

Compliance errors estimation for robotic manipulator using the Quantum Monte Carlo method *

Ramil Dautov
Innopolis University
Innopolis, Russia
r.dautov@innopolis.ru

Alexandr Klimchik
Innopolis University
Innopolis, Russia
a.klimchik@innopolis.ru

Abstract

The paper presents new approach to estimating the compliance errors of robotic manipulator. The existing methods are complicated with the non-linear character of equations coming from the theory. Quantum Monte Carlo computational technique is shown to overcome this problem. The algorithm of evaluating and compensating compliance errors using this technique is presented.

1 Introduction

The precision of industrial robots plays an essential role in manufacturing and high-quality machining. There are many factors that influence on the accuracy of robot's performance. The essential part of deviations originates from the interactions between the robot and a workpiece (loading). These interactions cause the deformations of manipulator components that lead to errors in performance (compliance errors).

Compliance errors are in high dependence on the configuration of the robot and this fact complicates their analysis. Different models have been developed to estimate and compensate the compliance errors [1, 2, 3]. The theory appears to be non-linear and the problem is usually resolved by using the approximations and implementing the computational schemes to solve equations.

However, there exists one computational method that provides the possibility to avoid such complexities and consider the theory in its original form – Quantum Monte Carlo scheme [4, 5].

Quantum Monte Carlo method is broadly used in condensed matter physics and physics of elementary particles. This technique allows us to reduce the problem of solving the highly non-linear equations to the problem that is common in statistical physics.

The main role here plays the action functional of the system. The equations of motion are solved indirectly by the help of statistical computational algorithms that find the solution by minimizing the action.

In this article we propose the possible implementation of Quantum Monte Carlo technique to the problem of compensation the compliance errors. The article is structured as follows. We begin with the brief review of the Quantum Monte Carlo method in section 2. Our review is based on the two well-developed introductory courses on implementations of the Quantum Monte Carlo technique [4, 5]. In Section 3 we show how this technique can be applied to evaluation and compensation the compliance errors. The corresponding algorithm is also presented.

* Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2 Quantum Monte Carlo method

In quantum physics we have a well-developed formalism that generalizes the action principle of classical mechanics. This formalism is called *Path Integral formulation* of quantum mechanics and was proposed by Richard Feynman in 1948 [6]. Path integrals replace the notion of classical trajectory with the functional integral over an infinite number of mechanically possible trajectories. These integrals are used to obtain the value of particular parameters of the system.

The path integrals reveal the correspondence between quantum and stochastic processes and also can be considered as an analytical continuation of a method of random walks. This property allows to transform path integral theory to the form analogous to the theory of statistical physics and molecular dynamics.

2.1 Path Integrals

Path integrals provide us possibility to calculate the value of a particular parameter of the system. Let us denote the parameter that we want to evaluate as A . In general, $A(x)$ is a function of coordinates. Then we can find the average value of $A(x)$ by following formula:

$$\langle A(x) \rangle = \frac{\int Dx(t) A(x) e^{\frac{i}{\hbar} S(x)}}{\int Dx(t) e^{\frac{i}{\hbar} S(x)}} \quad (1)$$

Here i – is an imaginary unit, \hbar - is a Planck constant (usually quantum physicists work in a system of units with $\hbar = 1$), $S(x) = \int_{t_i}^{t_f} dt L(x, \dot{x})$ – is an action functional. $\int Dx(t)$ designates the sum over all possible particle paths from the initial time t_i to the final t_f .

Although there are some theoretical advantages of formula (1), it is difficult to calculate the value of $A(x)$ analytically. To overcome this obstacle there has been developed a numerical method based on the Monte Carlo scheme. It has been already pointed out that path integral formulation reveals the correspondence between the quantum physics and statistical physics. To show this correspondence explicitly let us replace the time variable: $t \rightarrow it$ – so-called Wick's rotation (it is actually rotation in an imaginary plane). After Wick's rotation, considering $\hbar = 1$ we receive 'Euclidian' path integrals:

$$\langle A(x) \rangle = \frac{\int Dx(t) A(x) e^{-S(x)}}{\int Dx(t) e^{-S(x)}} \quad (2)$$

Euclidian path integrals look very similar to the integrals of thermal averages that we have in Thermodynamics and Statistical physics. This representation is much better for numerical work as it has no oscillations in sign within the powers of exponents.

It is important to take into account the changes in the action $S(x)$ caused by the Wick's rotation. For instance, let us consider the action of point-like particle with mass m . Before the Wick's rotation the action has the form:

$$S(x) = \int_{t_i}^{t_f} dt \left(\frac{m\dot{x}^2}{2} - U(x) \right) \quad (3)$$

After the Wick's rotation $t \rightarrow it$ we will have:

$$S(x) = \int_{t_i}^{t_f} dt \left(\frac{m\dot{x}^2}{2} + U(x) \right) \quad (4)$$

We see that Lagrangian $L(x, \dot{x})$ has changed its form and became actually the Hamiltonian of the system. And continuing the analogy between Euclidian path integrals and thermodynamics one can easily show that the time t now has relation with the ‘temperature’ T [4, 5]:

$$t \rightarrow \beta \equiv \frac{1}{k_B T} \quad (5)$$

Thus any code developed to conduct calculations in thermal physics can be applied for calculation path integrals (2).

2.2 Quantum Monte Carlo algorithm

Quantum Monte Carlo technique provides a numerical procedure for calculating the path integrals using the stochastic methods developed in statistical physics. To reduce our continuous formula (2) into the numerical discretized form we must address two issues.

Firstly, we need to find a convenient representation of an arbitrary path $\{x(t), t_i \leq t \leq t_f\}$ in the computer. A path is given by the $x(t)$ function which, in principle, can be complicated enough for computers. We approximate the path by specifying $x(t)$ only at the nodes on a discretized time axis:

$$t_j = t_i + ja, \quad \text{for } j = 0, 1, \dots, N \quad (6)$$

Here a is a grid spacing:

$$a = \frac{t_f - t_i}{N} \quad (7)$$

Then the path can be described by a vector:

$$x = \{x(t_0), x(t_1), \dots, x(t_N)\} \quad (8)$$

In lattice theories this vector is usually called ‘configuration’. The integral over all paths becomes an ordinary multidimensional integral over all possible values for each of the $x(t_j)$ ’s:

$$\int Dx(t) \rightarrow \int_{-\infty}^{+\infty} dx_1 dx_2 \dots dx_{N-1} \quad (9)$$

Here we denoted $x(t_j) = x_j$. We don’t integrate over endpoints x_0 and x_N since they are held fixed.

The second issue is related to the evaluation of the action on the discretized path (8). In fact, here we just need to substitute the derivatives in the action with their finite differences approximations. Integrals are replaced with the summation and the potential $U(x)$ should be calculated in each node $U(x_j)$. For instance, the discretized version of the action (4):

$$S_{lat}(x) = \sum_{j=0}^{N-1} \left[\frac{m}{2a} (x_{j+1} - x_j)^2 + aU(x_j) \right] \quad (10)$$

We have reduced mechanics to the problem of numerical integration. Now we can evaluate the path integrals using the standard methods of numerical multidimensional integration. The Monte Carlo procedure appears to be the more general and convenient technique for this purpose.

One can note that path integral of the form (2) is actually a weighted average over paths with weight $\exp(-S(x))$. Taking into account this fact we generate a large number N_{cf} of random paths (or configurations)

$$x^{(\alpha)} \equiv \{x_0^{(\alpha)}, x_1^{(\alpha)}, \dots, x_{N-1}^{(\alpha)}\} \quad \alpha = 1, \dots, N_{cf}, \quad (11)$$

on our grid in such a way that the probability $P[x^{(\alpha)}]$ for obtaining the particular path $x^{(\alpha)}$ is:

$$P[x^{(\alpha)}] \sim e^{-S[x^{(\alpha)}]} \quad (12)$$

Then unweighted average of $A(x)$ over this set of paths approximates the weighted average over uniformly distributed paths:

$$\langle A(x) \rangle \approx \bar{A} = \frac{1}{N_{cf}} \sum_{\alpha=1}^{N_{cf}} A[x^{(\alpha)}] \quad (13)$$

\bar{A} is the Monte Carlo estimation for $\langle A(x) \rangle$ on our lattice. The estimation will never be the exact since we use the discretized paths and N_{cf} will never be infinite. One can show that statistical errors will be of the order $\frac{1}{\sqrt{N_{cf}}}$ [4, 5].

There are many ways of how to generate the set of random paths $x^{(\alpha)}$ [4, 5]. We review only the main idea of these methods. For the detailed review the reader can refer to the articles [4, 5].

So, the general idea of the methods is the following:

- first, generate an arbitrary path $x^{(0)} = \{x_0^{(0)}, \dots, x_j^{(0)}, \dots, x_{N-1}^{(0)}\}$;
- at each j^{th} site we generate the random number Ω , with probability uniformly distributed between $-\epsilon$ and ϵ , $\epsilon -$ is a some constant;
- replace $x_j^{(0)} \rightarrow x_j^{(0)} + \Omega$ and calculate the change ΔS in the action caused by this replacement;
- if $\Delta S < 0$ then retain the new value for $x_j^{(0)}$;
- if $\Delta S > 0$ generate a random number σ uniformly distributed between 0 and 1; retain the new value for $x_j^{(0)}$ if $\exp(-\Delta S) > \sigma$, otherwise restore the old value;

This algorithm is repeated some number of times and at the end we obtain the path that minimizes the action and can be considered as the solution to the equations of motion. The constant ϵ is tuned experimentally by running the algorithm so that 40-60% of new x_j 's are retained.

3 Compliance errors estimation

The main problem of estimating the compliance errors is the necessity to deal with complicated non-linear equations of motion. In Quantum Monte Carlo method we have no need to work with the exact equations of motion. All the dynamical and kinematical information of the system is already included into the Lagrangian function. Using the Quantum Monte Carlo technique we can minimize the action and obtain the configuration of the system that represents the solution of the equations of motion.

To estimate the compliance errors, we must take into account two components. The first component represents the stiffness properties of the robot. These properties are defined by the stiffness matrix. In this work we assume that the stiffness matrix of the system is known. The stiffness matrix can be found by implementing the CAD-based approach [7].

The second component represents the forces/torques acting on the end-effector. Further, we also assume that this component is given. For instance, it can be obtained by direct measurements using the force/torque sensor adjusted on the end-effector.

Compliance errors occur due to the interaction between a workpiece and the end-effector. The control system of the robot knows the configuration of the manipulator at the given time and manages its movements to make the end-effector

follow the desired trajectory. However, because of the compliance the deflections appear and the end-effector follows another trajectory (*real* trajectory) that differs from the desired one.

The motion of the end-effector can be described by the Lagrange equations of the second kind:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = Q_j, \quad j = 1, \dots, N \quad (14)$$

N – the number of degrees of freedom, Q_j – the generalized force acting on the end-effector along the direction of the q_j coordinate, $L = T(q, \dot{q}) - U(q)$ – Lagrangian of the system. $T(q, \dot{q})$ – kinetic energy and $U(q)$ – potential energy of the system that contains contributions of a gravitational field and potential energy of elastic deformations caused by compliance.

In principle, we can decompose the movement of the end-effector into the sequence of quasistatic translations: within a sufficiently small period of time Q_j can be considered as constants and during that time the motion of the end-effector effectively is ruled by the Lagrangian of the form:

$$L = T(q, \dot{q}) - U(q) + \sum_{j=1}^N Q_j q_j \quad (15)$$

We see that the equation (14) can be derived from the (15) assuming the absence of non-conservative forces. Actually, to use the Quantum Monte Carlo technique we have to rewrite the Lagrangian (15) in a Euclidian way (compare with (4)):

$$L = T(q, \dot{q}) + U(q) + \sum_{j=1}^N Q_j q_j \quad (16)$$

Then within that small period of time we can use the Lagrangian (16) to write down the Euclidian action functional and using the Quantum Monte Carlo technique minimize it to obtain the real configuration $q = \{q_j\}$ of the robot. Then the obtained real configuration is compared with trajectory stored in the control system. The compliance errors are estimated as the differences between the real and desired (stored in the control system) trajectories. After evaluating the compliance errors, the control system is able to adjust the trajectory of the end-effector to eliminate the deflections.

The main difficulty that can arise here is the time consumption of the algorithm. The external forces acting on the end-effector can vary with high rate. The Monte Carlo algorithm takes some time to find the configuration minimizing the action. By the time the configuration is found the end effector can experience another external force Q'_j , and our obtained configuration $q = \{q_j\}$ becomes outdated.

In fact, we can shorten the time that is needed for the Monte Carlo algorithm to minimize the action by taking as a starting configuration $q^{(0)}$ in the Monte Carlo algorithm the desired configuration (just read it from the control system). On practice, the deviations of the desired trajectory from the real trajectory are not so large and taking the desired configuration as an initial will put the action into position near the minimum. So, a few iterations will be needed to minimize the action and find the real trajectory. This feature may help us to adjust the trajectory of end-effector in the on-line regime.

The whole algorithm is represented on the Figure 1.

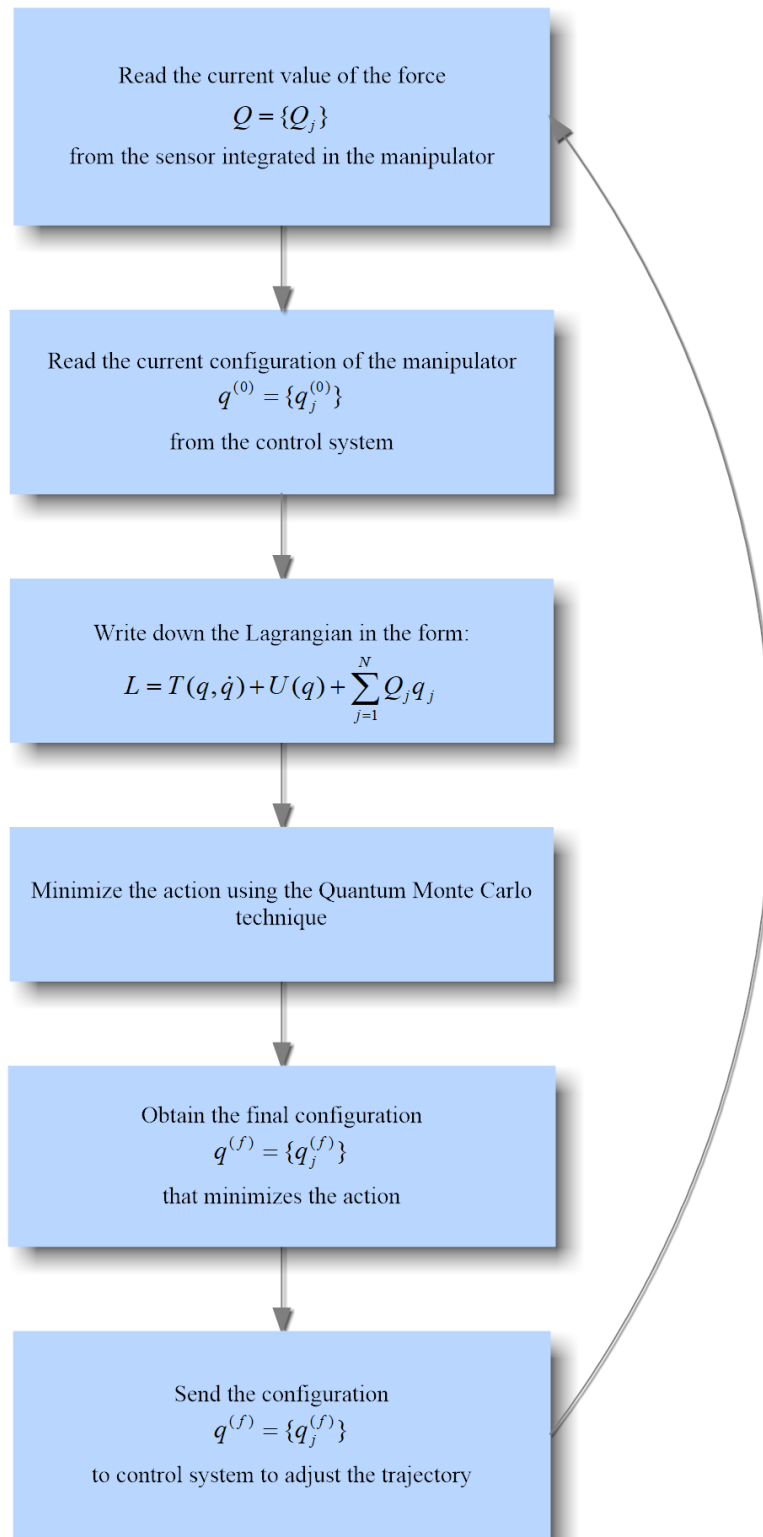


Figure 1: The algorithm for estimation and compensation the compliance errors

Summary

Compliance errors appear due to the interactions between the manipulator and the workpiece and they are in high dependence on the configuration of the robot. This fact makes the analysis and estimation of the errors problematic. We need to take into account the non-linear character of the compliance mechanism that leads us to the complicated math and implementation of different approximations.

Quantum Monte Carlo method has been proposed to overcome these issues. The developed algorithm provides an elegant way to find the real trajectory of the manipulator under the applied loadings. The obtained data then can be used by the control system to adjust the trajectory and compensate the compliance errors. However, Quantum Monte Carlo iterations can take considerable amount of time and the deeper analysis is required.

In future, the developed technique will be tested on the simulation of the compliance errors compensation of the Stewart's platform. Testing the Quantum Monte Carlo approach on that toy model will show the prospects of this method.

References

1. A. Klimchik, A. Pashkevich, D. Chablat, G. Hovland, Compliance error compensation technique for parallel robots composed of non-perfect serial chains, *Robotics and Computer-Integrated Manufacturing* 29, 2 (2012) 385-393.
2. S. Mamedov, D. Popov, S. Mikhel, A. Klimchik, Compliance Error Compensation based on Reduced Model for Industrial Robots ICINCO (2), 2018, 190-201.
3. A. Olabi, M. Damak, R. Béarée, O. Gibaru, S. Leleu. Improving the Accuracy of Industrial Robots by offline Compensation of Joints Errors. *IEEE International Conference on Industrial Technology*, 2012.
4. G. Peter Lepage, *Lattice QCD for Novices*, Proceedings of HUGS 98, edited by J.L. Goity, World Scientific (2000)
5. C. Gattringer, C.B. Lang, *Quantum Chromodynamics on the Lattice: An Introductory Presentation*, Lect. Notes Phys. 788 (Springer, Berlin Heidelberg 2010).
6. R. P. Feynman, Space-Time Approach to Non-Relativistic Quantum Mechanics, *Reviews of Modern Physics*. 20 (2): (1948) 367–387.
7. A. Klimchik, A. Pashkevich, D. Chablat, CAD-based approach for identification of elasto-static parameters of robotic manipulators, *Finite Elements in Analysis and Design* 75 (2013) 19-30.