A Dynamic Model Identification Package for the da Vinci Research Kit

Yan Wang, Radian Gondokaryono, Adnan Munawar, and Gregory S Fischer

Abstract—The da Vinci Research Kit (dVRK) is a teleoperated surgical robotic system. For dynamic simulations and model-based control, the dynamic model of the dVRK with standard dynamic parameters is required. We developed a dynamic model identification package for the dVRK, capable of modeling the parallelograms, springs, counterweight, and tendon couplings, which are inherent to the dVRK. A convex optimization-based method is used to identify the standard dynamic parameters of the dVRK subject to physically feasible constraints. The relative errors between the predicted and measured motor torque are calculated on independent test trajectories, which are less than 16.3% and 18.9% for the first three joints and 34.0% and 26.5% for all joints for the master tool manipulator (MTM) and patient side manipulator (PSM), respectively. We open source the identification software package. Although this software package is originally developed for the dVRK, it is easy to apply it on other robots with similar characteristics to the dVRK through simple configuration.

Index Terms—da Vinci Research Kit, dynamic model identification, surgical robot, tendon-driven robot.

I. INTRODUCTION

The da Vinci Research Kit (dVRK) is an open-source surgical robotic system whose mechanical components are obtained from the first generation of the da Vinci Surgical Robot®[1]. It has made the research on surgical robotics more accessible. To date, researchers from over 30 institutes around the world are using the physical dVRK [2], and some other researchers are using dVRK simulations [1].

Simulations and model-based control require the dynamic model of the dVRK. Fontanelli et al. [3] has obtained the dynamic model of the dVRK using dynamic model identification techniques. However, the dynamic parameters obtained in [3] are base parameters (also called lumped parameters) [4], a minimum set of dynamic parameters that can sufficiently describe the dynamic model of a robot. Although base parameters are adequate to represent the dynamics of a robot in dynamic equations, standard parameters are required for the efficient recursive Newton-Euler-based dynamic algorithms. Towards this end, several dynamics libraries utilize standard parameters, such as Rigid Body Dynamics Library (RBDL) [5] and Kinematics and Dynamics Library (KDL) [6].

The dynamic parameters vary between different robots of the same make and model due to manufacturing and assembly variances. Furthermore, the assembly components of the robots are subject to deformation and wear & tear along their life cycle which can potentially alter the dynamic model. As such, the dynamic model identification is required before implementation of any robust model-based control algorithm. This requirement drives the need for a robust open-source dynamic model identification package. There are existing software packages for the dynamic model identification of generic open-chain manipulators such as SymPybotsics [7] and FloBaRoID [8]. However, these packages lack the capability of modeling closed-loop kinematic chains, springs, counterweights, and tendon couplings, which are inherent to dVRK’s mechanical design. To address this, we utilize and extend the convex optimization-based method proposed by Sousa and Cortesão [9] to obtain the physically-feasible standard dynamic parameters of the dVRK arms.

This paper is structured into seven sections as the workflow of dynamic model identification in Fig. 1. Sections II and III explain the mathematical formulation of the kinematic and dynamic modeling of the Master Tool Manipulator (MTM) and Patient Side Manipulator (PSM). Section IV describes the trajectory optimization method to improve parameter identification quality. Section V presents the identification method to obtain the standard dynamic parameters with physical feasibility considered. The experimental results which validate the proposed approaches are presented in section V. Finally, the concluding arguments are entailed in section VII.

II. KINEMATIC MODELING OF THE dVRK

The dVRK-ROS package [1] employs the (Denavit-Hartenberg) DH convention based on kinematic frames located on the joint axes, whereas almost all the joints are actuated off axes using a combination of cams, links, or cables and pulleys. Consequently, we need to define additional coordinate axes to relate the joint motions - as defined in the dVRK-ROS package - with the motor torques.
TABLE I  
MODELING DESCRIPTION OF THE MTM. LINKS 1 TO 7 CORRESPOND TO THE LINKS DESCRIBED IN FIG. 2. $M_4$ CORRESPONDS TO THE FRICTION AND INERTIA MODELING OF MOTOR 4. THE DIMENSIONS ARE SHOWN IN FIG. 2.

| link | joint type | prev | succ | $\alpha_{k-1}$ | $\alpha_{k-1}$ | $d_k$ | $\theta_k$ | link inertia | motor inertia | friction | spring |
|------|------------|------|------|---------------|---------------|------|-----------|--------------|--------------|----------|--------|
| 1    | R          | 0    | 2, 3' | 0            | 0             | $-l_{base2pitch}$ | $q_1$ | ✓          | ✓           | ✓          | ✓       | ✓     |
| 2    | R          | 1    | 3    | 0            | $-\pi/2$      | 0    | $q_2 + \pi/2$ | ✓           | ✓          | ✓        | ✓      |
| 3    | R          | 2    | 4    | $l_{arm}$    | 0             | 0    | $q_3 + \pi/2$ | ✓           | ✓          | ✓        | ✓      |
| 3'   | R          | 3''  | 3''  | 0            | $-\pi/2$      | 0    | $q_4 + q_5 + \pi$ | ✓           | ✓          | ✓        | ✓      |
| 3''  | R          | 3    | 3''  | $l_{back2front}$ | 0            | 0    | $-q_4 - q_5 + \pi/2$ | ✓           | ✓          | ✓        | ✓      |
| 4    | R          | 3    | 5    | $-\pi/2$    | 5             | 0    | $q_6 + \pi/2$ | ✓           | ✓          | ✓        | ✓      |
| 5    | R          | 4    | 6    | 0            | $-\pi/2$      | 5    | $q_7 + \pi$   | ✓           | ✓          | ✓        | ✓      |
| 6    | R          | 5    | 6    | 0            | $-\pi/2$      | 0    | $q_6 + \pi/2$ | ✓           | ✓          | ✓        | ✓      |
| 7    | R          | 6    | -    | 0            | 0             | 0    | $q_4^m$      | ✓           | ✓          | ✓        | ✓      |
| $M_4$| R          | -    | -    | 0            | 0             | 0    |            | ✓           | ✓          | ✓        | ✓      |

To build the relationship between the robot joint motion in the dVRK-ROS package [1] and the torque of each motor, several types of joint coordinates are defined. $q^d$ is the joint coordinate used in the dVRK-ROS package. $q = [(q^b)^\top (q^a)^\top]^\top$ is the joint coordinate used in the kinematic modeling in this work, where $q^b$ is the basis joint coordinate which can adequately represent the kinematics of the robot, and $q^a$ is the additional joint coordinate which represents the joint coordinate of the passive joints in the parallel mechanism and can be represented by the linear combination of $q^b$. Since both the MTM and PSM have seven actuated degrees of freedom (DOF), the basis joint coordinate can be represented by $q^b = [q_1 \ q_2 \ \ldots \ q_7]^\top$. $q^m$ is the equivalent motor coordinate which is considered to happen at joints, with the reduction ratio caused by gearboxes and tendons included for most motors unless explicitly specified. Finally, $q^c = [(q^b)^\top (q^m)^\top]^\top$ defines the complete joint coordinate. The relation between these joint coordinates is illustrated for both the MTM and PSM in this section.

A. Kinematic Modeling of the MTM

The left and right MTMs are identical to each other, except the last four joints being mirrored to each other. Consequently, the two MTMs can be modeled in a similar fashion. The frame definition is shown in Fig. 2, and the kinematic parameters of the MTM are described in Table I. The kinematics of the MTM can be described as

- Joint 1 rotates around the Z-axis of the base frame, z₀.
- Joints 2, 3, 3’, 3”, and 3”’ construct a parallelogram, which is actuated by joints 2 and 3’.
- Joints 4, 5, 6, and 7 form a 4-axis non-locking gimbal.

The kinematics of the MTM is fully described by the basis joint coordinates $q^b$, which are equal to the dVRK joint coordinate $q^d$. $q^b = q^d$. The additional joints $q^a$ can be described as the linear combination of $q^b$ by

$$q^a = [q_{1b} \ q_{2b} \ q_{3b'} \ q_{3''b}]^\top = [q_2 + q_3 \ -q_3 \ q_3]^\top \quad (1)$$

Joints 1, 5, 6, and 7 are independently driven, and thus the motion of these joints is equivalent to their corresponding driven motors, $q_{3,5,6,7}^a = q_{3,5,6,7}^m$. The motion of $q_{4}^d$ depends on both $q_{3,m}$ and $q_{4}^d$ and can be described by

$$q_{4}^d = q_{4}^m - r_3/r_4 \cdot q_{4}^d \quad (2)$$

where $r_3$ and $r_4$ are the radii of the pulleys shown in Fig. 3. Based on the user guide of the dVRK, $r_3 \approx 14.01$ mm and $r_4 \approx 20.92$ mm, and thus $r_3/r_4 \approx 0.6697$.

The coupling between the dVRK-ROS joint coordinates $q_{2-4}^d$ and motor joint coordinates $q_{2-4}^m$ due to the parallelogram and tendon is resolved by the coupling matrix $A_m^{d}$ as

$$q_{2-4}^d = A_m^{d} q_{2-4}^m \quad (3)$$

where $A_m^{d} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0.6697 & -0.6697 & 1 \end{bmatrix}$, based on the user guide.
of the dVRK.

B. Kinematic Modeling of the PSM

The frame definition and kinematic dimensions of the PSM are shown in Fig. 4, and the corresponding parameters are shown in Table II. The kinematics of the PSM can be concluded as

- The first two revolute joints form a remote-center-of-motion (RCM) point which remains fixed in Cartesian space. This RCM is achieved via a double four-bar linkage with six links actuated by a single motor.
- The third joint is prismatic and provides insertion of the instrument through the RCM. The first three joints allow the 3-DOF Cartesian space motion.
- Revolute joints 4 and 5 construct the roll and pitch motion of the wrist to reorient the end-effector.
- The last two joints construct the yaw motion of the end-effector, as well as the opening and closing of the gripper.

We model the first five joints of the PSM identical to the dVRK-ROS package, that is \( q_{1-5} = q_{d1-5} \). The dVRK-ROS package models the last two joints as \( q_{6}^{d} \), the angle from the insertion direction to the bisector of the two jaw tips, and \( q_{7}^{d} \), the angle between the two jaw tips. However, the gripper jaws are designed and actuated as two separate links which we consider in our model. As shown in Fig. 5a, the relation between the dVRK-ROS joint coordinates \( q_{6-7}^{d} \), and the joint coordinates \( q_{6-7} \) in our model is described by

\[
q_{6-7} = \begin{bmatrix} q_{6}^{d} \\ q_{7}^{d} \end{bmatrix}^T = \begin{bmatrix} 0.5q_{6} + 0.5q_{7} \\ -q_{6} + q_{7} \end{bmatrix}^T \quad (4)
\]

Since the first four joints are independently driven, the equivalent motor motion is considered to occur at joints and thus is the same as joint motion, \( q_{1-4}^{d} = q_{1-4}^{m} \). Based on the user guide of the dVRK, the coupling of the wrist joint actuation can be resolved by the coupling matrix \( A_{m}^{d} \), mapping the motor joint coordinates \( q_{5-7}^{m} \) to the dVRK-ROS joint coordinates \( q_{5-7}^{d} \) by

\[
q_{5-7}^{d} = A_{m}^{d}q_{5-7}^{m} \quad (5)
\]
B. Dynamic Model Formulation

The Euler-Lagrange equation for closed-chain robots [12] is used to model the dynamics of the dVTK. The Lagrangian is calculated by the difference of the kinetic energy \( K \) and potential energy \( P \) of the robot, \( L = K - P \). Motor inertias and springs are not included in \( L \) and are modeled separately.

The relation from motor motion \( q^m \) to the torque of each motor \( i \) caused by link inertia is then computed as

\[
\tau_{Li}^m = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i^m} - \frac{\partial L}{\partial q_i^m} \quad \text{(11)}
\]

The friction torques of all the joints \( q^c \) are considered as

\[
\tau^c_i(q^c) = F_v(q^c) + F_c \text{sgn}(\dot{q}^c) + F_o \quad \text{(12)}
\]

where \( F_v \) and \( F_c \) are diagonal matrices encapsulating the viscous and Coulomb friction constants, and \( F_o \) is the vector of the Coulomb friction offset constants corresponding to the joint coordinate \( q^c \).

The torques caused by motor inertia are defined as

\[
\tau^m_{Mi}(q^m) = I_m \dot{q}^m \quad \text{(13)}
\]

For spring \( k \), we only model the stiffness constant \( K_{sk} \) as its parameter, which results into the spring torques

\[
\tau^s_{Ks}(q^c) = K_s \Delta l_s \quad \text{(14)}
\]

where \( K_s \) is a diagonal matrix of the stiffness constants of the springs, and \( \Delta l_s \) is their corresponding equivalent prolongation vector.

The joint torques caused by springs and frictions can be projected onto the motor joints, using the Jacobian matrix of their corresponding joint coordinate with respect to the motor joint angle \( q^m \) [12]. Thus, the motor torques \( \tau^m \) with link inertia, springs, frictions, motor inertia, and motion couplings considered are given by

\[
\tau^m = \tau^m_{Ll} + \tau^m_{Ml}(q^m) + \frac{\partial q^c}{\partial q^m} (\tau^s_{Ks}(q^c) + \tau^c_j(\dot{q}^c)) \quad \text{(15)}
\]

To identify \( \delta \), (15) is rewritten into (16) by the linear parameterization.

\[
\tau^m = H(q^m, \dot{q}^m, \ddot{q}^m) \delta \quad \text{(16)}
\]

QR decomposition with pivoting [4] is used to calculate the base parameters. With this method, we get a permutation matrix \( P_b \in \mathbb{R}^{n \times b} \), where \( n \) is the number of standard dynamic parameters and \( b \) is the number of base parameters. The base parameters \( \delta_b \) and the corresponding regressor \( H_b \) can be calculated by

\[
\delta_b = P_b^\top \delta, \quad H_b = HP_b \quad \text{(17)}
\]

C. Dynamic Modeling of the Master Tool Manipulator (MTM)

The dynamic modeling description for each link of the MTM is shown in Table I. All the nine links are modeled with link inertia. The frictions of all the modeling joints \( q \) are considered, except joint \( 3'' \) since joint \( 3'' \) and joint \( 3'' \) share the same joint coordinate. For an independently driven joint, the friction from the joint and its driven motor is coupled together and thus impossible to distinguish from each other.
is given by
\[ \tau_{s5} = f_s \cdot d_s = K_{s5}(l_s - l_r) \cdot d_s = K_{s5} \Delta l_{s5} \]  
(18)
where \( l_s \) is the length between the two axes connecting the spring, which can be calculated using the law of sines as
\[ l_s = \sqrt{h_s^2 + r_s^2 - 2h_s r_s \cos(\pi + q_o - q_s)} \]  
(19)
and \( l_r \approx 61.3 \) mm by measurement is the value of \( l_s \) when the spring is relaxed. Based on basic trigonometry, the moment arm \( d_s \) can be calculated by
\[ d_s = h_s r_s \sin(\pi + q_o - q_s)/l_s \]  
(20)
where \( h_s, r_s \) and \( q_o \) are constants shown in Fig. 7b. Thus, \( \Delta l_{s5} = (l_s - l_r)d_s \).

D. Dynamic Modeling of the PSM

The dynamic modeling description of the PSM is shown in Table II. Inertia is considered for all the links contributing to the Cartesian motion, including the counterweight, link 3'. The motor inertia of these joints is ignored since it is not significant compared to their link inertia. The inertia of the wrist and gripper links is minimal, and thus infeasible to identify. Therefore, we only model the inertia of motors for the wrist and gripper, corresponding to the motion of \( q^m_{4-7} \).

Since joints 2, 2', 2", 2", and 2'" are all driven by a single motor, their friction can be represented by the friction of one joint for simplicity. Thus, among these joints, only joint 2 is modeled with friction. Similarly, only joint 3 is modeled with friction out of joints 3 and 3'. Because of the contact between links 5 and 6, and between links 5 and 7 as shown in Fig. 5b, the frictions on joints 6 and 7 are modeled, corresponding to the motion of \( q_6 \) and \( q_7 \). Moreover, the friction between link 6 and link 7 due to the contact between the two jaw tips is considered, corresponding to the motion of \( q_7 - q_6 \). Additionally, the frictions on the motor sides of the last four joints are also modeled, corresponding to the motor motion of \( q^m_{4-7} \).

The torsional spring on joint 4 which rotates the joint back to its home position is modeled as
\[ \tau_{s4} = K_{s4}(-q_4) = K_{s4} \Delta l_{s4} \]  
(21)

IV. EXCITATION TRAJECTORY OPTIMIZATION

A periodic excitation trajectory based on Fourier series [13] is used to generate data for dynamic model identification. This trajectory minimizes the condition number of the regression matrix \( W_b \) for the base parameters \( \delta_b \), that decide the dynamic behavior of a robot.

\[ W_b = \begin{bmatrix} H_b(q_1^m, \dot{q}_1^m, \ddot{q}_1^m) \\ H_b(q_2^m, \dot{q}_2^m, \ddot{q}_2^m) \\ \vdots \\ H_b(q_n^m, \dot{q}_n^m, \ddot{q}_n^m) \end{bmatrix} \]  
(22)

where \( q_i^m \) is the motor joint coordinate at \( i^{th} \) sampling points and \( S \) is the sampling point number. The joint position \( q_{ak} \) of joint \( k \) can be calculated by
\[ q_{ak}(t) = q_{ak}^m + \sum_{l=1}^{n_H} a_{lk} \sin(\omega_l t) - b_{lk} \cos(\omega_l t) \]  
(23)
where $\omega_f = 2\pi f_f$ is the angular component of the fundamental frequency $f_f$, $n_H$ is the harmonic number of Fourier series, $a_{l_k}$ and $b_{l_k}$ are the amplitudes of the $l_l$-th order sine and cosine functions of joint $k$, $q_{mk}$ is the position offset of motor joint $k$, and $t$ is the time.

The motor joint velocity $\dot{q}_{mk}^m(t)$ and acceleration $\ddot{q}_{mk}^m(t)$ can be calculated easily by the differentiation of $q_{mk}^m(t)$. And the trajectory must satisfy the following constraints:

- The joint position $q$ is between the lower bound $q_l$ and the upper bound $q_u$, $q_l \leq q \leq q_u$.
- The robot is confined in its work space. The Cartesian position $p_k$ of frame $k$ is within its lower bound $p_l$ and upper bound $p_u$, $p_l \leq p \leq p_u$.

Finally, pyOpt [14] is used to solve this constrained nonlinear optimization problem.

V. Parameter Identification

To identify the dynamic parameters, we use the excitation trajectory described in Section IV to move the robot. Data is collected at each sampling time to obtain the regression matrix $W$ and the dependent variable vector $\omega$:

$$W = \begin{bmatrix} H(q_{m1}^1, q_{m2}^2, \ldots, q_{mS}^S) \\ H(q_{r1}^1, q_{r2}^2, \ldots, q_{rS}^S) \end{bmatrix}, \quad \omega = \begin{bmatrix} \tau_{m1}^m \\ \tau_{m2}^m \\ \vdots \\ \tau_{mS}^m \end{bmatrix}$$

(24)

where $S$ is the sampling point number, and $q_{mk}$ and $\tau_{mk}$ are the motor joint position and torque at the $i$th sampling point.

The identification problem can then be formulated into an optimization problem which minimizes the squared residual error $||\epsilon||^2$ with respect to the decision vector $\delta$.

$$||\epsilon||^2 = ||W\delta - \omega||^2.$$  

(25)

To get more realistic dynamic parameters and avoid over-fitting to the identification data, we utilized the physical feasibility constraints for the dynamic parameters:

- The mass for each link $k$ is positive, $m_k > 0$.
- The inertia matrix of each link $k$ is positive definite, $I_k > 0$ [15], and its eigenvalues $Y_x$, $Y_y$ and $Y_z$ should follow the so-called triangle inequality conditions [16], $Y_x + Y_y > Y_z$, $Y_y + Y_z > Y_x$, and $Y_z + Y_x > Y_y$.
- The COM of link $k$, $r_k$, is inside its convex hull, $m_k r_{ik} - l_k \leq 0$ and $m_k r_{ak} + l_k \leq 0$, where $r_{ik}$ and $r_{ak}$ are the lower and upper bound of $r_k$, respectively [9].
- The viscous and Coulomb friction coefficients for each joint $i$ are positive, $F_{ci} > 0$ and $F_{ei} > 0$.
- The inertia of motor $k$ is positive, $I_{mk} > 0$.
- The stiffness of spring $j$ is positive, $K_j > 0$.

The first two constraints regarding the inertia properties of each link $k$ can be derived into an equivalent [17] as

$$\tilde{D}_k(\delta_{Lk}) = \begin{bmatrix} \frac{1}{2} \text{tr} (L_k) \cdot 1 - L_k \\ l_k \\ m_k \end{bmatrix} > 0$$

(26)

We can also add the lower and upper bounds to $m_k$, $F_{ci}$, $F_{ei}$ and $K_j$ when we have more knowledge about them.

\begin{table}[h]
\centering
\caption{Joint Constraints of the MTM. The Units are ° for $q$ and rad/s for $\dot{q}$}
\begin{tabular}{cccccccc}
\hline
\hline $q_1$ & $q_2$ & $q_3$ & $q_4$ & $q_5$ & $q_6$ & $q_7$ \\
$\text{degree}$ & $\text{degree}$ & $\text{degree}$ & $\text{degree}$ & $\text{degree}$ & $\text{degree}$ & $\text{degree}$ \\
\hline
$q_{\text{min}}$ & -57 & -30 & -9 & -87 & -40 & -460 \\
$q_{\text{max}}$ & 29 & 60 & 39 & 195 & 38 & 450 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Joint Constraints of the PSM. The Units are ° or m for $q$ and rad/s or m/s for $\dot{q}$}
\begin{tabular}{cccccccc}
\hline
\hline $q_1$ & $q_2$ & $q_3$ & $q_4$ & $q_5$ & $q_6$ & $q_7$ \\
$\text{degree}$ & $\text{degree}$ & $\text{degree}$ & $\text{degree}$ & $\text{degree}$ & $\text{degree}$ & $\text{degree}$ \\
\hline
$q_{\text{min}}$ & -85 & -45 & 0.07 & -86 & -86 & -80 \\
$q_{\text{max}}$ & 85 & 46 & 0.235 & 86 & 86 & 80 \\
\hline
\end{tabular}
\end{table}

Finally, we use the CVXPY package [18] with the SCS solver [19] to solve this convex optimization problem. With the identified barycentric parameters, the standard inertia parameters are computed by solving (6) and (7).

VI. Experimental results

This section presents the experimental results of the dynamic model identification conducted on the dVRK arms.

A. Excitation Trajectory Generation and Robot Excitation

Two independent excitation trajectories are generated for each of the MTM and PSM. One is for identification, and the other is for test. The harmonic number $n_H$ is set to 6. The fundamental frequency $f_f$ of the MTM and PSM are 0.1 Hz and 0.18 Hz, respectively. The joint position and velocity are constrained in the optimization, as in Table III and IV. Since links 2$^1$ and 2$^2$ of the PSM have similar motion and are very close to each other, it is hard to get a trajectory with a low condition number of the regression matrix $W_k$ when both links 2$^1$ and 2$^2$ are considered. Links 2$^1$ and 2$^2$ have the similar problem. Therefore, the trajectory optimization of the PSM is based on the model without links 2$^1$ and 2$^2$. The obtained optimal excitation trajectories for identification of the MTM and PSM are shown in Fig. 8, with the condition number of 211 and 362, respectively. The trajectories for test can be found in the open-source package and not shown here.

When the robot moves along these trajectories, the trajectory amplitudes $a$ and $b$ are increased gradually from zero to their nominal values in the first five seconds, which ensures the continuity of velocity and acceleration. The joint position, velocity, and torque are then collected at 200 Hz with the robots running in position control mode.

The joint position and velocity are collected directly, and the joint acceleration is obtained by the second-order numerical differentiation of the velocity. A 6th order low-pass Butterworth filter is used to filter all the data with the cutoff frequencies of 1.8 Hz for the MTM and 5.4 Hz for the PSM.
To achieve zero phase delay, we apply this filter in both the forward and backward directions.

B. Identification

To get uniformly precise identification results for all joints, the residual error $\epsilon_i$ of each motor joint $i$ in (25) is weighted by $w_i = 1/\{\max\{\tau_i^m\} - \min\{\tau_i^m\}\}$. The identified dynamic parameters $\hat{\delta}$ from identification trajectories are used to predict the motor joint torque on test trajectories, $\hat{\omega} = W\hat{\delta}$. The relative root mean squared error is used as the relative prediction error to assess the identification quality, $\epsilon = ||\omega - \hat{\omega}||/||\omega||$.

| TABLE V RELATIVE PREDICTION ERROR ON TEST TRAJECTORIES |
|--------------------------------------------------------|
|               | $\tau_1^m$ | $\tau_2^m$ | $\tau_3^m$ | $\tau_4^m$ | $\tau_5^m$ | $\tau_6^m$ |
| MTM (%)       | 7.3        | 15.1       | 16.2       | 22.3       | 27.0       | 23.3       | 34.0       |
| PSM (%)       | 9.1        | 17.9       | 18.9       | 13.4       | 23.9       | 21.2       | 26.5       |

Fig. 9 and 10 show the comparison of the measured motor torque and predicted motor torque on the test trajectories using the identified dynamic parameters for the MTM and PSM, respectively. The relative prediction error of each motor joint is shown in Table V.

The maximum relative prediction error of the MTM occurs on motor joint 7, which is 34.0%. The relative prediction error of the first three motor joints is less than 16.3%, which correspond to the Cartesian motion and most of the link inertia of the MTM. The large backlash caused by the gearboxes and the small link inertia of the last four joints make it hard to identify their dynamic parameters accurately. Hence, the relative prediction error of the last four motor joints is relatively higher. The overall identification performance of the MTM is better than [3], in which the maximum relative prediction error is 43.5% for all the joints and 39.1% for the first three joints.

The maximum relative prediction error of the PSM occurs on motor joint 7, which is 26.5%. The relative prediction error of the first three motor joints is less than 18.9%, which correspond to the Cartesian motion and most of the link inertia of the arm. The relative prediction error of the last four motor joints is relatively larger since they are only modeled with
motor inertia and frictions and the magnitudes of the joint torques are very small. The overall identification performance of the PSM is also better than [3], in which the largest relative prediction error is 45.3% for all the joints and 31.6% for the first three joints.

VII. CONCLUSION

In this work, an open-source software package for the dynamic model identification of the full standard dynamic parameters of the dVRK is presented (https://github.com/WPI-AIM/dvrk_dynamics_identification). Link inertia, joint friction, springs, tendon couplings, cable force, and closed-chains are incorporated in the modeling. Fourier series-based trajectories are used to excite the dynamics of the dVRK, with the condition number of the regression matrix minimized. A convex optimization-based method is used to get the full standard dynamic parameters subject to physical feasibility constraints, which are more suitable for the fast recursive Newton-Euler computation. The identification performance is improved significantly compared to previous work [3]. The package is written in Python under Jupyter Notebooks with free dependent software modules, which makes it easy to read and replicate. Although this software package is initially developed for the dVRK, it is easy to use it for the dynamic model identification of other robots.

Future work will be devoted to further reducing the error by precisely modeling the cables and gears, and applying the identified dynamic model in simulations and model-based control.

ACKNOWLEDGMENT

This work has been supported by the National Science Initiative NSF grant: IIS-1637759.

REFERENCES

[1] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, “An open-source research kit for the da vinci® surgical system,” in Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014, pp. 6434–6439.
[2] da vinci research kit wiki community. [Online]. Available: http://research.intusurg.com/dvrkwiki/
[3] G. 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. IEEE, 2017, pp. 1464–1469.
[4] M. Gautier, “Numerical calculation of the base inertial parameters of robots,” Journal of Field Robotics, vol. 8, no. 4, pp. 485–506, 1991.
[5] M. L. Felis, “Rbdl: an efficient rigid-body dynamics library using recursive algorithms,” Autonomous Robots, pp. 1–17, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10514-016-9574-0
[6] R. Smits, “KDL: Kinematics and Dynamics Library,” http://www.orocos.org/kdl.
[7] C. D. Sousa, “Sympybotics v1.0,” 2014. [Online]. Available: https://zenodo.org/record/11365
[8] S. Bethge, J. Malzahn, N. Tsagarakis, and D. Caldwell, “Flobaroida software package for the identification of robot dynamics parameters,” in International Conference on Robotics in Alpe-Adria Danube Region. Springer, 2017, pp. 156–165.
[9] C. D. Sousa and R. Cortesão, “Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach,” The International Journal of Robotics Research, vol. 33, no. 6, pp. 931–944, 2014.
[10] P. Maes, J.-C. Samin, and P.-Y. Willems, “Linearity of multibody systems with respect to barycentric parameters: Dynamics and identification models obtained by symbolic generation,” Mechanics Based Design of Structures and Machines, vol. 17, no. 2, pp. 219–237, 1989.
[11] W. Khalil and E. Dombre, Modeling, identification and control of robots. Butterworth-Heinemann, 2004.
[12] Y. Nakamura and M. Ghodoussi, “Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators,” IEEE Transactions on Robotics and Automation, vol. 5, no. 3, pp. 294–302, 1989.
[13] J. Swevers, C. Ganseman, D. Bilgin, J. De Schutter, and H. Van Brussel, “Optimal robot excitation and identification,” IEEE transactions on robotics and automation, vol. 13, no. 5, pp. 730–740, 1997.
[14] R. E. Perez, P. W. Jansen, and J. R. Martins, “pyopt: a python-based object-oriented framework for nonlinear constrained optimization,” Structural and Multidisciplinary Optimization, vol. 45, no. 1, pp. 101–118, 2012.
[15] K. Yoshida and W. Khalil, “Verification of the positive definiteness of the inertial matrix of manipulators using base inertial parameters,” The International Journal of Robotics Research, vol. 19, no. 5, pp. 498–510, 2000.
[16] S. Traversaro, S. Brossette, A. Escande, and F. Nori, “Identification of fully physical consistent inertial parameters using optimization on manifolds,” in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016, pp. 5446–5451.
[17] P. M. Wensing, S. Kim, and J.-J. E. Slotine, “Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution,” IEEE Robotics and Automation Letters, vol. 3, no. 1, pp. 60–67, 2018.
[18] S. Diamond and S. Boyd, “CVXPY: A Python-embedded modeling language for convex optimization,” Journal of Machine Learning Research, vol. 17, no. 83, pp. 1–5, 2016.
[19] B. O’Donoghue, E. Chu, N. Parikh, and S. Boyd, “SCS: Splitting conic solver, version 2.0.2,” https://github.com/cvxgrp/scs, Nov. 2017.