vladimir Dukovski
Faculty of Mechanical Engineering
University "St. Cyril and Methodius"
Skopje, North Macedonia

***Abstract*—Beside there are lot of papers about parametric robot calibration, there is lack of research focused on the last step of the parametric calibration procedure – compensation. In this paper, an iterative numerical compensation algorithm is described in detail. In the case when the robot's movement is controlled using its machine coordinates in order to compensate the robot's errors, the issue is if such changes in machine coordinates eventually cause some unwanted changes or constraints violations in the feed or acceleration profile. Derivative analysis of compensated machine coordinates for robot motion typical in composite industry use case, showed there are not any violations of the constraints raised out of velocity, acceleration or jerk limitations, for all robot's axes**

Keywords—Robot calibration; compensation; derivative analysis

I. INTRODUCTION

In composite industry, the robot accuracy is one of the most important factors needed in the technological process. Any deviation from the given pose (position and orientation) of the robot tool could cause some product defect. In composite manufacturing, AFP (Automated fiber placement) or ATL (Automated tape laying) head, with very complex mechanical structure is used as a robot tool.

Most common defects in AFP/ATL technologies are gaps between fibers/tapes out of predefined tolerance, overlap of consecutive tapes, laminate wrinkling, local buckling of the first ply etc. More details about composite product defects, their sources and comprehensive analysis of such defects' types are given in Jordaens & Steensels [1].

One can conclude, an improvement in the robot accuracy is one of the key factors in reducing composite product defects, despite it not the only factor. To avoid composite product defects, entire process should be controlled in real time, including defect detection and robot accuracy control.

Most common approach in robot accuracy improvement is parametric calibration. Most comprehensive review of robot accuracy improvement state of the art is given in Abderrahim et al [2]. They define the robot calibration as "a process by which manipulator real parameters' values affecting its accuracy are established through direct or indirect measurement and used to improve its accuracy by modifying the positioning software".

Up to some modifications, the parametric calibration procedure follows these steps:

1. Establish kinematics model
2. Measurement
3. Robot's parameters identification
4. Compensation

Additionally, non-geometric sources of errors as temperature, links' stiffness, loads etc, in a similar manner could be included in the parametric calibration procedure.

Optotrac measurement system used to determine the tool position only and the Krypton K600 system with three linear CCD cameras developed by Metris company as well are described in [2]. Position accuracy is improved reducing the position error from 3.25mm to 0.29mm and the orientation error from 5.43mrad to 0.35mrad, for a robot with load of 158kg.

All the researches related to the robot's parametric calibration mainly follow the 4 steps procedure described above. There are some variations in type of metrological equipment, choices of sample of measured poses in the robot's workspace and whether nongeometric parameters are included or not.

Nguyen et al. [3] for example, have developed a model to calibrate 27 DH robot parameters, measuring set of positions that follow circle path. Their results are experimentally confirmed on Hyundai HA-06 robot, reducing the average volumetric positional error from 3.657mm before the calibration, to 0.129mm after the calibration. Orientation error expressed by the Euler angles is reduced from [0.330o, 0.672o, 1.400o] in average, to [0.009o, 0.017o, 0.013o] after the calibration.

Similar research is conducted by Santolaria et al. [4], taking specific choice of measurement points made by Circle

Point Analysis approach. This approach advances are having clear geometric and physical meaning of the errors, since the robot tool is forced to describe circle movement, employing one particular robot's axes at a time. They have used active target in order to achieve greater ranges measurements, but that increase the uncertainties. Some results can be found in [4] for the experiments made on Kuka KR 5 sixx R650 robot. One can conclude beside most clear result in geometric representation of measured errors, improvement in total volumetric error reduction is significantly worse – 33%-40% compared to 70-90% of improvement achieved by the models that perform parameters optimization in formal manner, without consideration of their physical meaning.

Very accurate contact sensor in combination to specially designed artefact of 3 spheres is used to calibrate 25 robot parameters in the research presented by Joubair & Bonev [5]. The details of parametric calibration algorithm to identify 36 geometric parameters of KUKA 480 R3330 robot could be found in [6]. Only positional errors are taken into consideration and 2 levels of algorithm validation are provided – using simulation and experimentally.

Zhou & Kang [7] go step beyond, including non-geometrical parameters in the optimization procedure, using genetic algorithm. They emphasize the stiffness of the robot's links as key reason why geometric parameters calibration does not provide maximal accuracy improvement across entire workspace, especially in a case of complex tool as significant load. Experimental verification shows average positional error reduction to 0.091mm. Compared to results in [3], including non-geometrical parameters in robot calibration procedure significantly improve the robot accuracy.

Comparison between geometric and non-geometric robot calibration is given in Joubair er al. [8]. The experimental data given there refer to Fanuc LR Mate 200 iC robot with both of approaches and using more than 7000 measurement points. Average positional error before calibration is 0.622mm. After the geometric calibration it is reduced to 0.250mm and after including stiffness of the last 5 robot's axes, it is reduced to 0.142mm. Tool orientation is not taken into consideration and an important note is given that only 200 measurement points are sufficient to identify all the robot parameters. More details how to include non-geometric parameters in the mathematical model are given by Kamil et al. [9]. In this research, different loads are used in order to experiment the impact of stiffness parameters. Experimental data refer to ABB IRB 1600 robot and the maximal positional error is 0.960mm if stiffness parameters are used, compared to 2.571mm if just geometric parameters are used.

In our research, we have designed parametric robot calibration algorithm including 30 geometric parameters based on the kinematic model built as screw theory model [10] and 5 stiffness parameters. This approach is experimentally confirmed on machine configuration consists of KUKA KR500 robot with ATL head manufactured by Mikrosam company. The average positional error is reduced from 2.040mm before the calibration, to 0.580mm after the calibration. Orientation deviations are included in the optimization procedure as well. Improvements in the orientation accuracy are obtained, reducing the total orientation error from 0.147o to 0.095o in average.

Making literature review, one can notice less information are provided for the step for in typical parametric calibration procedure – compensation. Usually, calibrated parameters are sent to the robot controller and there is lack of technical details how are they used to compensate the robot's errors. This issue is especially important from robot's kinematic and dynamic analysis point of view, in the case when the robot's movement is controlled using its machine coordinates in order to compensate the robot's errors. Interpolation also has some impact on the robot accuracy and some fluctuations could eventually lead to unwanted jumps in the velocity or acceleration profiles, for some of the robot axes.

In the following section, our iterative numerical compensation algorithm is elaborated in detail. In the section 3, derivative analysis of the machine coordinates is provided. Conclusions are given in the last section.

II. ITERATIVE NUMERICAL COMPENSATION ALGORITHM

Let $P_{desired}$ be a given pose vector of the 6R robot tool and its scalar components with respect to the reference coordinate system (RCS) are:

$$P_{desired} = \left[X, Y, Z, A, B, C\right]_{RCS}^{T} \tag{1}$$

The first three coordinates of the desired pose $P_{desired}$ refer to the position of tool center point (TCP) in millimeters, and the last three coordinates are Euler angles in degrees, and they refer to the tool orientation with respect to RCS. This format of desired pose in transformed to homogeneous 4x4 matrix for the purpose of easy calculations:

$$T = \begin{bmatrix} n_x & l_x & o_x & p_x \\ n_y & l_y & o_y & p_y \\ n_z & l_z & o_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

In order to determine appropriate machine coordinates for the given pose, the inverse kinematics algorithm based on the Paden-Kahan subproblems could be applied. Details of algorithmic approach of the generalized form of these subproblems are given in Dimovski et al [10]. Using this approach, 8 solutions of machine coordinates are usually determined, so trajectory planning algorithm is used to choose the most optimal one. For 6R industrial robot used in this research, 6 machine coordinates respective to 6 rotational axes are obtained:

$$\Theta_{ideal} = \left[\theta_{1,i}, \theta_{2,i}, \theta_{3,i}, \theta_{4,i}, \theta_{5,i}, \theta_{6,i}\right]^{T} \tag{3}$$

There are at least two reasons why this approach is idealized and mostly theoretical. First, nominal robot parameters are user, and second assumptions like last 3 axes of rotation are concurrent – they intersect at a single point in 3D space. In reality, there are always some deviations of such idealized assumptions. The point is – such deviations are quantified in the calibration procedure, so calibrated parameters should be used instead of nominal ones.

No matter what kind of calibration algorithm is applied, the new set of calibrated parameters is obtained, and they differ from the nominal parameters. Also, these calibrated values of the parameters cannot be simply substituted in the same inverse kinematics procedure, since changing the parameters values, the assumptions of parallelism, orthogonally and concurrency are not valid anymore.

Because of that, iterative numerical compensation algorithm is designed in order to calculate compensated machine coordinates appropriate to the given desired pose *P_{desired}*, using the new set of calibrated values of the robot parameters.

$$\Theta_{compensated} = \left[\theta_{1,c}, \theta_{2,c}, \theta_{3,c}, \theta_{4,c}, \theta_{5,c}, \theta_{6,c}\right]^T \quad (4)$$

This algorithm is based on the Newton method. This approach's fundamentals are given by Curry [11]. In the contemporary applied mathematics literature, more specifically papers refer to optimization problems, some details of the Newton method, its application and variety of modifications and implementations could be found, like in [12], [13].

The desired pose coordinates $P_{desired}$ are input in this algorithm, as well as the ideal machine coordinates Θ_{ideal}, used as initial guess. The algorithm actually iteratively makes deviation reduction in order to get as closer as possible to the desired pose $P_{desired}$, in the pose space, to some predefined tolerance ε that also comes as input in the algorithm. One additional input value is needed – iterative step δ as a measure of machine coordinates changes in each iteration.

In the first iteration, machine coordinates are set to the initial guess values:

$$\Theta^1_{compensated} \leftarrow \Theta_{ideal} \quad (5)$$

The forward kinematics algorithm with calibrated values of the parameters is called in the next step and its output is used to determine deviation from the desired pose $P_{desired}$. Such deviations are calculated and transformed into the pose space in order to obtain 6 scalar components of the deviations vector:

$$F_x = \left[\Delta_X, \Delta_Y, \Delta_Z, \Delta_A, \Delta_B, \Delta_C\right]^T \quad (6)$$

The vector $F_{x,i}$ determines the contribution of i-th machine coordinate in pose coordinates changes. Next, the differences

$$F_{x,i} - F_x \quad (7)$$

are calculated for every i = 1,2,3,4,5,6. These 6 vectors with length 6 are organized as 6x6 matrix and that matrix divided by the iterative step δ is actually Jacobian matrix *J*.

The machine coordinates $\Theta^1_{compensated}$ are then corrected by Jacobian inverse multiplied by the vector *Fx*:

$$\Theta^2_{compensated} = \Theta^1_{compensated} - J^{-1} \cdot F_x \quad (8)$$

The result of the first iteration $\Theta^2_{compensated}$ is used as input in the second iteration. Completely the same procedure is applied until the objective function defined as a square of the magnitude of the vector F_x become small enough, within the tolerance ε. That means iteration procedure stops when the condition:

$$\left\|F_x\right\|^2 < \varepsilon \quad (9)$$

is satisfied.

In this research, described iterative algorithm is implemented using the tolerance value of $\varepsilon = 10^{-15}$ and iterative step value of $\delta = 10^{-8}$. In most cases the iterative numerical compensation algorithm converges after 3-5 iterations.

Method for calculating the compensated machine coordinates after parametric robot calibration procedure, for given particular point $P_{desired}$ in the pose space is described in detail above. In practice, when 6R industrial robot, or even more complex machine configuration is used in composite tape laying is controlled by machine coordinates, usually an array of desired poses comes as input and G-code of machine coordinates should be produced in order to send it to the controller. That G-code is produced in some offline procedure such that for every desired pose, the described iterative numerical compensation algorithm is called in order to calculate appropriate compensated machine coordinates $\Theta_{compensated}$, based on calibrated values of the robot parameters, no matter what kind of parametric calibration procedure is performed to derive such calibrated robot parameters.

Beside the given desired poses, additionally machine configuration trajectory planning is very important in composite manufacture, since large number of machine coordinate vectors need to be interpolated through the G-code points and tool path has to be controlled and highly accurate along all the path. In online interpolation procedure, but in offline as well, the iterative numerical compensation algorithm calculations need to be performed very fast, and in the online case in real time.

Challenge is additional check if such changes in the machine coordinates due to compensation cause unwanted changes in feed, acceleration or jerk for each of the robot or machine configuration drivers and eventually violation of their limitations.

III. DERIVATIVE ANALYSIS OF THE MACHINE COORDINATES AS A FUNCTION OF TOTAL AXES DISPLACEMENT

In order to check if some unwanted changes and violation of feed, acceleration or jerk constraints occurs, the array of compensated coordinates vectors is analyzed against appropriate uncompensated coordinates vectors.

Tool path defined as line in 3D physical space, with length of approximately 1500mm is the use case analyzed in this research. Since the tool path is straight line, theoretically it could be defined using G-code of only 2 points. But, in this case, the interpolation between these points is completely left to the controller and the tool orientation is not controlled as well. That is inadmissible, especially in the case of robot usage in composite tapes laying, since then following the predefined position and orientation of the tool along all the path is extremely important. In the practice, additional number of poses is offline programmed, usually taking uniform Euclidian distance travel of the tool center point.

In this use case, the straight line is divided to 297 points at a uniform distance of 5mm and positions and orientations are determined for each of these points as in equation (1). Taking the nominal robot parameters and calling the standard Paden-Kahan inverse kinematics procedure, Θ_{ideal} vector of machine coordinates is calculated, for each of 297 poses. This way, the G-code of uncompensated machine coordinates is created.

The same 297 poses are used calling the iterative numerical compensation algorithm, described in the previous section, in order to obtain another G-code of compensated machine coordinates, concerned to the same path, the tool should follow.

Tool poses refer to the Pose Space and Euclidian distances used to generate such uniformly distributed poses along the tool path refer to the physical 3D space where the robot motion is done. That does not mean each of the robot axes would perform uniform displacement. Controlling the robot motion sending the machine coordinates to the robot controller, the focus of interest is moved to the Machine Space. In the case of 6R industrial robot used in this research it is 6D machine space. Beside between any 2 successive points in the G-code, the tool always should travel 5mm displacement, in general, every robot axis should perform different displacement if any. As a consequence, velocities of the robot axes are different in different point of time. In this case, trajectory planning is left to the controller and it applies its own algorithms to interpolate several values taking into account to not violate the machine configuration dynamics.

In order to take full control of each robot axis motion, some algorithm of the look-ahead class of algorithms should be applied, so after the complete trajectory planning every machine coordinate would be expressed as function of time t. Analysis of the performances for some of these algorithms can be found in [14]. Application of such trajectory planning algorithm to express the robot machine coordinates with respect to time t is out of the scope of this particular research.

One of the most important researches in the area of machine configuration trajectory planning, considering the feed optimization is elaborated by Sencer et al. [15]. Nomenclature used in this paper is especially useful and it allows making clear distinction between trajectory planning as a function with respect to total vector displacement s of the axes in the machine space and optimization of the velocities of each of the axes expressing the axis displacement as a function with respect to the time t. Even more, feed, acceleration and jerk constraints for each machine configuration axis is expressed with respect to total vector displacement s, allowing trajectory planning, optimizing the feed and in this case, allowing a simple analysis if there are any constraints violations, before trajectory planning is done.

More details how such constraints could be derived are given by Altintas & Kaan [16].

Starting from 297 vectors of machine coordinates given with appropriate G-code, total vector displacement of the robot axes is calculated using Euclidian metric in 6D machine space. The total axes vector displacement s is uniformly divided to 2900 parts, and for every appropriate value of s, interpolated values of each machine coordinate is obtained by linear interpolation:

$$\mathbf{q}(s) = [\theta_1(s), \theta_2(s), \theta_3(s), \theta_4(s), \theta_5(s), \theta_6(s)]^T \qquad (10)$$

The velocity of the axis *i* is denoted by (*i*=1,2,3,4,5,6):

$$\dot{\theta}_i = \frac{d\theta_i}{dt} \qquad (11)$$

The acceleration of the axis *i* is denoted by:

$$\ddot{\theta}_i = \frac{d^2\theta_i}{dt^2} \qquad (12)$$

The jerk of the axis *i* is denoted by:

$$\dddot{\theta}_i = \frac{d^3\theta_i}{dt^3} \qquad (13)$$

Consequently, the vectors of velocity, acceleration and jerk are denoted by $\dot{\mathbf{q}}(s)$, $\ddot{\mathbf{q}}(s)$ and $\dddot{\mathbf{q}}(s)$ respectively. Evaluation of these functions can be done only after the trajectory planning. In fact, they are used in trajectory planning optimization - machine configuration motion program, minimizing the total time needed to pass entire path and keeping all feed, acceleration and jerk constraints satisfied.

Equation (10) determines an array of values for *i*-th machine coordinate (*i*=1,2,3,4,5,6), for each of 2900 values of total vector displacement s of the robot axes.

Numerical procedure for approximation of the first, second and third derivative with respect to the total vector displacement s are applied. These derivatives are denoted as follows:

$$\theta_{i,s} = \frac{d\theta_i}{ds} \qquad (14)$$

$$\theta_{i,ss}, = \frac{d^2\theta_i}{ds^2} \qquad (15)$$

$$\theta_{i,sss} = \frac{d^3\theta_i}{ds^3} \qquad (16)$$

This way, vectors containing the first, second and third derivative of the 6 robot axes positions with respect to total vector displacement s are obtained:

$$\mathbf{q_s}(s)=\left[\theta_{1,s}(s),\theta_{2,s}(s),...,\theta_{6,s}(s)\right]^T \tag{17}$$

$$\mathbf{q_{ss}}(s)=\left[\theta_{1,ss}(s),\theta_{2,ss}(s),...,\theta_{6,ss}(s)\right]^T \tag{18}$$

$$\mathbf{q_{sss}}(s)=\left[\theta_{1,sss}(s),\theta_{2,sss}(s),...,\theta_{6,sss}(s)\right]^T \tag{19}$$

If the maximal values of each axis velocity are given, the normalized partial derivatives could be expressed:

$$q^v_{i,s}=\frac{\theta_{i,s}(s)}{v_{i,\max}} \tag{20}$$

The velocity (feed) maximal allowed values, for each robot axis, for particular value of total vector displacement *s*, are determined by:

$$\dot{s}_V \max(s)=\frac{1}{\max_{i=1,2,3,4,5,6}\left\{\left|q^v_{i,s}(s)\right|\right\}} \tag{21}$$

Derivation of the equation (21), as well as acceleration and jerk constraints derivation details could be found in [6].

The equation (20) actually determines the first derivatives constraints of the functions $\theta_{i,s}(s)$, raised out of velocity limitations. The first derivatives have to satisfy the constraints as well raised out of the acceleration and jerk limitations.

$$q^a_{i,s}=\frac{\theta_{i,s}(s)}{a_{i,\max}} \tag{22}$$

$$q^j_{i,s}=\frac{\theta_{i,s}(s)}{j_{i,\max}} \tag{23}$$

In total, there are 3 constraints for the first derivatives of the functions $\theta_{i,s}(s)$. In the equations (22) and (23), notations $a_{i,\max}$ and $j_{i,\max}$ are acceleration and jerk limitations for *i*-th axis respectively.

Similarly, the second derivative of the functions $\theta_{i,s}(s)$ with respect to the total vector displacement of the robot axes has to satisfy the following 2 constraints raised out of acceleration and jerk limitations respectively, for each robot axis:

$$q^a_{i,ss}=\frac{\theta_{i,ss}(s)}{a_{i,\max}}$$

(24)

$$q^j_{i,ss}=\frac{\theta_{i,ss}(s)}{j_{i,\max}}$$

(25)

The third derivative of the functions $\theta_{i,s}(s)$ with respect to the total vector displacement of the robot axes has to satisfy 1 constraint raised out of jerk limitations, for each robot axis:

$$q^j_{i,sss}=\frac{\theta_{i,sss}(s)}{j_{i,\max}} \tag{26}$$

According to that, the maximal allowed feed values, satisfying the maximal acceleration and jerk limitations, for each robot axis, as a function with respect to total vector displacement s are determined by:

$$\dot{s}_A,\max(s)=\sqrt{\frac{1}{\max_{i=1,2,3,4,5,6}\left\{\left|q^a_{i,ss}(s)\right|\right\}}} \tag{27}$$

$$\dot{s}_J,\max(s)=\sqrt[3]{\frac{1}{\max_{i=1,2,3,4,5,6}\left\{\left|q^j_{i,sss}(s)\right|\right\}}} \tag{28}$$

Visual analysis of the first, second and third derivative of the machine coordinate θ_1, referred to A_1 robot's axis is shown on Fig. 1. First four diagrams show the displacement $\theta_1(s)$ and its derivatives $\theta_{1,s}(s)$, $\theta_{1,ss}(s)$ and $\theta_{1,sss}(s)$ with respect to *s*, taking $s\in[0.4;1.5]rad$. Taking wider range of the total vector displacement s allows an analysis of the changes of appropriate functions, how frequent such changes happen, are there any unwanted fluctuations and if there is some significant difference between compensated and uncompensated values. If there are some violations of the limitations raised out of velocity, acceleration and jerk of the robot's first axis, could be noticed as well.

Blue graphs refer to the uncompensated values and the red ones to the compensated values of the machine coordinate θ_1 of the robot's axis A_1, for all diagrams.

One can conclude there are not significant deviations between uncompensated and compensated values; especially there are not such deviations that may cause limitations violation. The red graphs are slightly shifted compared to the blue ones, without significant increase or decrease of the maximal or minimal values in the observed interval.

First diagram simply shows the machine coordinate $\theta_1(s)$ as a function with respect to *s*.

Second diagram shows the first derivative $\theta_{1,s}(s)$ and as well the constraints raised out of velocity limitations of the appropriate robot's axis, shown as dotted green line; the constraints raised out of acceleration limitations of the appropriate robot's axis, shown as dotted magenta line; and the constraints raised out of jerk limitations of the appropriate robot's axis, shown as dotted cyan line. One can conclude there are not violations of all the constraints at all.

Third diagram shows the second derivative $\theta_{1,ss}(s)$ and as well the constraints raised out of acceleration and jerk limitations of the robot's axis A_1, shown in similar manner – using magenta and cyan dotted line respectively.

Finally, the fourth diagram shows the third derivative $\theta_{1,sss}(s)$ and the constraints raised out of jerk limitations of the robot's axis A_1, shown as dotted cyan line.

This analysis shows there are not limitations violations neither for first, second or third derivative on the observed interval.

In order to capture the more frequent changes of the presented function, the interval $s\in[0.710;0.725]rad$ is used in last four diagrams to zoom the diagrams from the previous four diagrams.

IV. CONCLUSION

Parametric calibration algorithm is designed and is experimentally confirmed on machine configuration consists of KUKA KR500 robot with ATL head manufactured by Mikrosam company. As result, the average positional error is reduced from 2.040mm before the calibration, to 0.580mm after the calibration and the total orientation error is reduced as well from 0.147^o to 0.095^o in average.

In order to include identified robot parameters in the robot control, using machine coordinates, iterative numerical compensation algorithm is designed and explained in detail.

The issue if some unwanted changes would appear as result of machine coordinates changes, derivative analysis of the machine coordinates as a function of total axes displacement is provided. As use case, typical robot motion in composite industry is used and derivative analysis showed there are not any constraints violations neither related to velocity, acceleration or jerk limitations of the robot axes.

REFERENCES

[1] Jordaens, A., Steensels, T. “Formation of defects in flat laminates during automatic tape laying” (framework of a master’s thesis), Faculty of engineering technology, Leuven, Belgium, 2015

[2] Abderrahim, M., Khamis, A., Garrido, S., Moreno, L. “Accuracy and Calibration Issues of Industrial Manipulators” – chapter in Low, K.H. (ed) “Industrial Robotics - Programming, Simulation and Applications”, pp.131-146, ISBN: 3-86611-286-6, I-Tech, 2007

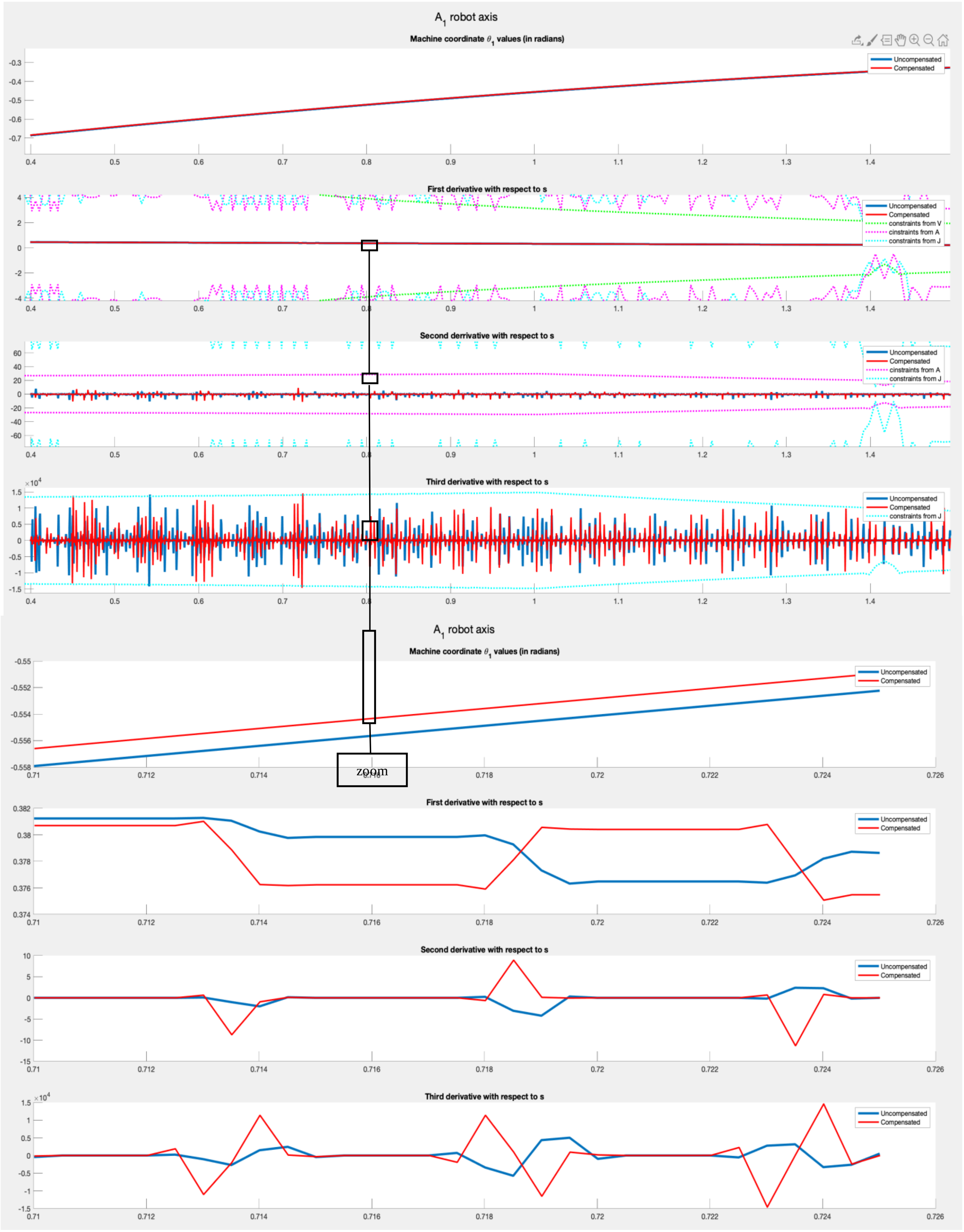

Fig. 1 First, second and third derivative analysis of the machine coordinate θ_1 with respect to s

[3] Nguyen, H-N., Zhou, J., Kang, H-J. “A New Full Pose Measurement Method for Robot Calibration”, Sensors 2013, 13, 9132-9147, DOI: 10.3390/s130709132, 2013

[4] Santolaria, J., Conte, J., Ginéz, M. “Laser tracker-based kinematic parameter calibration of industrial robots by improved CPA method and active retroreflector”, The International Journal of Advanced Manufacturing Technology, 66(9-12), pp.2087-2106, DOI:10.1007/s00170-012-4484-6, Springer, 2013

[5] Joubair, A., Bonev, I.A. “Kinematic calibration of a six-axis serial robot using distance and sphere constraints”, The International Journal of Advanced Manufacturing Technology, 77(1-4), pp.515-523, DOI:10.1007/s00170-014-6448-5, Springer, 2015

[6] Samak, S., Dimovski, I., Trompeska, M., Hristoski, M., Kochoski, F., Dukovski, V. “Computer-based simulation and validation of robot accuracy improvement method and its verification in robot calibration procedure”, ETAI 2018 International Conference, ETAI Society of Macedonia, 2018

[7] Zhou, J., Kang, H-J. “A hybrid least-squares genetic algorithm-based algorithm for simultaneous identification of geometric and compliance errors in industrial robots”, Advances in Mechanical Engineering 2015, Vol. 7(6) I-12, DOI:10.1177/1687814015590289, SAGE, 2015

[8] Joubair, A., Nubiola, A., Bonev, I. “Calibration Efficiency Analysis Based on Five Observability Indices and Two Calibration Models for a Six-Axis Industrial Robot”, SAE International Journal of Aerospace 6, no. 2013-01-2117, pp. 161-168, DOI:10.4271/2013-01-2117, 2013

[9] Kamali, K., Joubair, A. Bonev, I.A., Bigras, P. “Elasto-geometrical Calibration of an Industrial Robot under Multidirectional External Loads Using a Laser Tracker”, 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 4320-4327, DOI: 10.1109/ICRA.2016.7487630, IEEE, 2016

[10] Dimovski, I., Trompeska, M., Samak, S., Dukovski, V., Cvetkoska, D. “Algorithmic approach to geometric solution of generalized Paden-Kahan subproblem and its extension”, International Journal of Advanced Robotic Systems, 2018:I-II, DOI: 10.1177/1729881418755157, SAGE, 2018

[11] Curry, H.B. “The Method Of Steepest Descent For Non-linear Minimization problems”, Quarterly of Applied Mathematics 2.3 (1944), pp. 258-261, 1944

[12] Nestorov, Yu. ”Introductory Lectures on Convex Optimization: A Basic Course”, Springer, 2004

[13] Lemaréchal, C. “Cauchy and the gradient method", Doc Math Extra, 251 (2012): 254, 2012

[14] Dimovski, I., Samak, S., Cvetkoska, D., Trompeska, M., Kochoski, F. “Speed control in numeric controlled systems”, Математички билтен Vol.39(LXV) No.1 2015 (49-63), ISSN 0351-336X, UDC: 519.713-74, CMM, 2015

[15] Sencer, B., Altintas, Y. & Croft, E. “Feed optimization for five-axis CNC machine tools with drive constraints”, International Journal of Machine Tools and Manufacture, 48. 733-745. DOI:10.1016/j.ijmachtools.2008.01.002, 2008

[16] Altintas, Y., and Kaan, E. "Feedrate optimization for spline interpolation in high speed machine tools", CIRP Annals 52.1 (2003): 297-302, 2003