A Case Study of Gravity Compensation for da Vinci Robotic Manipulator: A Practical Perspective

K. LU[1], H. B. Lin[1], C. W. Vincent Hui[1], and K. W. Samuel Au[1]

I. INTRODUCTION

An improved method for gravity compensation of da Vinci Robot manipulator is proposed in this paper. The existing method provided by [1] is a single-step least square estimation (LSE) approach. This method can operate precisely around the center of the workspace, however, for certain configurations, the predicted errors can increase significantly and the estimation can sometimes fail to converge to desired positions. We believe these configuration dependent performance are mainly due to the following two reasons: (1) the use of a Simple Manipulator Model (SMM) without considering the parallelogram mechanism; (2) uneven mass distributions across the manipulator. To improve the performance, we propose to use an improved Manipulator Model with Parallelogram Mechanism (MMPM) for the parameter estimation. During the estimation, an iterative least square estimation (LSE) method is presented to improve the parameter estimation for the distal joints/links. Followed the initial parameter estimation, we also perform additional calibration to fine tune the output torques to compensate for model uncertainties in the physical hardware. We hope this paper can serve as a case study on how to apply gravity compensation effectively.

II. METHODOLOGY

The dynamics of a serial manipulator is governed by the following equation:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau, \tag{1}$$

where $M(q)$ is the inertia matrix, $C(q,\dot{q})$ is the Centrifugal and Coriolis force, and the $G(q)$ is the gravity force. The derivation of the dynamic parameter identification can either be through the Newton-Euler and the Lagrangian approaches[2]. Due to the linear property of the equations with respect to dynamic parameters, we can rearrange the dynamics into linear equations as follows:

$$\boldsymbol{\tau} = Y(q,\dot{q},\ddot{q})\beta, \tag{2}$$

where β and Y are the dynamic parameter vector and regressor matrix, respectively. The Least Square Estimation (LSE) is applied traditionally to estimate the robot dynamic

K. W. Samuel. Au is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China (corresponding author, samuelau@cuhk.edu.hk)

K. LU, H. B. Lin, C. W. Vincent Hui are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China.

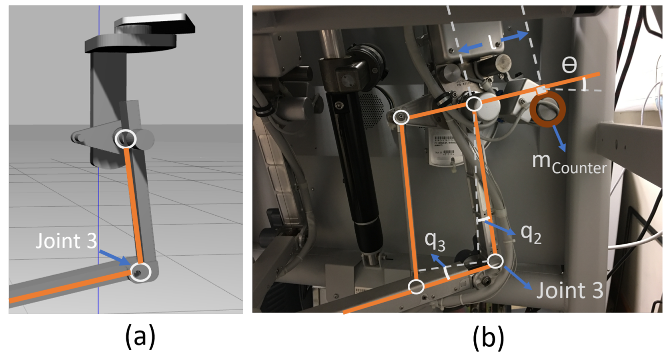

Fig. 1: (a) Simple Manipulator Model (SMM) without considering the parallelogram mechanism. (b) Improved Manipulator Model with Parallelogram Mechanism (MMPM)

parameter by doing the left pseudo inverse of the regressor matrix[3]:

$$\beta = (Y^T Y)^{-1} Y^T \hat{\tau} \tag{3}$$

III. RESULT

A. Model Comparison: SMM vs MMPM

Fig. 1a shows the model used in [1] , where the MTM was modeled as a 7-DoF serial-link robot without the consideration of the counter mass and parallelogram mechanism, making it a 10-parameter estimation problem ($\beta \in \mathbb{R}^{10\times 1}$). Our proposed model includes the effect of the parallelogram mechanism as well as the counter mass (Fig. 1b). Using [4], the effect of the counter mass is modeled as follows:

$$\hat{\tau_3} = \tau_3 - m_{counter} gl * cos(q_2 + q_3), \tag{4}$$

where $m_{counter}$ is the counter mass, and τ_3, $\hat{\tau_3}$ are the actual and measured torques for joint 3, respectively. This model introduces one extra parameter, making it an 11-parameter estimation problem ($\beta \in \mathbb{R}^{11\times 1}$). As can be seen in Fig. 2, the MMPM has much less predicted error in Joint 3, as compared to SMM.

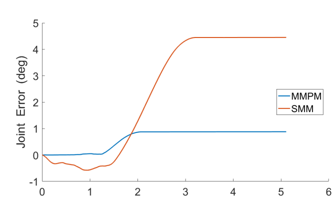

Fig. 2: Comparison of the Joint 3 drift errors obtained through SMM and MMPM

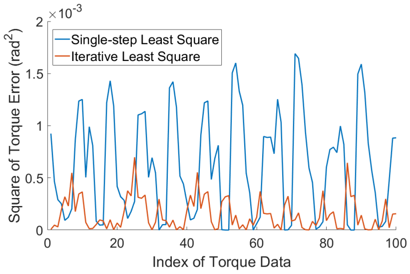

Fig. 3: Comparison of predicted joint torque error (joint 5): Single-step vs Iterative Least Square Methods

B. Least Square Estimation: Single-step vs Iterative Approaches

Looking into the regressor matrix Y for the MTM, we find that it forms an upper triangular matrix from the second row to the sixth row as follows:

$$\begin{array}{cl} \left[\begin{matrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ y_{21} & \dots & y_{2n-4} & y_{2n-3} & y_{2n-2} & y_{2n-1} & y_{2n} \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & y_{d-1n-3} & y_{d-1n-2} & y_{d-1n-1} & y_{d-1n} \\ 0 & \dots & 0 & 0 & 0 & y_{dn-1} & y_{dn} \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{matrix}\right] & \begin{matrix} \\ \leftarrow \text{Last} \\ \\ \leftarrow \text{Next} \\ \leftarrow \text{Start here} \\ \\ \end{matrix} \end{array}$$

Taking advantages of this property, we can first estimate the parameters for the most distal link (6-th row of the regressor matrix) and then using the parameter estimation results from 6-th row to find the parameters for the second most distal link (5-th row). We will iterate this process until we finish the estimation for the first link in the MTM. As this iterative method allows us to address the parameter estimation for each link/joint sequentially, it can improve the convergence and accuracy of the parameter estimation for the distal links/joints with relatively low inertiae and masses. As shown in Fig. 3, the iterative method can achieve much lower estimated torque error as compared to the "Single-step" approach.

C. Additional Manual Calibration

Due to the measurement noise and other disturbances, additional calibration is required to verify and re-fine the estimation of the gravity force. The calibration process is performed sequentially starting from the most distal link. We apply the identified dynamic parameters to the selected joint while disabling the output torques for others. If the predicted torques fail to compensate the gravity (the joint may drift a bit), corresponding dynamic parameters need to be re-calibrated through real data. Later, the newly obtained parameters for the most distal joint will be applied to re-fine the torque estimation problem for the second most distal joint. We will iterate this process until we finish the calibration for each single joint in the manipulator.

$$\hat{\beta} = \beta \pm \Delta \tag{5}$$

IV. Discussions

Fig. 4 illustrates the MTM and frame definitions, while Fig. 5 shows the overall performance of the gravity compensation on the hardware. We measure the compensation

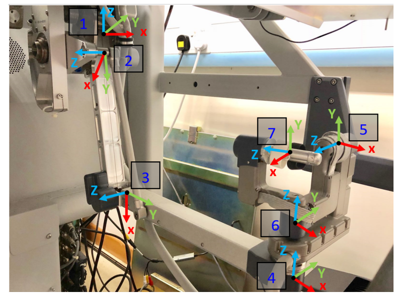

Fig. 4: MTM and frame definition

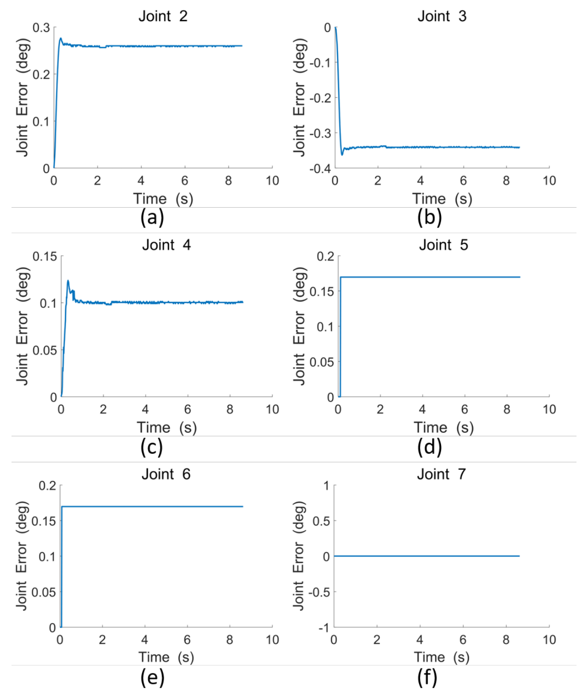

Fig. 5: Joint Drift

based on the drift for each joint of the manipulator after the estimated gravity force is applied. As can be seen, on average, the joint drift for each joint is less than 0.5 degrees. Although some improvement has shown on the torque prediction of distal joints, it can still be sensitive to friction and elastic force in some extreme joint configuration such as the power cable.

References

[1] Alex Rutfield, Amaid Zia, Brandon Boos, Junius Santoso, and Nuttaworn Sujumnong. Gravity compensation control of da vinci robot. `https://github.com/WPI-AIM/dvrk_gravity_comp`, 2017.

[2] B Siciliano, L Sciavicco, L Villani, and G Oriolo. Robotics–modelling, planning and control. advanced textbooks in control and signal processing series, 2009.

[3] GA Fontanelli, F Ficuciello, L Villani, and B Siciliano. Modelling and identification of the da vinci research kit robotic arms. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 1464–1469. IEEE, 2017.

[4] Anton Deguet. Gravity compensation. `https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/Control-Gravity-Compensation`, 2017.