International Journal of Modeling and Optimization, Vol. 8, No. 4, August 2018

Improvement of Synchronization Algorithms in Home Position for Robots with 5 Degrees of Mobility

Niculae Mihai

***Abstract*—This paper will present a method to improve spatial trajectories errors for end effectors of any robot using combined two solutions of home positions functions. The paper will present the magnitude of the errors in different situations, using different synchronization and zeroing systems and their influence on positioning precision of the end effectors on the spatial trajectory. There will also be presented the results of experimental research measured in the laboratory on the ceiling mounted robot at various points of the trajectory and the influence of positional errors on the calculation algorithms for all five degrees of mobility.**

***Index Terms*—Forward kinematics, ceiling mounted robot, source code, synchronization algorithms, combined "and" function.**

I. Introduction

The home position (homing) represents the reference position, zero, from which the motion algorithms begin to move the robot relative to this reference. Homing is a sequence of predefined motions that are normally required in order to configure the system's absolute position after power up. The homing errors of the industrial robots are caused by many factors: mechanical factors, electric and electronic factors, imprecise or inappropriate mathematical algorithms, programming errors, etc. [1]. This sequence is carried out by searching for an absolute known sensor along the mechanical travel, and updating the internal position accordingly. Once the absolute position has been registered in the drive, the system uses this point as homing position. For instance, incremental encoder is a kind of feedback that gives a relative position only and does not provide any information regarding the absolute position of the system [2].

Certain encoders, such as quadrature encoders, are incremental and do not provide absolute position. To use an incremental encoder in a position application, a known reference position must be established. As the encoder is rotated, it increments or decrements the position from the initial home position, and thereby the position is obtained [3].

Many homing methods, which are applicable to robot design, exist in academic literature. We present as examples three such positioning references robot methods before applying motion algorithms. First one is Elmo commands method who modify the sensor position counter, log the event position counter into array, flag a digital output and stop the axis motion. The second method is PLCopen who uses standard function block. The PLCopen are based upon the IEC1131-3 standard, the only global standard for industrial control programming and use function like HomeAbsSwitch - Absolute Switch homing plus limit switches or HomeRefPulseSet - Homing using a set of encoder reference pulses "Zero Mark" [4].

Another homing option implemented in intelligent servo drive interface is called DS402 who include 36 possibilities of capturing the main position sensor. This including "Touch probe homing". This method is defined by CANopen protocol for capturing the main position sensor. [4], [5]. Can support PLCopen Homing function block based on IEC 61131 language and reduce development time by offer "ready to use" function blocks.

In this paper we study the influence of electromechanical factors that create positioning errors on home position limiters, when homing the robot, before each cycle of motion.

Fig. 1. The experimental research stand including robot and travel space.

II. Methods and Materials

We study the case when the robot has to start moving; always from a fixed point of origin to each of the 5 axes of rotation, and from practical experiments we can see that this fixed point of origin is never perfectly fixed in reality in experiments revealing certain positional errors. The biggest errors are the electrical micro switches, which have the highest hysteresis, followed by the electromagnetic proximity limiters, then the optical limiters. Never these types of switches provide a perfect zero for each axis. Positioning errors vary between 20 and 200 micrometers, depending on the type of switch chosen. The experimental results have shown us that, at all times, the position of Home

Manuscript received May 18, 2018; revised June 19, 2018.

Niculae Mihai is with Technoaccord Inc., Laval, Quebec, Canada (e-mail: halimazerrouki519@gmail.com).

DOI: 10.7763/IJMO.2018.V8.653

is not the same on any degree of mobility. That is why, through this work, we want to observe, in what limits affect the positioning accuracy, these synchronization errors. Figure 1 shows the experimental stand, which includes the ceiling mounted robot, used for fast food preparation, equipped with various timing limiters, fixed to the mechanical structure in precise positions corresponding to the robot workspace. Movement commands are given from the robot controller at various points of travel on the spatial trajectory. Measurement of displacements in theoretical plane is done with relative incremental encodings, and error measurement is done with high precision measuring instruments, see Fig. 2.

Algorithms used for calculating displacements are specific to solving the inverse (IK) and forward kinematics problem (FK) by using 4x4 transformation matrices and using Denavit & Hartemberg parameters. [6].

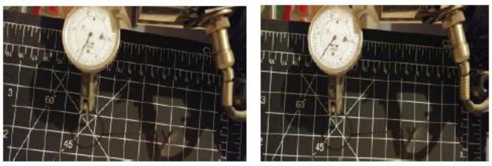

Fig. 2. High precision measuring instruments.

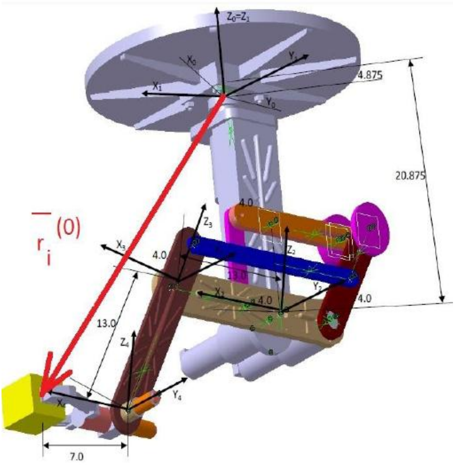

Fig. 3. The kinematic structure of the experimental robot 5DOF.

These representations and homogenous transformations based on orthonormal matrices given the end effector position with finally goal to obtain the desired spatial trajectory, theoretically with minimum errors [7].

III. The Mathematical Model of the Ceiling Mounted Robot

The mathematical model contents the vector-matrix equations established with the quaternion algebra, the compound of the spatial vectors, and the transfer matrix between some of the space plane.

The equation is:

$$(\bar{r}_i^{(0)}) = (\bar{r}_{i-1}^{(0)}) + [D_{i-1}^0] \cdot (\bar{r}_i^{(i-1)}) \tag{1}$$

where: (r_i^0),(r_{i-1}^0) are the matrices for the absolute position of *i* and *i*-1joint; $[D^0_{i-1}]$- the matrix for the transfer by rotation or translation of the vector from *i*-1 system to base system,*0*. The kinematic structure of the experimental robot is shown in Fig. 3 [7].

The coordinates of the characteristic point in relation to the base point (zero) are [4]:

$$r_5^0 = \begin{pmatrix} x_5^0 \\ y_5^0 \\ z_5^0 \end{pmatrix} \tag{2}$$

where:

x = 13c1*c2-13*(c1*c2*s3+c1*s2*c3)+7*(c4*(c1*c2*c3-c1*s2*s3)-s4*(c1*c2*s3+c1*s2*c3));
y=13*s1*c2-13*(s1*c2*s3+s1*s2*c3)+7*(c4*(s1*c2*c3-s1*s2*s3)-s4*(s1*c2*s3+s1*s2*c3));
z=-20.875-13*s2-13*(-s2*s3+c2*c3)+7*(c4*(-s2*c3-c2*s3) -s4*(-s2*s3+c2*c3)); (3)

For inverse kinematics we use the proper Neural Network with 4 layers, the delay block and recurrent links couplet with FK and IPIJMM algorithm (4). [8]

where *(T)*- column matrix of the target; (*FK*)- column matrix of the forward kinematics;

$[J(q_i)]^T\{[J(q_i)][J(q_i)]^T\}^{-1}$ -pseudo inverse matrix of the Jacobian; α - convergence step of the iteration.

The matrix model for the proper used Neural Network is presented in equations (5):

$$n_1 = [\underbrace{w^1}_{p_1} + \underbrace{tcg_1}_{p_2} \cdot \varepsilon_1](p - a_2(t - p_3 + 1)) + (b_1 + \varepsilon_1)$$

$$a_1 = \frac{p_4(1-e^{-n_1})}{1+e^{-n_1}}; \quad \varepsilon_1 = t_1 - a_1$$

$$(dq_i) = \alpha[(T_i) - (FK(q_i))][J(q_i)]^T\{[J(q_i)][J(q_i)]^T\}^{-1}$$
$$(q_j) = (q_i) + (dq_i) \tag{4}$$

$$n_2 = [w^2 + \underbrace{tcg_2}_{p5} \cdot \varepsilon_2](a_1(t - p_6 + 1)) + (b_2 + \varepsilon_2)$$

$$a_2 = \frac{p_7(1-e^{-n_2})}{1+e^{-n_2}}; \quad \varepsilon_2 = t_2 - a_2$$

$$q_i = p_8(a_2 - \varepsilon_f); \quad \varepsilon_{pos} = t_3 - r_i$$

$$n_3 = [w^3 + \underbrace{tcg_2}_{p_5} \cdot \varepsilon_{pos}](q_i) + (b_3 + \varepsilon_{pos})$$

$$a_3 = \frac{p_9(1-e^{-n_3})}{1+e^{-n_3}}; \quad \varepsilon_f = t_2 - a_3 \tag{5}$$

where a_i- matrix of output data of the sensitive sigmoid function; n_i- matrix of input in sensitive sigmoid function; t_i- matrix of target data of each layer; ε_i - matrix error after each layer; *t* – step delay; p_i- parameter of the neural network what can be changed for optimizing the results; w^i - weight matrix; b_i - biases matrix.

Movement algorithms were inserted into the robot controller for 10 spatial displacement points [8]. Following the calculations, the theoretical displacement values, which are shown in Table I and II.

TABLE I: THEORETICAL VALUES OF CHARACTERISTIC POINTS [P1-P5]

	HOME	P1	P2	P3	P4	P5
x	20.0000	18.5117	16.7908	14.8744	12.8054	10.6317
y	0.0000	0.9702	1.7648	2.3559	2.7219	2.8488
z	-33.8750	-35.5792	-37.1129	-38.4503	-39.5684	-40.4477

TABLE II: THEORETICAL VALUES OF CHARACTERISTIC POINTS [P6-P10]

	P6	P7	P8	P9	P10
x	8.4045	6.1764	3.9998	1.9254	0.0000
y	2.7308	2.3709	1.7808	0.9810	0.0000
z	-41.0726	-41.4317	-41.5187	-41.3319	-40.8750

TABLE III: ERRORS MEASURED EXPERIMENTALLY

	err fi1	err fi2	err fi3	err fi4	err fi5
Exp 1	-0.0001	0.0000	0.0000	0.0000	0.0000
Exp 2	0.0003	0.0004	0.0005	0.0007	0.0015
Exp 3	0.0000	0.0000	0.0000	0.0000	0.0001
Exp 4	0.0001	0.0001	0.0002	0.0002	0.0005
Exp 5	0.0005	0.0006	0.0009	0.0013	0.0026
Exp 6	0.0011	0.0014	0.0018	0.0027	0.0055
Exp 7	0.0002	0.0003	0.0004	0.0006	0.0012
Exp 8	-0.0001	-0.0001	-0.0002	-0.0002	-0.0005
Exp 9	-0.0002	-0.0002	-0.0003	-0.0005	-0.0010
Exp 10	-0.0003	-0.0003	-0.0004	-0.0006	-0.0013
Exp 11	-0.0001	-0.0002	-0.0002	-0.0003	-0.0007
Exp 12	-0.0008	-0.0010	-0.0013	-0.0020	-0.0040
Exp 13	-0.0004	-0.0005	-0.0007	-0.0011	-0.0022
Exp 14	-0.0002	-0.0002	-0.0003	-0.0004	-0.0008
Exp 15	0.0002	0.0002	0.0003	0.0005	0.0010
Exp 16	-0.0001	-0.0001	-0.0002	-0.0002	-0.0005
Exp 17	-0.0004	-0.0005	-0.0007	-0.0010	-0.0021
Exp 18	-0.0001	-0.0002	-0.0002	-0.0003	-0.0007
Exp 19	-0.0002	-0.0002	-0.0003	-0.0004	-0.0009
Exp 20	0.0002	0.0003	0.0004	0.0005	0.0011
Exp 21	-0.0001	-0.0001	-0.0001	-0.0002	-0.0004
Exp 22	0.0002	0.0003	0.0004	0.0005	0.0011
Exp 23	0.0004	0.0005	0.0006	0.0009	0.0019
Exp 24	0.0000	0.0000	0.0000	0.0000	0.0001
Exp 25	-0.0001	-0.0002	-0.0002	-0.0003	-0.0007
Exp 26	-0.0005	-0.0006	-0.0008	-0.0012	-0.0025
Exp 27	-0.0004	-0.0004	-0.0006	-0.0009	-0.0018
Exp 28	-0.0005	-0.0007	-0.0009	-0.0013	-0.0027
Exp 29	0.0002	0.0002	0.0003	0.0005	0.0010
Exp 30	0.0003	0.0004	0.0005	0.0007	0.0015

IV. EXPERIMENTAL RESULTS

In the experimental stand were mounted high precision measuring instruments, and experimentally measured the practical errors caused by Home position synchronization limiters. Hysteresis errors were considered and these errors were avoided by moving functions applied to the motion algorithms that generate these functions. Which consists of moving each of the 5 axes with high speed to these micro switches, stopping with inertia after touching them, but keeping the OFF condition of the limiter, then returning at very low speeds until they release again, generating the situation ON. In this way, hysteresis was avoided and we were able to measure only the errors caused by the inaccurate positioning on the limiter. We performed 30 experimental tests, three for each of the P1-P10 points on the spatial trajectory, and we centralized the results in the Table III.

We selected the displacements where the biggest positioning errors on the Home Limitations are highlighted, and we applied these errors to the internal robot coordinates for each axis. By repetitive application of FK and IK at each of the ten points of displacement, we obtained the errors of the characteristic points, which we centralized in the Table IV and V.

TABLE IV: PRACTICAL VALUES OF CHARACTERISTIC POINTS (P1-P5)

	HOME	P1	P2	P3	P4	P5
x err	20.0003	18.4981	16.7898	14.8686	12.7733	10.5607
y err	-0.0020	0.9750	1.7650	2.3565	2.7220	2.8422
z err	-33.8746	-35.5941	-37.1138	-38.4541	-39.5846	-40.4735

TABLE V: PRACTICAL VALUES OF CHARACTERISTIC POINTS (P6-P10)

	P6	P7	P8	P9	P10
x err	8.3887	6.1829	4.0126	1.9412	0.0079
y err	2.7279	2.3727	1.7856	0.9884	0.0045
z err	-41.0763	-41.4310	-41.5190	-41.3346	-40.8776

The obtained errors are shown in table 6 and 7.

TABLE VI: VALUES OF CHARACTERISTIC POINT ERRORS (P1-P5)

	HOME	P1	P2	P3	P4	P5
X-axis error	0.0003	-0.0136	-0.0010	-0.0057	-0.0321	-0.0710
Y-axis error	-0.0020	0.0049	0.0002	0.0006	0.0001	-0.0066
Z-axis error	0.0004	-0.0149	-0.0009	-0.0038	-0.0162	-0.0258

TABLE VII: VALUES OF CHARACTERISTIC POINT ERRORS (P6-P10)

	P6	P7	P8	P9	P10
X-axis error	-0.0158	0.0066	0.0128	0.0158	0.0079
Y-axis error	-0.0029	0.0018	0.0047	0.0074	0.0045
Z-axis error	-0.0037	0.0007	-0.0003	-0.0027	-0.0026

It is noticeable that the errors variation is high. The largest error is found on the x-axis at point P5 and is 0.071 inches. This value is inadmissible for a robot with 5 degrees of mobility. Therefore, it is necessary to find solutions for reducing the positioning errors on the spatial trajectory.

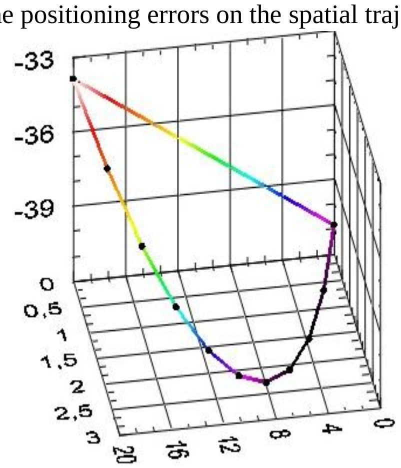

Fig. 4. 3D graphic of theoretical values of characteristic points.

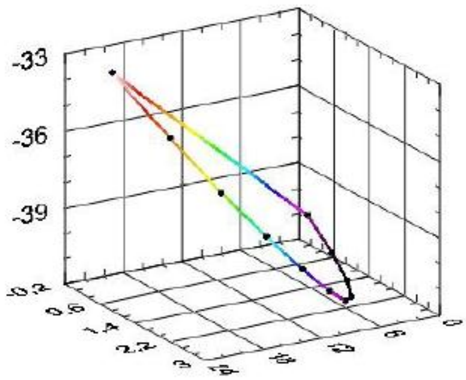

Fig. 5. 3D graphic of practical values of characteristic points

V. Solution of Sincronization by the Simultaneous Use of the Electric Limiter and of the Index Z from the Encoder

To reduce the positioning errors of the studied robot, we apply a combined synchronization method. Because the controller used for the robot control only accepts incremental encodings relative to two signals at 90 degrees A and A_, B and B_, we are obliged to design a separate electronic circuit for the acquisition of the Z signal of the encoder [9]. The signal thus amplified will connect to the "Z pulse" gate of the logic circuit "and", and the signal from the electric limiter will connect to the "Switch" gate of the logic circuit. See diagram shown in Fig. 6 [10].

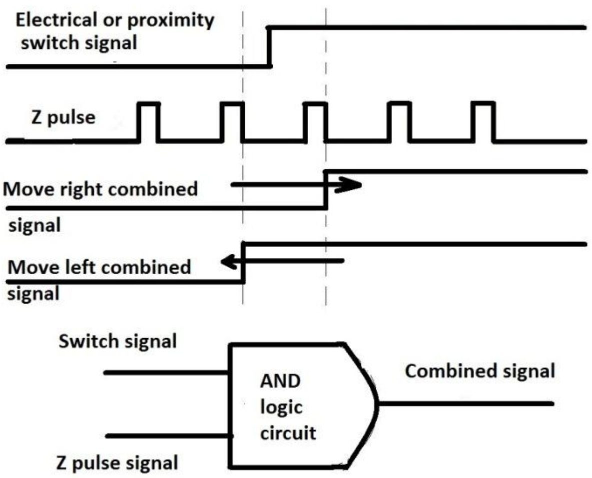

Fig. 6. The electrical scheme of switches and encoder signals.

Each axis of the robot studied has the Encoder-Servomotor-Reducer assembly. See Fig. 7.

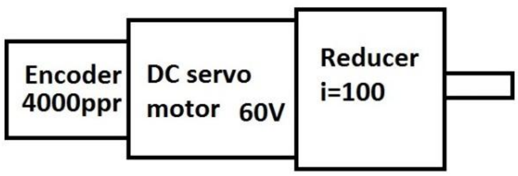

Fig. 7. The servomotor-reducer-encoder assembly used on each robot axis.

The reducer does not have a backlash and therefore the Z signal of the encoder gives us a good positioning accuracy. The encoder has on A and B signals 4,000 pulses per rotation. The Z signal is one pulse per revolution. Due to the gearbox, which has a transmission ratio of 100, we have a very good positioning accuracy given by the Z pulse. According to the diagram in Figure 6, the logic circuit receives an ON signal on the "Switch signal" input, but nothing happens on the output as long as no signal from the encoder is received. This signal gives ON the "Z pulse signal" input, which decides, at what point it will turn ON, the output signal of the logic circuit. In this way, on a 360/100 = 3.6 degree range, on the robot module, which is attached to the output shaft of the harmonic reducer, we have only one synchronization signal that is taken into account by the controller. This is the Z signal of the encoder. Other errors due to positioning on the electrical limiter are not considered as reference elements. This only determines the location where the robot module on the i (i = 1 ... 5) stops and starts the reverse motion to remove the hysteresis and establish the 3.6 degree place where the HOME synchronization will occur. Otherwise, on a fairly wide range, the errors of the electrical limiters are eliminated, and the HOME position is set by the Z pulse of the encoder with a very high accuracy.

```
#ifndef getabspos_h
#define getabspos_h
#ifdef __linux__
   #define __stdcall
   #define GETABSPOS_API
#else
   #ifdef GETABSPOSITION_EXPORTS
      #define GETABSPOS_API __declspec(dllexport)
   #else
      #define GETABSPOS_API __declspec(dllimport)
   #endif
#endif
#ifdef __cplusplus
extern "C" {
#endif
typedef int (__stdcall*
CncGetAbsolutePositionType)(double *x, double *y,
double*z, double *a, double *b, double *c);
int GETABSPOS_API __stdcall
GetAbsolutePosition(double *x, double *y, double*z,
double *a, double *b, double *c);
#ifdef __cplusplus
}
#endif
#endif //getabspos_h
```

Fig 8. GetABSposition.h function.

In order for this electronics circuit to work together with the controller used, a Dynamic-link library must be created in the Windows operating system [9].

In that case, we make a DLL with one exported function called GetAbsolutePosition with this interface: int __stdcall GetAbsolutePosition(double *x, double *y, double*z, double *a, double *b, double *c) . This DLL "GetAbsolutePosition()" with dummy implementation is presented in fig. 8. This function implements the communication with the each driver who control the servomotor for each axis and return the positions of all axes.

This DLL include a source code caled "dllmain.cpp " why defines the entry point for the DLL application [10]. See fig. no. 9.

```
#include "stdafx.h"

BOOL APIENTRY DllMain( HMODULE hModule,
                  DWORD  ul_reason_for_call,
                  LPVOID lpReserved
                    )
{
   switch (ul_reason_for_call)
   {
   case DLL_PROCESS_ATTACH:
   case DLL_THREAD_ATTACH:
   case DLL_THREAD_DETACH:
   case DLL_PROCESS_DETACH:
      break;
   }
   return TRUE;
}
```

Fig. 9. Defines the entry point for the DLL application.

The whole project is included several files like: "stdafx.cpp" which is the source file that includes just the standard includes,"GetABSPosition.pch" will be the pre-compiled header and "stdafx.obj" contain the pre-compiled type information. Other files are "GetAbsPosition.vcxproj" and " GetAbsPosition.vcxproj.filter". From Graphics User Interface we can command "Homing" for each axis independently or using F8 function we can homing all axes together.When homing is done, the controller gives "Home complete" message and sets the incremental encoders to zero for each axis.

VI. Conclusion and Future Work

This research work presented in this scientific paper allowed us to highlight errors in the robot positioning system, errors that negatively influenced the trajectorial positioning of the end-effector. Error detection and measurement were performed by using FK and IK algorithms obtained initially by computer simulation and then by robot command at predetermined spatial points. Error measurement was done with high precision electronic tools. We have then made an improvement of the electronic controller by adding a new electronic module that synchronizes the signals from the HOME position limiters together with those from the Z impulse encoder. This has resulted in a reduction of the trajectory positioning errors under 0.025 mm, which represents an improvement of approximately 72% at the sync level. This means that we can use the same incremental relative encoders as Homing with the same high efficiency, if we use them in combination with the limit switches. This also leads to a reduction in costs.

There remains to be done to improve the absolute position synchronization software, to improve the information processing functions from the servomotor driver, and to process the travel and trajectory control algorithms for the relative incremental encoders.

Acknowledgment

The author thanks the Canadian provincial and federal government for the financial support for this research work, as well as the industrial robot dynamics department of the IMST faculty of the "Politehnica" University of Bucarest.

References

[1] S. Kucuk and Z. Bingul, "Robot kinematics: Industrial robotics: Forward and inverse kinematics," *Industrial Robotics, Theory, Modelling and Control*, edited by Sam Cubero,Pro Literatur Verlag, 2006.

[2] Elmo motion control, *Servo Control Special Functionality*, [Online]. Available: http://www.elmomc.com

[3] A. Olaru and N. Mihai, "The dynamics of idustrial robots," Editura Bren, Bucarest 1999.

[4] Delta computers systems. RMC tools online helps – Motion controllers. [Online]. Available: http://www.deltamotion.com

[5] Vector, Solutions for your CANopen Networking.[Online]. Available: http://www.canopen-solutions.com

[6] J. Denavit and R. S. Hartenberg, "A kinematic notation for lowerpair mechanisms based on matrices," *Journal of Applied Mechanics*, vol. 1, pp. 215-221, 1955.

[7] A. Olaru, S. Olaru, and N.Mihai, "Proper assisted research method solving of the robots inverse kinematics problem," *Applied Mechanics and Materials*, vol. 555, 2014, pp.135-147.

[8] A. Olaru, E. Masehian, S. Olaru, and N. Mihai, "Achieving extreme precisions for multiple manipulators using a proper coupled neural network matrix method and LabVIEW instrumentation, " 2016.

[9] HEDS-9040, HEDS-9140, Three Channel Optical Incremental Encoder Modules, Catalog, Copyright © 2002 Agilent Technologies, Inc. Obsoletes 5988-2558EN February 11, 2002

[10] B. Eding, "Eding CNC software," open source interface.

Niculae Mihai was born in Alexandria, Romania, onFebruary 14, 1967. He has obtained Ph.D. in robotics at University "Politehnica" of Bucarest, Romania in 2000.

He is president of Technoaccord Inc. in Laval, Quebec, Canada since 2009. His personal activity is involve in the field of robotics, recherche scientific and experimental development with his company for mathematics althgoritms of kinematics and dynamics of industrial robots. He has published every years several paper with his scientific research.

Dr. Mihai work in colaborations with Prof. Univ. Ph.D. Eng. Adrian Olaru from the "Politehnica" University of Bucarest, Romania. He obtained two invention brevets in the robotics aplication fields. He is member of "Ordre des ingenieurs du Quebec" since 2005.