Inverse dynamics modeling of robots based on sparse spectral gaussian process regression

Cuiyi Huang1*, Yang Wang1

1College of Information Engineering, Wuhan University of Technology, Wuhan, Hubei, 430070, China

*Corresponding author’s e-mail: 790947654@qq.com

Abstract. In robot control systems, inverse dynamics calculation of the robotic arm is essential. In this paper, we propose a different inverse dynamics modeling method—sparse spectral Gaussian process regression (SSGPR). Since using Gaussian process regression for robot inverse dynamics modeling is computationally intensive, to address this drawback, it is proposed to apply sparse spectral Gaussian process regression on robot inverse dynamics modeling, the core of which is the sparsification of the GP's spectrum for the purpose of improving computational efficiency. Experiments show that sparse spectral Gaussian process regression can improve the computational efficiency and outperform other regression methods while ensuring the computational accuracy.

1. Introduction

In recent years, machine learning has been widely used, and more and more attention has been paid to robot control. The inverse dynamics of manipulator arm is one of the main fields. Inverse dynamics refers to finding the moments applied to the joints of a robot at a given moment when the positions of the joints, joint velocities and joint accelerations are known. Well-known methods such as Lagrangian-Eulerian and Newton-Eulerian methods [1, 2, 3, 4] are commonly used to build the robot dynamics model. These methods are numerical recursive methods that require a large amount of computational time to obtain a solution.

Based on the powerful approximation capability of neural networks in solving problems in nonlinear mapping, many scholars have devoted themselves to the study of building inverse dynamics models with neural networks. Nan Liu et al. [5] used a deep learning neural network based on LSTM (Long Short-Term Memory) to predict the inverse dynamics of a manipulator. Misaki et al. [6] have reformulated the inverse dynamics model based on recurrent neural network (RNN) which trained RNN with random trajectories based on a bandlimited m-sequence. Siqi et al. [7] proposed an active trajectory generation framework to systematically design information trajectories to train deep neural networks (DNN) inverse dynamics modules that significantly improves the training of DNN inverse dynamics module with data efficiency. Different from the traditional methods, Steffan et al. [8] proposed a new method, Pseudo-Symbolic Dynamic Modeling (PSDM) for deriving closed equations of motion for a tandem chain of motion using basic inertial parameters. The results show that PSDM is a feasible algorithm for small and medium-sized robots. Rain et al. [9] constructed an input-output mapping of a robot inverse dynamics model using an adaptive-network-based fuzzy inference system (ANFIS). However, the online training of this method requires sensor interfaces to collect joint information and a large amount of training time.
Gaussian process regression [10] is a powerful nonparametric regression method based on Bayesian theory, which is gradually becoming a common tool for learning inverse dynamics models of robots. Junghoon et al. [11] proposed an inverse dynamics design method based on Gaussian process regression and verified the accuracy of Gaussian process inverse dynamics learning. Vijayakumar et al. [12] proposed local weighted projection regression theory (LWPR) which can achieve high prediction accuracy and reduce computational cost. In [13] Nguyen-Tuong et al. introduced the local Gaussian process regression (LGP). This method combines LWPR and Gaussian process regression (GPR) to provide high processing performance for high-dimensional data.

This paper presents a method for inverse dynamics modeling of robots based on sparse spectrum Gaussian process regression. Lázaro et al. [14] proposed a sparse spectrum Gaussian process regression method that, instead of offsetting the increase in kernel expansion as is usually done in related methods, achieves computational efficiency by sparsifying the spectrum of the GP using finite-dimensional feature mapping. In this paper, this sparse spectral Gaussian process regression method is applied to robot inverse dynamics modeling and experiments are conducted. The experimental results verify the accuracy of SSGPR inverse dynamics learning and this method improve the computational efficiency of GPR.

2. Robotics inverse dynamics

The inverse dynamics equations of a robot arm relate the joint position \( q \), velocity \( \dot{q} \) and acceleration \( \ddot{q} \) to the corresponding forces \( F \) and moments \( \tau \) operating on the arm. Accurate inverse dynamics models are important to achieve flexible control or to detect external forces applied to the manipulator. The inverse dynamics can be described analytically by the well-known rigid body dynamics (RBD) formulation

\[
\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \varepsilon(q, \dot{q}, \ddot{q})
\]

where \( D \) denotes the inertia matrix, \( C \) and \( g \) are Coriolis force terms and gravity terms, respectively, and \( \varepsilon \) denotes uncertainty such as nonlinear friction. In order to control the robot, for example, using feedforward control, to pass the generalized forces we compute to the motor controller, which can significantly improve the accuracy and responsiveness of the control. Typically, the motor commands are as follows

\[
\tau = \tau_{FF} + \tau_{FB}
\]

where \( \tau_{FF} \) is the feedforward component and \( \tau_{FB} \) is the feedback component. In a general linear feedback controller (PD), the feedback term \( \tau_{FB} \) is generally expressed as

\[
\tau_{FB} = K_p e + K_v \dot{e}
\]

where \( e = q_d - q \) is the tracking error and \( K_p \) and \( K_v \) are the feedback gains. Choosing the correct gain can be seen as a trade-off between tracking the desired trajectory with the inverse dynamics model and keeping the system stable.

When \( \varepsilon(q, \dot{q}, \ddot{q}) \) is neglected, there are various ways to derive an inverse dynamics model from the above equations of motion, the most common being the Lagrangian and Newton-Euler formulations derived from rigid body dynamics. However, the application of these methods may lead to inaccurate inverse dynamics models because many robot parameters are usually not known in advance and inertial parameters such as mass, center of mass, and inertia tensor matrix must be assumed. Furthermore, in utilizing rigid body dynamics is the effect of nonlinear side gaps and friction are neglected, resulting in feedforward commands that are only rough approximations of the equations of motion

\[
\tau_{FF} = H(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + g(q_d)
\]

thus, it can lead to poor torque prediction and result in inaccurate robot motion. Gaussian process regression has higher accuracy than traditional RBD models. However, robot control requires processing a large amount of high-dimensional data and sampling frequencies up to 100 Hz. As the number of data...
points increases, their computational cost can become very large. Therefore, it is not feasible to simply use Gaussian process regression techniques, and in order to make progress in this direction, it is crucial to develop Gaussian process regression methods suitable for the robotics domain.

3. Gaussian process regression

GPR is a nonparametric model for regression analysis of data using a Gaussian process prior, and the essence of its solution is Bayesian inference. Assuming that the output can be represented by a linear model

\[ y = f(x) + \epsilon = w^T x + \epsilon \]

where \( w \) is the weight vector and \( \epsilon \) is the noise of the disturbance. Assume that \( \epsilon \) obeys the Gaussian distribution \( \epsilon \sim N(0, \sigma^2) \). If we already know the deterministic value of \( w \), given \( n \) input samples \( X = [x_1, ..., x_n]^T \), we can get the likelihood function of \( n \) output samples \( Y = [y_1, ..., y_n]^T \)

\[ P(y|w, X) = N(Xw, \sigma^2 I) \]

(6)

Since we do not know what the distribution of \( w \) is, we can assume that the prior of \( w \) is \( w \sim N(0, \Sigma_w) \). In the Bayesian framework, according to the posterior distribution of the observed values \( X \) and \( y \) on the weight vector \( w \), and by applying Bayesian prior and likelihood rules, the posterior of \( w \) is obtained as

\[ P(w|X, y) = N(A^{-1}X^Ty, \sigma^2 A^{-1}) \]

(7)

where \( A = X^TX + \sigma^2 \Sigma_p^{-1} \). Combining the linear relation \( y = f(x) + \epsilon = w^T x + \epsilon \), the value of the function of the prediction point \( x \) is

\[ P(y|x, X, y) = N(x^T A^{-1}X^Ty, \sigma^2 (1 + x^TA^{-1}x)) \]

(8)

where \( x \) is the test set and \( X \) is the training set. Generalizing Bayesian linear regression to nonlinear problems by replacing \( x \) with a nonlinear feature map \( \phi(x) \), Eq. (8) can be changed to

\[ P(y|x, X, y) = N(\phi(x)^T A^{-1} \phi(X)^Ty, \sigma^2 (1 + \phi(x)^T A^{-1} \phi(x))) \]

(9)

where \( A = \phi(X)^T \phi(X) + \sigma^2 \Sigma_p^{-1} \). Define the kernel function as

\[ k(x_i, x_j) = \phi(x_i)^T \Sigma_p \phi(x_j) \]

(10)

where \( k_* = k(x, X) \), \( G = k(X, X) + \sigma^2 I \). Then we can replace equation (9) to the kernel function form that we want.

But \( \Sigma \) is an \( n \times n \) matrix, and it takes longer time to compute the inverse matrix of \( \Sigma \). The time complexity of training GPR is \( O(n^3) \), and in prediction, the time complexity of the predicted mean is \( O(n) \) and the time complexity of the predicted variance is \( O(n^2) \). Since the time complexity of GPR applied to large-scale data sets is too large, we propose sparse spectral Gaussian process regression.

4. Sparse spectral gaussian process regression

GPR uses Gaussian process as a priori, it assumes that the learning samples are sampled by a Gaussian process, so its estimation results are closely related to the kernel function. The practical meaning of the kernel function in GPR is the covariance function, which describes the correlation between the learning samples, and the kernel function formula we usually use is the squared exponential covariance formula

\[ k(x_i, x_j) = \sigma_f^2 \exp \left( -\frac{1}{2} (x_i - x_j)^T L (x_i - x_j) \right) \]

(11)

where \( L = \text{diag}(l^{-2}) \), \( l \) is the length scales and \( \sigma_f^2 \) is the signal variance, because

\[ k(x_i, x_j) = k(x_i - x_j) = k(r) \]

(12)

It is stated by the Wiener-Sinchin theorem that the power spectral density of a stationary random process and its autocorrelation function are a pair of Fourier pairs.
According to Bochner's theorem, any stationary covariance function \( k(\tau) \) can be expressed as a Fourier transform of positive finite measure [15]. This means that \( S(\omega) \) in Eq. (14) is a positive finite measure, and in particular, \( S(\omega) \) is proportional to the probability density function \( p(\omega) \)

\[
S(\omega) = \frac{2\pi}{\sigma^2} p(\omega) \quad (15)
\]

Then equation (13) can be expressed as

\[
k(x_i, x_j) = \sigma^2 \int_{R^d} e^{i(x_i - x_j) \omega^T} p(\omega) d\omega \quad (16)
\]

Equation (16) can be viewed as the expectation of \( e^{i(x_i - x_j) \omega^T} \), approximated by the Monte Carlo method

\[
k(x_i, x_j) = \sigma^2 E \left( e^{i(x_i - x_j) \omega^T} \right) \\
\approx \frac{\sigma^2}{M} \sum_{m=1}^{M} e^{i(x_i - x_j) \omega_m^T} \\
= \frac{\sigma^2}{M} \sum_{m=1}^{M} e^{ix_i \omega_m^T} (e^{ix_j \omega_m^T})^* \quad (17)
\]

According to Euler’s formula \( e^{ix} = \cos x + isinx \) we can get

\[
k(x_i, x_j) = \frac{\sigma^2}{M} \sum_{m=1}^{M} \cos \left( \omega_m^T (x_i - x_j) \right) \quad (18)
\]

where \( \omega_1, ..., \omega_M \) are sampled from the distribution of the probability density function \( p \) and \( M \) is the number of spectral points. The equivalent of Eq. (18) can be regarded as a sparse GP, which approximates the full stationary GP by replacing the complete spectrum with a set of discrete spectral points. If the number of spectral points tends to infinity, the SSGPR will infinitely approximate the full GPR. According to Eq. (10), \( \phi(x) \) can be expressed as

\[
\phi(x_i) = \frac{\sigma^2}{\sqrt{M}} \left[ \cos \omega^T x_i, \sin \omega^T x_i, ..., \cos \omega_m^T x_i, \sin \omega_m^T x_i \right]^T \quad (19)
\]

For simplicity, we set \( \Sigma_p = I \) (unit matrix). The probability density function of the squared exponential covariance function can be obtained by inverse Fourier transform

\[
p^{ARD}(\omega) = \frac{1}{k_{ARD}(0)} \int_{R^d} e^{-i\omega^T \tau} k_{ARD}(\tau) d\tau = \sqrt{2\pi L} \exp \left( -\frac{1}{2} \omega^T L \omega \right) \quad (20)
\]

The spectral point \( \omega_m \) can be obtained by sampling from the above equation. One of the main goals of sparse approximations is to reduce the computational burden while retaining as much predictive accuracy as possible. Sampling from the spectral density constitutes a way of building a sparse approximation.

The computational cost of training the SSGPR algorithm is \( O(nM^2) \). In prediction, the cost of the predicted mean is \( O(M) \) and the cost of the predicted variance is \( O(M^2) \). \( n \) denotes the number of prediction points and \( M \) denotes the number of spectral points. These computational costs are the same as those of most recently proposed sparse GP approximations.

5. Experiment

To validate the performance of SSGPR, we performed robot inverse dynamics learning using SSGPR.
on two datasets and compared it with two regression methods, v-SVR and LGP [13]. The large robot dynamics datasets used for our experiments are the real sarcos robot dataset and the simulated Sarcos dataset. In the sarcos dataset, the robot performs various rhythmic motion patterns with the goal of predicting the torque of each individual joint, and is a regression prediction dataset generated from a bionic robot with 7 degrees of freedom containing a total of 7 moments with 13922 training points and 5569 test points, while the SL_sarcos dataset has 14094 training points and 5520 test points. Each data task of these two datasets consists of a 21-dimensional data vector representing 7 joint position information, 7 joint velocity information and 7 joint acceleration information, and 7 targets, the torque of each joint of the Sarcos robot arm.

We can use the gradient descent method to find the minimal value of the negative log marginal likelihood function of SSGPR to optimize the training of hyperparameters, and the negative log marginal likelihood function of SSGPR is

\[ -\ln p(y|X, \theta) = \frac{1}{2\sigma_n^2} (y^T y - y^T \Phi A^{-1} \Phi^T y) + \frac{1}{2} \ln \det A - \frac{D}{2} \ln \sigma_n^2 + \frac{m}{2} \ln 2\pi \sigma_n^2 \]  

\( \theta \) is the variable containing all hyperparameters. Considering the spectral points as parameters rather than constants, the log marginal likelihood function can also be used for spectral points optimization. In this experiment, we set the number of iterations of the gradient descent method as 100 and the number of adopted spectral points \( M \) as 400. The prediction accuracy evaluation criterion used in the experiment is the normalized mean square error (NMSE). The NMSE is calculated as follows

\[ \text{NMSE} = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \]  

where \( \hat{y} \) is the predicted output of the test set and \( \bar{y} \) is the mean value of the actual output sample.

Figure 1 shows the prediction accuracy NMSE of SSGPR, v-SVR and LGP on the sarcos dataset and SL_sarcos dataset respectively for each joint to build the inverse dynamics model. It can be seen that SSGPR has higher prediction accuracy prediction compared to both v-SVR and LGP regression approaches.

6. Conclusion

This paper presents the use of sparse spectral Gaussian process regression for the learning of robot inverse dynamics, and the feasibility of this method is verified. SSGPR reduces the time complexity by replacing the full spectrum with a set of discrete spectral points to approximate the full stationary GP. Experimental results show that SSGPR has higher prediction accuracy in estimating inverse dynamics models compared with regression methods such as v-SVR and LGP. In the future, we can combine the
proposed method with online learning so that the inverse dynamics model can be updated online to correct for changes due to robot wear or different loads.

References

[1] Uicker, J. J. (1965) On the dynamic analysis of spatial linkages using 4 x 4 matrices. Ph.D. dissertation, Northwestern University.

[2] Li, C. J. (1988) A New Method of Dynamics for Robot Manipulators. In: IEEE Transactions on Systems, Man, and Cybernetics, 18: 105-114.

[3] Orin, D. E., McGhee, R. B. M., Vukobratovic, and Hartoch, G. (1979) Kinematic and kinetic analysis of open-chain linkages utilizing Newton-Euler methods. Math. Biosc., 43: 107-130.

[4] John Hollerbach, M. (1980) A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity. IEEE Transactions on Systems, Man, and Cybernetics, 10: 730-736.

[5] Liu, N. et al. (2019) Modeling and Simulation of Robot Inverse Dynamics Using LSTM-Based Deep Learning Algorithm for Smart Cities and Factories. IEEE Access, 7: 173989-173998.

[6] Hanafusa, M., Ishikawa, J. (2020) A Practical Method for Acquiring Inverse Dynamics Model and its Application to Mechanical Impedance Control of Human-Cooperative Robot. In: 2020 IEEE/SICE International Symposium on System Integration (SII), pp. 579-584.

[7] Zhou, S., Schoellig, A. P. (2019) Active Training Trajectory Generation for Inverse Dynamics Model Learning with Deep Neural Networks. In: 2019 IEEE 58th Conference on Decision and Control (CDC), pp. 1784-1790.

[8] Steffan, L., Rishad, I., Mojtaba, A. (2021) A numeric derivation for fast regressive modeling of manipulator dynamics. In: Mechanism and Machine Theory, pp. 156.

[9] Rain, T., Dovgal, V. M., Soe. Y. N. (2019) Efficient Method for Inverse Dynamics of Robot Manipulators by using Adaptive-network-based Fuzzy Inference System. In: 2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), pp. 1-6.

[10] Rasmussen, C. E., and Williams, C. K. (2006) Gaussian Processes for Machine Learning. Massachusetts Institute of Technology: MIT-Press.

[11] Lee, J., Oh, S. (2019) Data-Based Design of Inverse Dynamics Using Gaussian Process. In: 2019 IEEE International Conference on Mechatronics (ICM), pp. 449-454.

[12] Vijayakumar, S., Schaal, S. (2000) Locally weighted projection regression: an O(n) algorithm for incremental real time learning in high dimensional spaces. In: The 17th international conference on machine learning(ICML), pp. 288-293.

[13] Nguyen-Tuong, D., Peters, J. (2008) Local Gaussian process regression for real-time model-based robot control. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 380-385.

[14] Lázaro-Gredilla, M., Quiñonero-Candela, J., Rasmussen, C. E., Figueiras-Vidal, A. R. (2010) Sparse spectrum Gaussian process regression. Journal of Machine Learning Research, 11: 1865–1881.

[15] Rahimi, A., Recht, B. (2008) Random features for large-scale kernel machines. In Advances in neural information processing systems, 20 : 1177–1184.