

# STUDY ON IMPROVING THE MEASUREMENT ACCURACY BY TOUCH PROBE WITH A COBOTIC MULTI-APPLICATION PLATFORM

Paul-Nicolae Ancuța<sup>1</sup>, Anghel Constantin<sup>1</sup>, Iulian-Sorin Munteanu<sup>2</sup>, Valentin Gornoavă<sup>3</sup>

<sup>1</sup>The National Institute of Research and Development in Mechatronics and Measurement Technique (INCDMTM), Bucharest, Romania

<sup>2</sup>Faculty of Industrial Engineering and Robotics, Bucharest, Romania

<sup>3</sup>Nicprem Impex SRL

Email: [paul.ancuta@incdmtm.ro](mailto:paul.ancuta@incdmtm.ro)

**Abstract** - This paper presents some comparative results obtained from a batch of tests in order to determine the optimal software algorithm implementation designed for the command of a multi-application cobotic platform with remote control function. The application shown is a high precision measurement system for the movement of a high-precision touch probe manipulated with an intelligent cobotic platform (consisting of two robots series-parallel with 11 DOF) in order to obtain accurate measurement values. The movement of the touch probe is done with high precision using a hexapodal system.

**Keywords:** COBOT, hexapod, High precision touch probe, Least square circle detection.

## 1. Introduction

The paper presents the experiments and result that have been carried out in order to determine the optimal method for dimensional measurement using a high precision touch probe attached to a hexapod robot, which in turn is attached to the arm of an UR5e cobot. The cobot's arm movement is accomplished using a software program developed initial in PolyScope. The movement of the hexapod robot is accomplished using the manufacturer software PI MikroMove and the proprietary original software developed during the project, in combination with the complex movement of the cobot.

## 2. Related Works

Paper [1] presents a study about the functionality of a compact robotic system consisting in two hexapods serially connected and controlled in such a way that they have identical configurations any time, in terms of accuracy and manoeuvrability.

The double hexapod system may be used as positioning systems for emerging new technologies and possibly in robotic surgery.

Paper [2] presents the direct dynamics study of Gough-Stewart platforms mounted in parallel one above the other. The dynamics of each solid of the hexapod platform comprises 12 scalar differential equations (3 for each translation and 9 for each solid

rotation) and 6 algebraic scalar orthogonality equations, plus the algebraic constraints.

## 3. Materials and Method

A measuring configuration consisting of a robotic arm[3], a hexapod[4] and a TP200 electronic probe[5] having a M2 stylus( 4mm diameter) connected was used to perform the experiments.

The hexapod is fixed with a flange on the last segment of the UR5e Cobot arm and it is connected to the industrial computer IPC via a dedicated C-887.52x controller. The probe is also connected to a specialized PI200-3 controller.

Figure 1 shows the hardware assembly:



Figure 1: Cobotic platform configuration

A trajectory of motion consisting of a set of points in the working space of the Cobot has been defined.

The movement of the Cobot's arm from one point to another is done using a software program written in PolyScope, which is the GUI software of the UR5 Cobot.

The measurement begins when the Cobot's arm reaches one of the measuring points of the trajectory. The PC software controls the movement of the hexapod axes from position [0,0,0,0,0] to the destination position, monitors the event that triggers the measured elevation values, and commands the axes to return to position [0,0,0,0,0]. The measurement ends after a predefined number of repetitions of movement on the X axis or operator intervention.

The method of measuring by palpation of the mechanical part is shown in Figure 2.



Figure 2: Touch probe measurement method

Experiments were performed to measure the motion on the X axis as well as to measure the circumference of the hole shown in Figure 2.

All the fastening elements, the flange and the accessories were made in the **CERMISO Center within INCDMTM**. These have been machined on Makino D200Z (5-axis machining), Makino U HEAT 6 (wire EDM) and Makino EDAF 2 (wire EDM) high-precision machines, ensuring tens of micrometres of precision in execution.

• **Measuring procedure using PI MikroMove Software**

Physik Instrumente provides software developers with a set of programs that includes GCS (General Command Set). GCS facilitates the development of dedicated macros and contains the necessary software functions to be integrated in programs implemented in MATLAB, LABVIEW, Visual C ++ and other programming languages.

A software application provided by Physik Instrumente, which incorporates GCS, is PIMikroMove Software [6]. This application allows positioning and visualization of the movements of each axis of the hexapod.

The research team developed a macro embedded in PIMikroMove software for moving the X axis in the range 0..14 mm and returning to the home position, when the probe (connected to the DIO4 digital input of the cobot) encounters an obstacle.

The source code of the macro and the obtained results are presented in the Figure 3 .

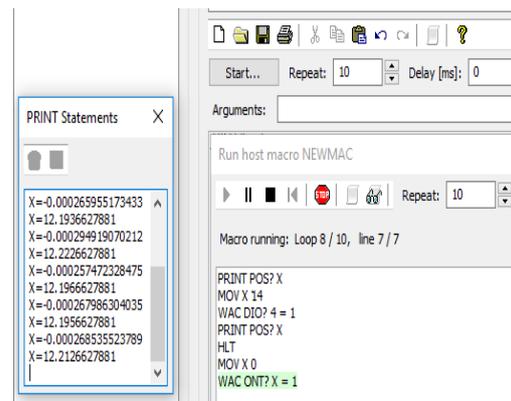


Figure 3: Macro for measuring X-axis displacement

• **Measuring procedure using original Software**

An original piece of software was written to measure and monitor the movement of the hexapod. The software uses a PIPython library which a collection of modules that communicate with the hexapod via a GCS command wrapper. The software application uses several threads. The threads functionality is presented as follows:

-The **COBOT** data exchange and command thread receives the message that is issued by the Cobot when the Cobot arm has reached a measuring point, signals this message to the measuring thread, waits for the signal generated at the end of the measurement carried out by hexapod, and sends this signal back to the Cobot for positioning the Cobot arm on the next point of the trajectory

-The **probe** thread monitors the digital value of the electrical signal emitted by the probe and signals this event to the hexapod measurement thread.

-The **hexapod measurement** thread controls the speed of movement of the translation axes of the hexapod, acquires the position of the hexapod when electronic probe encounters an obstacle, processes the acquired data set and delivers the measurement values

There is also an optional thread used for viewing in real-time the current position of each axis of the hexapod during its movement.

The software application performs the following types of measurements:

- segments measurements
- circle measurements
- repeatability of the linear movement

During the tests the hexapod moved on the X and Y axis with speeds in the range [1..5] mm / s. Higher speeds randomly generate false palpation signals, due to the force of inertia generated by the movement of the hexapod, which has a mass of about 2.5 kg.

Figure 4 shows the significant area of the software application's screen when determining the repeatability of the X-axis movement

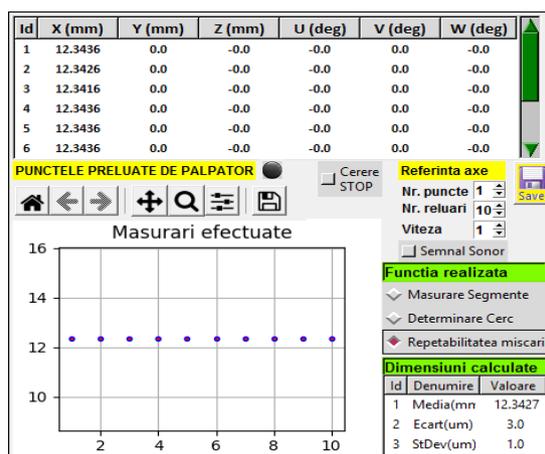


Figure 4: Software application screen for determining the repeatability of X axis movement

The following values have been calculated for each set of measured data:

$$Ecart = \text{Max}(x_i) - \text{Min}(x_i)$$

$$Media = \text{Average}(x_i)$$

$$StdDev = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (1)$$

- **Algorithms used for circle measurement**

Figure 5 shows the screen of the original software application related to circle measurement.

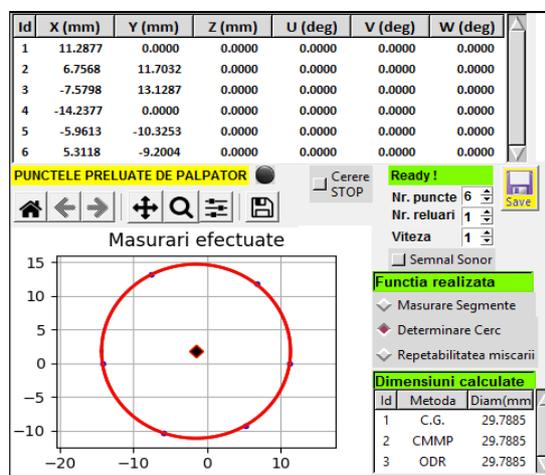


Figure 5: Circle measurement software application screen

Three algorithms for circle measurement have been implemented in our software application.

**a) Algebraic approach**

Paper [7] presents a method of finding the least square circle of a set of N points belonging to  $R^2$ . Let

$$\bar{x} = \frac{1}{N} \sum_i x_i; \bar{y} = \frac{1}{N} \sum_i y_i \quad (2)$$

$$g(u, v) = (u - u_c)^2 + (v - v_c)^2 - \alpha \quad (3)$$

Where  $(u_c, v_c)$  are the coordinates of the centre of the circle expressed in  $(u, v)$  coordinates and  $\alpha = R^2$

The centre of the circle and the radius are obtained by minimizing the following function:

$$S = \sum_i (g(u_i, v_i))^2 \quad (4)$$

and express them in  $(x, y)$  coordinates.

The algorithm that has been used in the software implementation is a modified version of the original algorithm:

**Step1:** Calculate the estimated  $[x_{ce}, y_{ce}]$ . coordinates of the centre of the circle as means of the  $[x_i, y_i]$  coordinates of the point set.

**Step2:** Define the linear system of equations

$$S_{uu} * u_c + S_{uv} * v_c = \frac{(S_{uuu} + S_{uvv})}{2} \quad (5)$$

$$S_{uv} * u_c + S_{vv} * v_c = \frac{(S_{uuv} + S_{vvv})}{2} \quad (6)$$

**Step3:** Solve the above linear system of equations.

**Step4:** Determining the new coordinates of the centre of the circle:

$$x_{cf} = x_{ce} + u_c \quad (7)$$

$$y_{cf} = y_{ce} + v_c \quad (8)$$

**Step5:** Calculate the distances between each point  $[x_i, y_i]$  of the set of points and the circle's centre  $[x_{cf}, y_{cf}]$

**Step6:** Calculate the value  $R_c$  of the circle's radius as the arithmetic mean of the distances calculated in Step5 augmented by 2 mm (the radius of palpation stylus)

**b) Least square circle**

Scipy software package[8] provides fundamental algorithms and their numerical implementation for scientific computing in Python.

Scipy.optimize.leastsq method finds the set of parameters that minimize a residual.

The distance between a point  $[x_i, y_i]$  and the centre of the circle  $[x_c, y_c]$  is:

$$R_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \quad (9)$$

The residual that will be minimized is:

$$residual = R_i - \frac{1}{N} \sum_i R_i \quad (10)$$

The algorithm that has been used is as follows:

**Step1:** Calculate the estimated  $[x_{ce}, y_{ce}]$  coordinates of the centre of the circle as means of the  $[x_i, y_i]$  coordinates of the point set.

**Step2:** Scipy.optimize.leastsq is called having the following parameters: the function defined by the equation (10) and the initial coordinates  $[x_{ce}, y_{ce}]$ . The coordinates of the centre of the circle  $[x_{cf}, y_{cf}]$  that minimize the residual is obtained.

**Step3:** Calculate the distances between each point  $[x_i, y_i]$  and the centre of the circle  $[x_{cf}, y_{cf}]$ .

**Step4:** Calculate the value  $R_c$  of the circle's radius as the arithmetic mean of the distances calculated in Step3 augmented by 2 mm (the radius of palpation stylus)

**c) Orthogonal distance regression**

Scipy.odr is a dedicated package used to deal with orthogonal distance regression. How to use scipy.odr is shown in [9].

The algorithm that has been used is as follows:

**Step1:** Determine the initial estimation of the parameters  $[x_{ce}, y_{ce}, R_m]$  as follows:

- calculate the estimated coordinates  $[x_{ce}, y_{ce}]$  of the center of the circle as arithmetic means of the coordinates  $[x_i, y_i]$  of the initial set of points

- calculate the value  $R_m$  as the average of the distances between each point  $[x_i, y_i]$  and the point having  $[x_{ce}, y_{ce}]$  coordinates.

**Step2:** Define the implicit function

The following implicit function definition that will be used is defined as follows:

$$(x - x_c)^2 + (y - y_c)^2 - R_c = 0 \tag{11}$$

**Step3:** Create the model having as parameter the default function determined in Step2

**Step4:** Create the data set having as parameters the set of the  $[x_i, y_i]$  coordinates of the initial points

**Step5:** Instantiate ODR with data, model and initial parameter estimate determined at Step4, Step3 and Step1

**Step5:** Run the model. The output data consists of the new centre coordinates  $[x_{cf}, y_{cf}]$  and the distances between each  $[x_i, y_i]$  point and  $[x_{cf}, y_{cf}]$  point

**Step6:** Calculate the value  $R_c$  of the circle's radius as the arithmetic mean of the distances calculated in Step5 augmented by 2 mm (the radius of palpation stylus)

**4. Experimental Results**

Table 1 gives a summary of the repeatability of moving using PIMikroMove Software and the macro described above.

Table 1. Results obtained using PIMikroMove macro

	Speed 1 mm/s	Speed 2 mm/s	Speed 3 mm/s	Speed 4 mm/s	Speed 5 mm/s
	12.2137	12.2170	12.2946	12.3173	12.3920
	12.2137	12.2370	12.2466	12.3573	12.3520
	12.2137	12.2470	12.2466	12.3093	12.3670
	12.1937	12.2370	12.2766	12.3533	12.4170
	12.2227	12.2110	12.2406	12.3533	12.3720
	12.1967	12.2410	12.2946	12.2933	12.3620
	12.1957	12.2630	12.2526	12.3013	12.3620
	12.2127	12.2350	12.2406	12.2853	12.4270
	12.2067	12.2130	12.2916	12.3013	12.3670
	12.1987	12.2130	12.2826	12.3533	12.4270
A	0.0290	0.0520	0.0540	0.0720	0.0750
B	12.2068	12.2314	12.2667	12.3225	12.3845
C	0.0100	0.0174	0.0233	0.0287	0.0290

Row A: (MAX-MIN) differences

Row B: AVERAGE values

Row C: Standard deviation of the measured values

Data are graphically depicted in Figure 6:

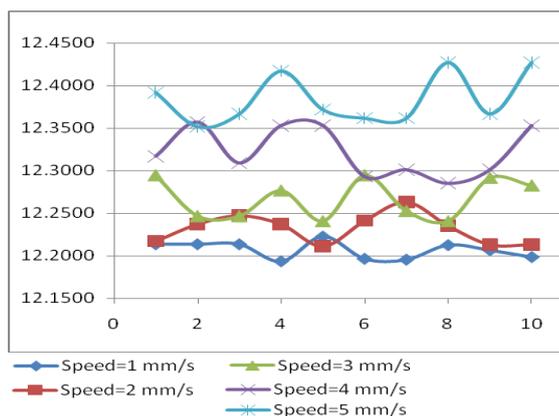


Figure 6: Measuring X-axis displacements using a piece of macro software

Table 2 gives a summary of the measuring data obtained from the hexapod using original software.

Table 2. Results obtained using original software

	Speed 1 mm/s	Speed 2 mm/s	Speed 3 mm/s	Speed 4 mm/s	Speed 5 mm/s
	12.3436	12.3470	12.3545	12.3613	12.3720
	12.3426	12.3490	12.3545	12.3653	12.3770
	12.3416	12.3470	12.3575	12.3653	12.3720
	12.3436	12.3470	12.3575	12.3613	12.3770
	12.3436	12.3450	12.3515	12.3613	12.3670
	12.3436	12.3490	12.3545	12.3653	12.3720
	12.3426	12.3470	12.3575	12.3693	12.3770
	12.3406	12.3470	12.3545	12.3653	12.3770
	12.3426	12.3450	12.3545	12.3653	12.3770
	12.3426	12.3470	12.3545	12.3653	12.3770
A	0.0030	0.0040	0.0060	0.0080	0.0100
B	12.3426	12.3470	12.3551	12.3645	12.3745
C	0.0011	0.0013	0.0019	0.0025	0.0035

Table 3 gives a summary of the measurement of the circle’s diameter using original software.

Table 3. Results obtained for the diameter of the circle

	Speed 1 mm/s	Speed 2 mm/s	Speed 3 mm/s	Speed 4 mm/s	Speed 5 mm/s
	29.7335	29.7445	29.7778	29.7972	29.8393
	29.7365	29.7470	29.7728	29.8026	29.8363
	29.7350	29.7510	29.7747	29.8010	29.8377
	29.7299	29.7525	29.7757	29.7999	29.8382
	29.7292	29.7477	29.7758	29.7986	29.8345
	29.7301	29.7484	29.7747	29.8027	29.8345
	29.7304	29.7512	29.7757	29.8054	29.8379
	29.7299	29.7506	29.7777	29.8040	29.8430
	29.7318	29.7512	29.7745	29.8026	29.8394
	29.7302	29.7512	29.7716	29.8012	29.8394
A	0.0073	0.0080	0.0062	0.0082	0.0085
B	29.7317	29.7495	29.7751	29.8015	29.8380
C	0.0025	0.0025	0.0019	0.0025	0.0025

Row A: (MAX-MIN) differences

Row B: AVERAGE values

Row C: Standard deviation of the measured values

Data are graphically depicted in Figure 7:

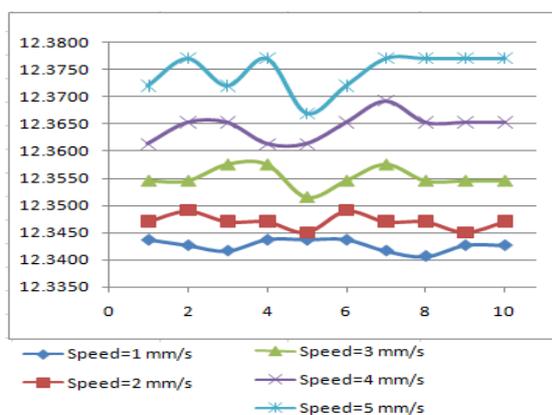


Figure 7: Measuring X-axis displacements using original software

Data are graphically depicted in Figure 8.

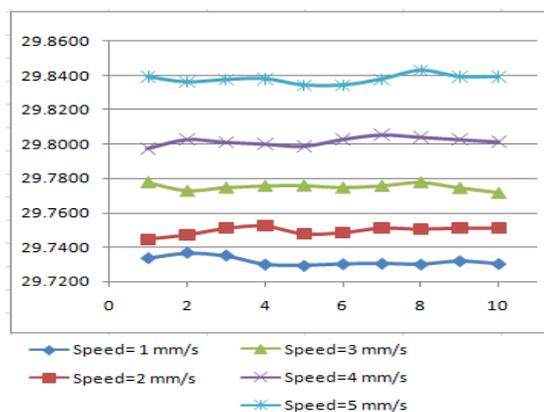


Figure 8: Circle diameter measurement using original software

## 5. Conclusions

By examining the results regarding the measuring of the repeatability of the X-axis movement the following observations have been generated:

- the lowest measured values were obtained using the MikroMove PI Software application, but these values have the highest standard deviation

- 1mm / s speed for X axis ensures, in all cases, the set with the lowest measured values and the smallest standard deviation

- increasing the speed of movement on the X axis causes increased value sets for the measured distance and increasing the standard deviation

- having speeds movement equal to 5 mm / s, the acquired data always have the fourth decimal (which is hundreds of nanometres) equal to 0.

Regarding measuring the circles, each increase in the speed of movement of the hexapod by 1 mm / sec determines an increase of the calculated value of the circle diameter by 20 microns

## Acknowledgements

This work was carried out through the NUCLEU Program, conducted with the support of MCID (Ministry of Research, Innovation and Digitization), project number: PN 19240201-17N/20219.

## References

- [1] Mărgăritescu, M., Dumitriu, D., Brişan, C., Rolea, A.M.E., Constantin, A.: Complex and robust motion performed in extended workspace with a double hexapod robotic system Open Access. *Mechanika* 26(6), pp. 532-539 (2020)
- [2] Dumitriu, D. N., Mărgăritescu, M., Rolea, A. M. E.: Direct dynamics of Gough-Stewart hex-apod platforms using the redundant parameterization of rotations by full rotation matrices. *International Journal of Mechatronics and Applied Mechanics* 2017(1), pp. 159-167 (2017)
- [3] <https://www.universal-robots.com/products/ur5-robot> accessed on: 18.05.2022
- [4] <https://www.physikinstrumente.com/en/products/parallel-kinematic-hexapods/h-811s2-6-axis-motion-hexapod-7008881/> accessed on: 18.05.2022
- [5] <https://www.renishaw.com/cmmsupport/knowledgebase/en/tp200-tp200b--22340> accessed on: 18.05.2022
- [6] <https://pimikromove.software.informer.com/> accessed on:14.04.2022
- [7] [https://dtcenter.org/sites/default/files/community-code/met/docs/write-ups/circle\\_fit.pdf](https://dtcenter.org/sites/default/files/community-code/met/docs/write-ups/circle_fit.pdf) last accessed on: 20.05.2022
- [8] <https://www.scipy.org> accessed on 20.05.2022
- [9] <https://docs.scipy.org/doc/scipy/reference/odr.html> accessed on: 20.05.2022