Proceedings of the 17th International Conference on Manufacturing Systems – ICMaS
ISSN 1842-3183

University POLITEHNICA of Bucharest, Machine and Manufacturing Systems Department
Bucharest, Romania

THE MONTE CARLO SIMULATION FOR THE PARALLEL ROBOTS

Cristina IBRAIM, Ramona Cristina VARBAN, Nicolae PREDINCEA

Abstract: *Monte Carlo simulation is a stochastic technique used to solve mathematical problems. The word "stochastic" means that it uses random numbers and probability statistics to obtain an answer. Monte Carlo methods have been used for centuries, but only in the past several decades has the technique gained the status of a full-fledged numerical method capable of addressing the most complex applications.*

Keywords*: parallel robot, Monte Carlo simulation, accuracy, Gauss distribution, workspace.*

1. INTRODUCTION

Numerical methods that are known as Monte Carlo methods can be loosely described as statistical simulation methods, where statistical simulation is defined in quite general terms to be any method that utilizes sequences of random numbers to perform the simulation. The name "Monte Carlo" was coined by Metropolis (inspired by Ulam's interest in poker) during the Manhattan Project of World War II, because of the similarity of statistical simulation to games of chance, and because the capital of Monaco was a centre for gambling and similar pursuits.

Monte Carlo is now used routinely in many diverse fields, from the simulation of complex physical phenomena such as radiation transport in the earth's atmosphere and the simulation of the esoteric sub nuclear processes in high energy physics experiments.

Statistical simulation methods may be contrasted to conventional numerical discretization methods, which typically are applied to ordinary or partial differential equations that describe some underlying physical or mathematical system. In many applications of Monte Carlo, the physical process is simulated directly, and there is no need to even write down the differential equations that describe the behaviour of the system. The only requirement is that the physical (or mathematical) system be described by probability density functions.

The essential characteristic of Monte Carlo is the use of random sampling techniques to arrive at a solution of the physical problem. In contrast, a conventional numerical solution approach would start with the mathematical model of the physical system, discretizing the differential equations and then solving a set of algebraic equations for the unknown state of the system.

It is natural to think that Monte Carlo methods are used to simulate random, or stochastic, processes, since these can be described. However, this coupling is actually too restrictive because many Monte Carlo applications have no apparent stochastic content, such as the evaluation of a definite integral or the inversion of a system of linear equations. As can be seen, the range of applications is enormous, from the simulation of galactic formation to quantum chromo dynamics to the solution of systems of linear equations.

1.1. Major components of a Monte Carlo Algorithm

The primary components of a Monte Carlo simulation method include the following:

- *Probability distribution functions (pdf's)* - the physical (or mathematical) system must be described by a set of pdf's.
- *Random number generator* - a source of random numbers uniformly distributed on the unit interval must be available.
- *Sampling rule* - a prescription for sampling from the specified pdf's, assuming the availability of random numbers on the unit interval, must be given.
- *Scoring (or tallying)* - the outcomes must be accumulated into overall tallies or scores for the quantities of interest.

1.2. Gauss Distribution

The normal (or Gaussian) distribution is one which appears in an incredible variety of statistical applications. A good reason for this is the central limit theorem. This theorem tells us that sums of random variables will, under the appropriate conditions, tend to be approximately normally distributed. Even when right conditions are not met however the distributions found for many experimentally generated sets of data still tend to have a bell shaped curve that often looks quite like that of a normal.

$$G(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (1)$$

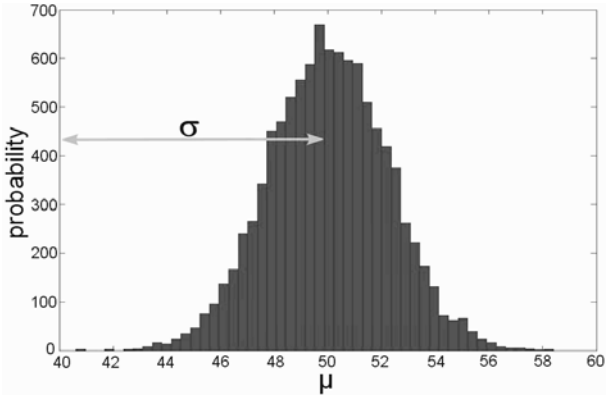


Fig. 1. Gauss Distribution.

The mean of the distribution is μ and the variance is σ. For example the distribution for $x = 50$ mm, the variance 10 mm and 10000 points.

The main task is generate a normally distributed random variables and to store them in a vector.

$X = 50$ Variance = 5 Number of random points = 100
function Gauss = GaussDistribution
$var = 5$; $nr\_points = 100$;
$y = 50 + \text{sqrt}(var) * \text{randn}(1, nr\_points)$;

53.7867	51.0226	52.3861	49.7147
50.4205	48.9865	48.6134	45.6948
46.8126	50.5993	49.1892	47.7995
53.0522	51.9386	49.0408	52.1730
47.4432	54.5787	50.0175	51.6698
51.2066	48.4531	52.4240	51.4073
48.3094	50.5340	47.3094	49.3108
46.5611	48.6131	49.5196	52.4194
49.6176	48.1881	49.9872	47.3996
48.5531	53.5157	50.2911	48.2016
49.7709	48.5072	48.4221	49.5078
48.9923	50.3149	53.5360	49.3358
49.2146	51.9511	51.8363	51.3356
50.0886	47.4811	51.5259	49.3920
50.3668	48.9416	51.0102	49.3939
48.7228	46.8324	47.6366	53.9504
54.7131	47.7136	49.7493	50.1406
49.0177	54.1032	51.2521	48.6138
47.2118	50.2128	50.1485	46.9316
51.5095	50.1267	52.9588	49.4801
51.1517	48.5421	50.0314	52.7122
53.1549	50.1935	55.6916	49.8082
52.8774	50.5538	49.2576	54.1237
50.6419	49.3980	49.5925	50.7316
49.9858	50.1439		

2. ABSOLUTE ACCURACY ANALYSIS

2.1. Find absolute accuracy of parallel robots with Monte Carlo Simulation

The Monte Carlo Simulation method is a combined numerical and statistical method for problem solving.

The problem has to be formulated as an independent random experiment with an input and an output.

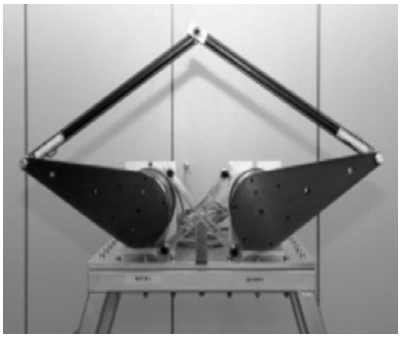
The input is a list of random numbers with a specific distribution.

The Monte Carlo simulation consists of many repetitions of the random experiment with a changing random number input list. The output of the experiment is also a list of values which is statistically evaluated.

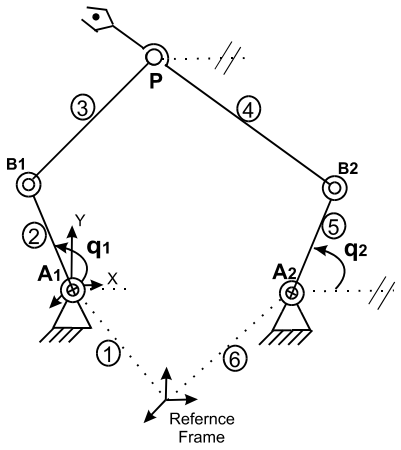
The basic idea is to perform the direct kinematical algorithm for a huge numbers of parameters variations.

The choice of the right distribution for parameters variation depends on the given task; in this case the variation will be for each parameter a Gauss Distributions. Of course, many other distributions can be used.

For the structure from Fig. 2 we have 6 independent parameters for each link that mean that we will have $6 \times 6 = 36$ vectors with normal distributed values.



a



b

Fig. 2. Analyzed structure of robot with five articulations: a – model; b – kinematic diagram (dof: 2 (x, y), workspace: 400×600 mm^2, speed: 4 m/s, pay load: 1 kg).

Table 1

Vectors with parameters variation

	A	**B**	**C**	α	β	γ
Link 1	A1	B1	C1	Alfa1	Beta1	Gama1
Link 2	A2	B2	C2	Alfa2	Beta2	Gama2
Link 3	A3	B3	C3	Alfa3	Beta3	Gama3
Link 4	A4	B4	C4	Alfa4	Beta4	Gama4
Link 5	A5	B5	C5	Alfa5	Beta5	Gama5
Link 6	A6	B6	C6	Alfa6	Beta6	Gama6

Table 2

Input parameters

Joint1 = Rotation Type = driving joint Teta = 90 degree A = -150 B = 0 C = 0 Alfa = 0 Beta = 0 Gama = 0	Joint2 = Rotation Type = driven joint Teta = 0 degree A = 300 B = 0 C = 0 Alfa = 0 Beta = 0 Gama = 0	Joint3 = Rotation (Cut joint) Type = driven joint Teta = 90 degree A = 500 B = 0 C = 0 Alfa = 0 Beta = 0 Gama = 0
Joint6 = Rotation Type = driving joint Teta = 90 degree A = 150 B = 0 C = 0 Alfa = 0 Beta = 0 Gama = 0	Joint5 = Rotation Type = Driven joint Teta = 0 degree A = 300 B = 0 C = 0 Alfa = 0 Beta = 0 Gama = 0	Joint4 = Rotation (Cut joint) Type = driven joint Teta = 90 degree A = 500 B = 0 C = 0 Alfa = 0 Beta = 0 Gama = 0

The next step is to chose and sends the parameters to CCD.

For example: the input values for parameters (Table 1) are presented in Table 2.

To find the absolute accuracy of parallel structure the algorithm for Monte Carlo Simulation can be computed by using the following steps: (Fig. 3)

Step 1 Read input data about the structure (input the desired values of the driving variables; assign arbitrary values to the driven variables)

Step 2 Based on input data and the distribution rules create 36 vectors with variation of the parameters. Once this has been completed then we have to define how the parameters will be chosen.

Step 3 Chose another random parameter combination then the parameters will be stored in the data structure, now we have another structure with another position for end effector.

Step 4 Using the recursion formulae and current value of parameters perform the Forward Coordinate Transformation for the new structure.

Step 5 Compute the error (we must have here a value for accepted error as a condition to get the solution)

Step 6 If we have an error smaller then accepted error we can consider that as a solution and we have to store this in a vector.

If the error is still greater then accepted error we will apply the minimisation.

Step 7 At the end we will have 2 vectors, one with deviation on x and one with deviation on y.

With these two vectors we will compute the next values:

Dxmed – medium deviation on x

Dxmax – maximum deviation on x

Dymed – medium deviation on y

Dymax – maximum deviation on y

Dmed – medium distance between nominal position and the position computed after parameter variation.

Dmax – maximum distance between nominal position and the position computed after parameter variation.

We can to count the number of steps needed to achieve the target point.

Step 8 If the number of points is enough we can print the results otherwise go to *Step 3*.

2.2 Absolute accuracy analysis over the entire workspace

Mesh Analysis over the entire workspace. The workspace analysis will be computed by using the following steps:

Step 0 Create a mesh surface for the workspace (the mesh surface will be in fact two matrices one with coordinates on x and another with coordinates on y)

Step 1 For each point from the mesh surface we need to compute the Inverse Kinematical algorithm (IKP) and the result will be q1 and q2.

For some points the solution for IKP will be complex that means that the point is out of workspace. Where the point is in workspace go to *Step 3*.

Step 2 Apply the algorithm for Monte Carlo Simulation [5]. This time we will store the values for Dmed, Dmax, Dxmed, Dxmax, Dymed, Dymax and Steps in another matrix. Now we have a meshed surface and we know the accuracy in each point from the workspace.

Step 3 Create a coloured surface based on accuracy matrix.

Is possible to create many graphics for example: medium or maximum accuracy, medium or maximum deviation on *x* for entire workspace, medium or maximum deviation on y for entire workspace, how many steps are needed to find the solution?

In Fig. 4 we can see easily how this algorithm works and some resuts.

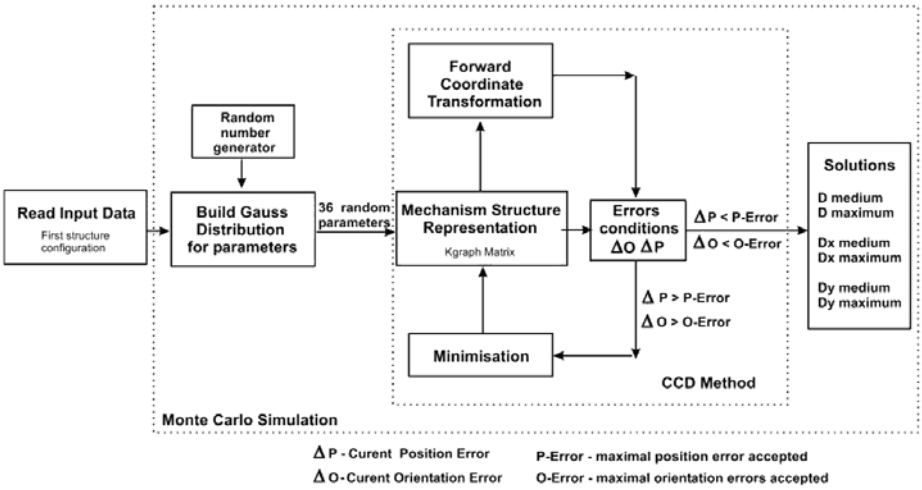


Fig. 3. Complete algorithm for Monte Carlo Simulation.

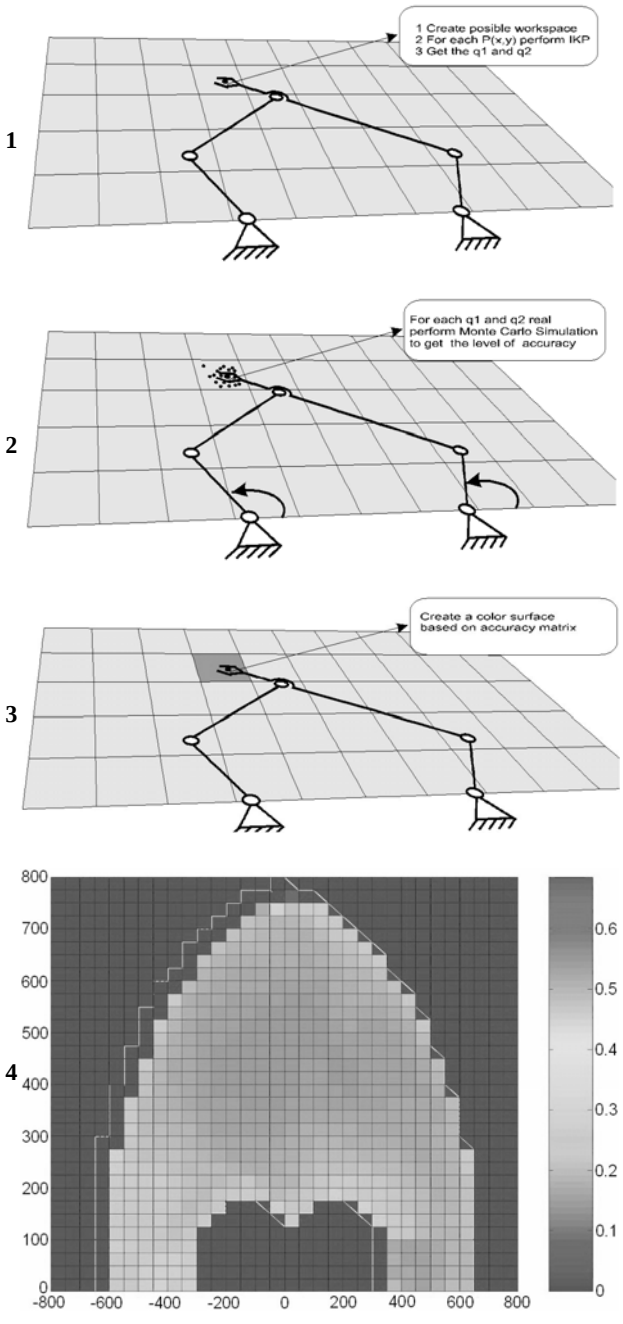


Fig. 4. Create workspace analysis.

3. CONCLUSIONS

Simulation shall be run, determining the maximum as well as the average values of the pose deviation. Running these simulations for a huge number of poses over the entire workspace the results may be used to determine parameters, which variation can be neglected during the parameter identification process thus speeding up the calibration procedure.

This article is intended to determine the influence of geometric parameter variation on the absolute accuracy of parallel robots.

REFERENCES

[1] Costa, M., Smaby, N., (1997). *Calibration and precision manufacturing*, ME319 Robotics and Vision Lab, pp. 3-5.

[2] Germer, C., Hansen, U., Franke, H. and Büttgenbach, S., (2003). *Development of a 3D-CAD add-in for tolerance analysis and synthesis in micro systems*, Symposion on Design, Test, Integration and Packing of MEMS/MOMS.

[3] Kevin, L., Conrad, Panayiotis S. Shiakolas, (2000). *Robotic calibration issues: Accuracy, Repeatability and Calibration*, Mediterranean Conference on Control & Automation GREECE.

[4] Li-Chun, Wang, Tommy, Cheng Chen, Chih, (1993). *On the Numerical Kinematic Analysis of general Parallel Robotic Manipulators*, IEEE Trans. Robotics Automat.

[5] Nae, L., Ibraim, C., Olaru, A., Predincea, N., (2007). *Construction and direct kinematics model of parallel robots*, Proceedings of the 16 th International Conference on Manufacturing Systems - ICMaS.

Authors:

Eng., Cristina IBRAIM, Concepteur CAO, Renault Technologie Roumanie S.R.L.,
E-mail: cristina_ibraim@yahoo.com
Eng., Ramona Cristina VARBAN, PhD Student, "Politehnica" University of Bucharest, Faculty of Engineering and Management of Technological Systems, Machines and Production Systems Department,
E-mail: crissram@yahoo.com
Ph.D. Eng., Nicolae PREDINCEA, Professor, "Politehnica" University of Bucharest, Faculty of Engineering and Management of Technological Systems, Machines and Production Systems Department,
E-mail: nicolaepredincea@yahoo.c