Smart Identification for Serial Robots: Programmable Modeling, Software Development, and Engineering Application

Shiheng Xu, Dingxu Guo, Xiaoxu Zhang, *Member, IEEE*, Shu Zhang, and Jian Xu

***Abstract*—Given the significant role of model-based control in industrial robots, it is urgent to provide an efficient tool for accurate model parameter identification. Although extensive issues, such as excitation trajectory design, joint friction representation, and identification algorithm construction, have been figured out, two challenging problems still hinder their popularization in engineering applications. First, conventional modeling approaches are unprogrammable to different robot configurations, making the modeling process frustrating even for professional users. Second, the model loading and identification process are not visualized, making the identification algorithm operation not straightforward. Focusing on these two problems, we proposed a recursive modeling procedure and encapsulated it into a newly designed identification software. Compared to the existing approaches, the greatest strength of this work is that the developed software only requires the minimum input, i.e., DH table and measured motion-torque data, to finish the parameter identification. An experiment with a self-built 3-link manipulator was conducted to demonstrate the software's convenient operation. In addition, self and cross-validation results verified that the software promises high identification accuracy, indicating a bright prospect for its application in industrial robots.**

***Index Terms*—**Recursive modeling, parameter identification, serial robots, sliding window filter, linear regression.

I. INTRODUCTION

FDURoID is tailored for manipulator dynamic parameter identification. Its functionalities encompass reading experimental data, robot modeling using MDH parameters, and conducting dynamic parameter identification. This software is a comprehensive desktop application with graphics user interface. It facilitates a clear, step-by-step process for users to complete the identification of manipulator dynamic parameters and visually displays the identification results.

Built using Python 3.9[1] and Qt 6.4[2], this user-friendly desktop application boasts a modern interface crafted with Qt Quick technology, ensuring cross-platform compatibility.

II. SOFTWARE MANUAL

A. Installation and runtime

Utilizing cx-Freeze[3] to encapsulate Python scripts into standalone executables, this software offers effortless installation and runtime. Users are spared from configuring runtime environments or installing dependent packages. To begin using this software, download the zip package from the website and extract it locally. Locate and click on 'FDURoID.exe' within the extracted folder to launch the software.

B. Operating instructions for the dynamic identification

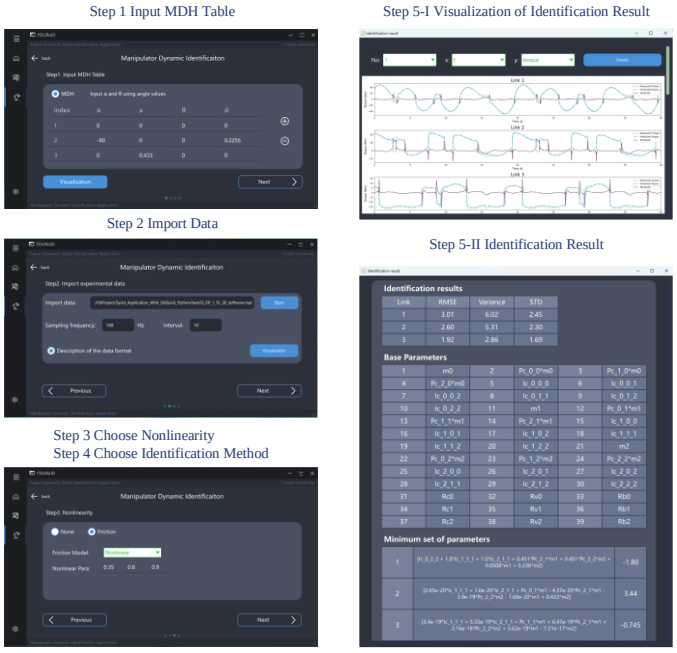

Fig. 1. Procedure for the identification of manipulator dynamic parameters in FDURoID software.

This work was supported by the National Natural Science Foundation of China (Grant No. 12372065, 12372022, and 11932015). *(Corresponding author: Xiaoxu Zhang).*

Shiheng Xu and Jian Xu are with the Academy for Engineering and Technology, Fudan University, Shanghai 200433, China (e-mail: 21210860090@m.fudan.edu.cn (Shiheng Xu); jian_xu@fudan.edu.cn (Jian Xu)).

Xiaoxu Zhang is with Academy for Engineering & Technology, MOE Frontiers Center for Brain Science, Fudan University, Shanghai 200433, China, and Yiwu Research Institute, Fudan University, Yiwu 322000, China. (Email: zhangxiaoxu@fudan.edu.cn).

Dingxu Guo and Shu Zhang are with School of Aerospace Engineering and Applied Mechanics, Tongji University, Shanghai 200092, China. (Email: guodingxu@tongji.edu.cn (Dingxu Guo); zhangshu@tongji.edu.cn (Shu Zhang)).

This article has supplementary downloadable material available at xxx, provided by the authors.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org

Begin by accessing the main interface of the software. Click on the function tab located on the left side. Select 'Manipulator' to enter the interface dedicated to the manipulator-related functions. From there, click the 'Identification' button in the central area of the software to access the operational interface for parameter identification. The steps are shown in Fig. 1.

The first step is to input the MDH (Modified Denavit-Hartenberg) parameter table[4] of the robot arm and use the "+" or "-" buttons located on the right side of the interface to add or remove a row from the table. The MDH parameters consist of rotation α, arm length a, joint angle θ, and offset d. Both the rotation α and joint angle θ should be entered as angular values. For a detailed description of the MDH parameters, please see the reference[4]. Once the input is finalized, clicking the 'Visualization' button in the lower left corner allows observation of the robot's position, ensuring the correctness of the entered MDH parameter table.

The second step involves importing experimental data. It's recommended that experimental data be filtered beforehand to minimize noise interference and enhance the accuracy of the identification. Initially, the software supports the import of the '.mat' file. These files encompass the angle signal, angular velocity signal, angular acceleration signal, and torque signal corresponding to each axis of the manipulator. Specifically, in the case of an n-DOF manipulator, the data columns are structured as follows: columns 1 to n represent joint angle signals, columns n+1 to 2*n represent joint angular velocity signals, columns 2*n+1 to 3*n represent joint angular acceleration signals, and columns 3*n+1 to 4*n represent joint torque signals. Upon preparing the experimental data, open it within the software by clicking the 'Open' button. This action prompts a local file browser window where you can select the experimental data file and open it. Alternatively, simply drag and drop the file into the 'Import data' area to access and open the file. Once the file is successfully opened, the file path will be displayed in the input field. Furthermore, input the sampling frequency of the experimental data in Hz. To mitigate potential memory shortage issues during calculations, an 'Interval' input is included. This specifies the number of points between which the experimental data will be read. It's recommended to do data truncation or resampling during the preprocessing of experimental data to address this issue. Once the experiment file is opened correctly and the relevant information is input, you can select "Visualization" to display the experimental data visually. This allows you to verify the accuracy of the read data. Clicking the "Details" button enables options to save the image or perform additional operations.

The third step is to select the friction model. Use the radio button to select whether to consider joint friction. "None" signifies the exclusion of joint friction, while "Friction" indicates consideration of joint friction. Upon choosing "Friction," access the ComboBox below to select the joint friction model from the options listed in Table 1. When "Coulomb-viscosity" is chosen, no further input is necessary. However, opting for "Nonlinear" requires inputting the nonlinear parameters. Arrange the input columns from left to right, correlating with the nonlinear parameters of each joint of the manipulator from bottom to top. Typically, the value for nonlinear parameters ranges between 0.3-0.9 [5], [6], adjustable based on the specific circumstances. The default value for each joint is 0.5. At present, the software exclusively supports the identification of joint friction. Additional types of joint nonlinearities and joint friction models will be included in forthcoming software updates.

TABLE I

FRICTION MODEL IN SOFTWARE

Name	Friction Model
Coulomb-viscosity	$\tau_f = F_v\dot{q} + F_c \operatorname{sign}(\dot{q})$
Nonlinear	$\tau_f = \left(F_c + F_v\lvert\dot{q}\rvert^{\alpha}\right)\operatorname{sign}(\dot{q}) + B$

The fourth step is to select the identification method. The software operates on the principle of least squares to identify the parameters. Supported identification methods include ordinary least squares (OLS) and weighted least squares (WLS), which can be chosen using the Radio Button. If opting for the weighted least squares method, the weighting function can be selected from the ComboBox options displayed in Table 2. Here, r represents the residual difference between the reconstructed torque and the measured torque derived from the ordinary least squares method's outcome.

TABLE II

WEIGHT FUNCTION IN WLS

Weight Function	Equation
'andrews'	$w = \left(\lvert r\rvert < \pi\right) * \frac{\sin(r)}{r}$
'bisquare'	$w = \left(\lvert r\rvert < 1\right) * \left(1 - r^2\right)^2$
'cauchy'	$w = \frac{1}{1 + r^2}$
'fair'	$w = \frac{1}{1 + \lvert r\rvert}$
'huber'	$w = \frac{1}{\max\left(1,\ \lvert r\rvert\right)}$
'logistic'	$w = \frac{\tanh(r)}{r}$
'talwar'	$w = 1 * \left(\lvert r\rvert < 1\right)$
'welsch'	$w = e^{-r^2}$

After following the aforementioned steps, clicking the 'Start' button initiates an automatic process encompassing modeling, identification, and self-verification. Then the results are visually presented. The page displays reconstructed torque contrasted with measured torque, represented for each link via

line charts, accompanied by residual. Additionally, an analysis of the error showcases root mean square error(RMSE), standard deviation(STD), and variance between the reconstructed and measured torque for each link. Subsequently, the display includes the basic parameter set, encompassing parameters such as mass, center of mass position, moment of inertia, and friction parameters, thereby demonstrating the number of parameters within the dynamic model. Finally, the minimum parameter set of the manipulator, comprising the final identified parameters represented in symbolic form, is presented.

With these steps completed, the dynamic parameter identification process for the manipulator concludes.

REFERENCES

[1] *Python 3.9.7*. (2021). Python Software Foundation. Accessed: December 27, 2023. [Online]. Available: https://www.python.org/downloads/release/python-397

[2] *Qt 6.4.0*. (2022). The Qt Company. December 27, 2023. [Online]. Available: https://wiki.qt.io/Qt_6.4_Release

[3] *Cx_Freeze*. (2023). Marcelo Duarte. Accessed: December 27, 2023. [Online]. Available: https://github.com/marcelotduarte/cx_Freeze

[4] J. J. Craig, *Introduction to robotics : mechanics and control*. Upper Saddle River, Nj: Pearson, 2018.

[5] Y. Han, J. Wu, C. Liu, and Z. Xiong, "An Iterative Approach for Accurate Dynamic Model Identification of Industrial Robots," *IEEE Trans. Robot.*, vol. 36, no. 5, pp. 1577–1594, Oct. 2020, doi: https://doi.org/10.1109/tro.2020.2990368.

[6] T. Xu, J. Fan, Q. Fang, Y. Zhu, and J. Zhao, "Robot dynamic calibration on current level: modeling, identification and applications," *Nonlinear Dynamics*, vol. 109, no. 4, pp. 2595–2613, Jun. 2022, doi: https://doi.org/10.1007/s11071-022-07579-0.