

A Hybrid Model and Learning-Based Force Estimation Framework for Surgical Robots

Hao Yang, Haoying Zhou, Gregory S. Fischer and Jie Ying Wu

Abstract—Haptic feedback to the surgeon during robotic surgery would enable safer and more immersive surgeries but estimating tissue interaction forces at the tips of robotically controlled surgical instruments has proven challenging. Few existing surgical robots can measure interaction forces directly and the additional sensor may limit the life of instruments. We present a hybrid model and learning-based framework for force estimation for the Patient Side Manipulators (PSM) of a da Vinci Research Kit (dVRK). The model-based component identifies the dynamic parameters of the robot and estimates free-space joint torque, while the learning-based component compensates for environmental factors, such as the additional torque caused by trocar interaction between the PSM instrument and the patient’s body wall. We evaluate our method in an abdominal phantom and achieve an error in force estimation of under 10% normalized root-mean-squared error. We show that by using a model-based method to perform dynamics identification, we reduce reliance on the training data covering the entire workspace. Although originally developed for the dVRK, the proposed method is a generalizable framework for other compliant surgical robots. The code is available at https://github.com/vu-maple-lab/dvrk_force_estimation.

I. INTRODUCTION

Robot-assisted minimally invasive surgery systems have revolutionized the laparoscopic surgery procedures. They provide an ergonomic setup for the surgeon while improving patient recovery time [1]. However, the lack of direct haptic feedback on all but a few select instruments is a significant barrier for surgeons in building surgical intuition. To enable haptic feedback on surgical robots more widely, various approaches have been suggested [2]–[4] that include integrating force sensors [5], [6] and sensorless force estimation [7], [8].

Sensor-based methods have the advantage of measuring forces independently of robot dynamics. Chua et al. propose a modular 3 degrees-of-freedom force sensor to measure grip force [5]. Fontanelli et al. propose an optic-based force sensor, which can be integrated with a trocar and sense the interactive force between the instrument shaft and patient body [6]. These sensor-based methods, however, require extra mechatronics, as well as extra manufacturing and sensor fusion efforts.

Another direction for providing haptic feedback is sensorless force estimation through calibrating robot models. Liu

et al. develop a sensorless model-based scheme to identify system dynamics and estimate contact force for an industrial serial robot [9]. However, accurate dynamic models for cable-driven surgical robots are complex as they contain parallelogram mechanisms, counterweight, springs, and cable friction. Additionally, the components of robots are subject to deformation and wear and tear along their life cycle. Moreover, during surgeries, the surgical robots move about the remote center of motion (RCM) [10] through a trocar inserted into the patient’s abdomen to guide the instrument. The instruments are then inserted into the abdominal cavity through the trocar. The interaction force between the patient body wall, the trocar, and the instrument is hard to model, which complicates in vivo force estimation [11]. Therefore, it is essential to develop force estimation methods that can provide an accurate dynamic model that can account for these interactions.

Both sensor-based and model-based approaches face challenges in widespread adoption. Force sensors are expensive and difficult to sterilize while model-based force estimation methods are limited. On the other hand, recent learning-based force estimation methods are highly dependent on training data and generalize poorly [7], [8], [12]. In this work, we propose a hybrid approach to force estimation that combines the generalizability of model-based method with the adaptability of the learning-based method. Specifically, we propose a hybrid model and learning-based joint torque estimation method and the Jacobian to convert joint torques to Cartesian forces. We first perform an optimization-based dynamic parameter identification and joint torque estimation of a dVRK [13]. Then, we add a neural network second step to account for extra joint torques caused by setup-specific trocar interaction that can be trained with minimal data.

II. RELATED WORKS

A. Model-based Torque Estimation

One class of dynamic identification and torque estimation methods is model-based. Sang et al. and Piqué et al. propose explicit dynamic models for the da Vinci Research Kit (dVRK) Patient Side Manipulator (PSM), with the free space dynamic parameters identified through ordinary least square optimization [14], [15]. Souza et. al state that in order to achieve a physically feasible robot, the mass inertia matrix in the inverse dynamic equation needs to be positive semidefinite and the mass of each link should always be positive [16]. Fontanelli et al. use these constraints to form a linear matrix inequality-based method for dynamic parameter and joint torque identification for dVRK PSM [17]. On top

This work is supported in part by an Intuitive Surgical Technology Research Grant and NSF AccelNet award OISE-1927275. Hao Yang and Jie Ying Wu are with the Department of Computer Science, Vanderbilt University, TN 37212, USA. Haoying Zhou and Gregory S. Fischer are with the Department of Robotics Engineering, Worcester Polytechnic Institute, MA 01609, USA. All correspondence should be addressed to Hao Yang hao.yang@vanderbilt.edu

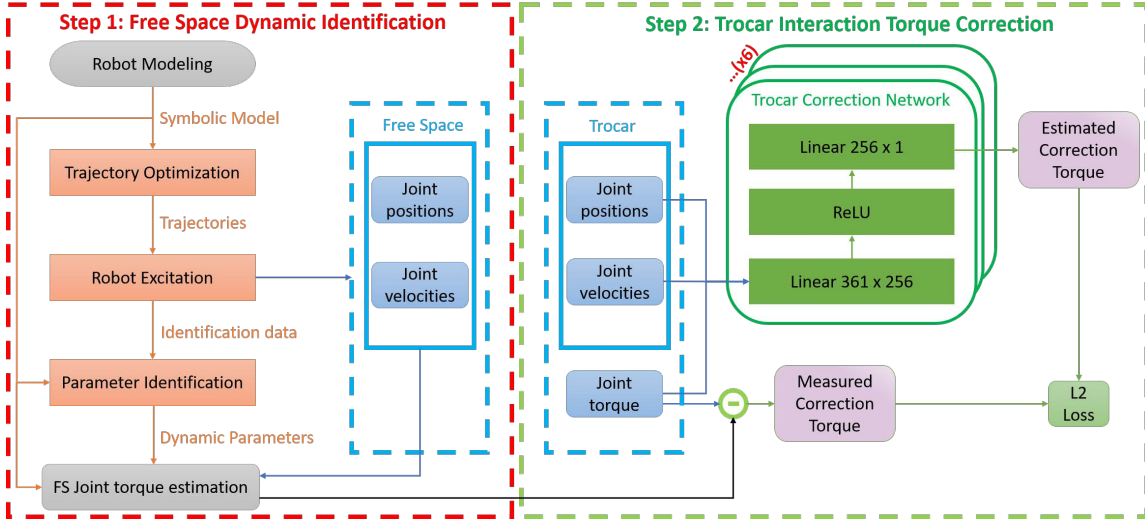


Fig. 1. Block diagrams of the proposed two-step hybrid joint torque estimation method. The first step, shown on the left, is the model-based dynamic parameter identification and free space torque estimation process. The second step, shown on the right, is the learning-based torque correction scheme about in-trocar interaction.

of this, Wang et al. develop a convex optimization-based dynamic model identification package that estimates the free space torque of the dVRK [18]. These methods construct explicit dynamic models and identify the dynamic parameters from free space torque. All these methods work well in free space dVRK torque estimation at the tooltip. However, none of them can address the challenge created by multiple points of interaction between the instrument shaft, trocar, and patient body.

B. Learning-based Torque Estimation

Alternatively, Yilmaz et al. propose a neural network (NN) based method to estimate the external force [7]. Wu et al. extend the work by adding a correction network for trocar interactions on an abdominal phantom [8]. Further work leverages transfer learning to adapt pseudo-clinical settings, such that force estimation can be applied across different robots, different instruments, and different pivot points on an abdominal phantom [12]. However, our experiments show that these learning-based methods are highly reliant on training data quality to ensure estimation accuracy. Mismatch in the training and test workspaces can result in high errors.

III. METHODS

The hybrid torque estimation method proposed in this paper has two steps, which are shown in Fig. 1. Step one is model-based dynamic identification and free space torque estimation. We move the robot along an optimized excitation trajectory without any contact and optimize the robot's dynamic parameters. Step two is learning-based trocar interaction torque compensation. We train a NN to learn patient and setup specific effects. The NN is designed to be trained with a subset of the workspace that is relevant to the procedure after the instruments have engaged and passed through the ports. Then, we validate the effectiveness of this

method by doing Cartesian force estimation, where the robot instrument tip is in contact with external objects while the shaft is inserted through the trocar and incision port.

A. Model-based Free Space Dynamic Identification and Torque Estimation

To address free space joint torque estimation, we first aim to use the dynamic parameter identification method proposed in [15], [18], [19] to obtain the dynamic model of the patient-side robot. The dynamic identification can be achieved through the following five steps:

- Build the symbolic kinematic model of the PSM based on the modified DH parameters [20] of the PSM
- Construct the symbolic dynamic model of the PSM using Euler-Lagrangian approach [21], [22]
- Calculate the optimized excitation trajectory based on the given joint constraints and the dynamic model
- Run the excitation trajectory on the physical PSM and collect kinematic and dynamic data through dVRK
- Solve for the model parameters using the least square and optimization methods

1) *Kinematic Modeling of the PSM*: To build the relationship between the robot joint motion and the torque of each motor, we define four different types of joint coordinates:

- \mathbf{q}^d : the joint coordinates obtained from the dVRK [13] package with coupled wrist joints
- \mathbf{q} : the joint coordinates used in the kinematic modeling with uncoupled wrist joints
- \mathbf{q}^m : the equivalent motor coordinates at joints
- \mathbf{q}^c : the complete joint coordinates, defined as $\mathbf{q}^c = [\mathbf{q}^T (\mathbf{q}^m)^T]^T$

The frame definition of the PSM is shown in Fig. 2, and the corresponding parameters are shown in Table I. Here we follow the convention from [18] and establish the following

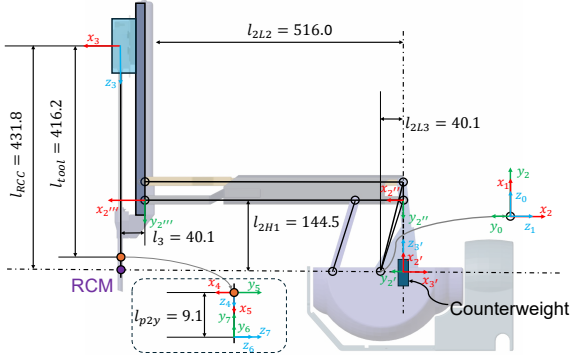


Fig. 2. The Planar View of the Frame definition for the PSM. The unit of the link lengths is mm.

TABLE I

DH TABLE OF THE PSM MODIFIED FROM [18]. TWO PASSIVE JOINTS OF THE PARALLELOGRAM MECHANISM OF THE SECOND JOINT REMOVED.

i	Ref	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	$\frac{\pi}{2}$	0	$q_1 + \frac{\pi}{2}$
2	1	0	$-\frac{\pi}{2}$	0	$q_2 - \frac{\pi}{2}$
2'	2	l_{2L3}	0	0	$\frac{\pi}{2}$
2''	2'	l_{2H1}	0	0	$\frac{\pi}{2} - q_2$
2'''	2''	l_{2L2}	0	0	q_2
3	2'''	l_3	$-\frac{\pi}{2}$	$q_3 + l_{c2}$	0
3'	2	l_{2L3}	$-\frac{\pi}{2}$	q_3	0
4	3	0	0	l_{tool}	q_4
5	4	0	$\frac{\pi}{2}$	0	$q_5 + \frac{\pi}{2}$
6	5	l_{p2y}	$-\frac{\pi}{2}$	0	$q_6 + \frac{\pi}{2}$
7	5	l_{p2y}	$-\frac{\pi}{2}$	0	$q_7 + \frac{\pi}{2}$
M_6	-	0	0	0	q_6^m
M_7	-	0	0	0	q_7^m
F_{67}	-	0	0	0	$q_6 - q_7$

Note for the table: (1) Ref stands for the reference link frame of link i . (2) a_{i-1} , α_{i-1} , d_i and θ_i are the modified DH parameters of link i . (3) The parameters of link inertia, motor inertia, joint friction and spring stiffness are omitted since they are the same as [18]. (4) M_6 and M_7 correspond to the modeling of motors 6 and 7, respectively. F_{67} corresponds to the modeling of the relative motion between links 6 and 7. (5) $l_{c2} = -l_{RCC} + l_{2H1}$.

relationships among q , q^d and q^m :

$$q_{1-5} = q_{1-5}^d, q_{1-4}^d = q_{1-4}^m \quad (1a)$$

$$\begin{bmatrix} q_6^d & q_7^d \end{bmatrix}^T = \begin{bmatrix} \frac{q_6 + q_7}{2} & (-q_6 + q_7) \end{bmatrix}^T \quad (1b)$$

$$\begin{bmatrix} q_5^d \\ q_6^d \\ q_7^d \end{bmatrix} = A_m^d \begin{bmatrix} q_5^m \\ q_6^m \\ q_7^m \end{bmatrix} = \begin{bmatrix} 1.0186 & 0 & 0 \\ -0.8306 & 0.6089 & 0.6089 \\ 0 & -1.2177 & 1.2177 \end{bmatrix} \begin{bmatrix} q_5^m \\ q_6^m \\ q_7^m \end{bmatrix} \quad (1c)$$

2) *Dynamic Modeling of the PSM*: For each link i , it has the mass m_i , the position of the center of mass (COM) \mathbf{r}_i with respect to link frame i and the inertia matrix \mathbf{I}_i with respect to the COM. Given the above parameters, we can calculate the first moment of inertia $\mathbf{M}_i \in \mathbb{R}^3$ and the inertia matrix $\mathbf{L}_i \in \mathbb{R}^{3 \times 3}$ with respect to the link frame i :

$$\begin{aligned} \mathbf{M}_i &= m_i \mathbf{r}_i \\ \mathbf{L}_i &= \mathbf{I}_i + m_i ([\mathbf{r}_i]_{\times})^T [\mathbf{r}_i]_{\times} \end{aligned} \quad (2)$$

where $[\cdot]_{\times}$ is the operator to construct a skew-symmetric matrix from a given vector.

Complete details of the dynamic parameters can be found in [18]. Essentially, we use barycentric parameters [23] to denote link inertia as δ_{L_i} and other parameters as δ_{O_i} . We combine all parameter vectors of n joints and obtain the overall dynamic parameters δ :

$$\delta = [\delta_{L1}^T \delta_{O1}^T \cdots \delta_{Ln}^T \delta_{On}^T]^T \quad (3)$$

To address dynamic modeling for the dVRK PSM, we use the Euler-Lagrange method [21] since it can handle joint constraints. The Lagrangian L is calculated by the difference between the kinetic energy K and potential energy P of the robot, i.e. $L = K - P$. Motor inertia, joint frictions, and spring stiffness are not included in L and are modeled separately. The torques $\tau_{L_i}^m$, τ_M^m , τ_f^c and τ_s^c , which are caused by link inertia, motion inertia, joint frictions, and spring stiffness, can be computed following the formulas in [18].

The overall torques τ^m are given by:

$$\begin{aligned} \tau^m &= \tau_{L1}^m + \tau_M^m (\ddot{\mathbf{q}}^m) + \frac{\partial \mathbf{q}^c}{\partial \mathbf{q}^m} [\tau_f^c (\dot{\mathbf{q}}^c) + \tau_s^c (\mathbf{q}^c)] \\ &= \mathbf{H}(\mathbf{q}^m, \dot{\mathbf{q}}^m, \ddot{\mathbf{q}}^m) \delta \end{aligned} \quad (4)$$

To calculate the base parameters, which are a minimum set of dynamic parameters that can fully describe the dynamic model of the robot, we use QR decomposition with pivoting [24]. Then, we can introduce a permutation matrix $\mathbf{P}_b \in \mathbb{R}^{n \times b}$, where n is the number of standard dynamic parameters and b is the number of base parameters. Therefore, we can calculate the base parameters δ_b and the corresponding regressor \mathbf{H}_b as:

$$\delta_b = \mathbf{P}_b^T \delta, \mathbf{H}_b = \mathbf{H} \mathbf{P}_b \quad (5)$$

3) *Excitation Trajectory Optimization*: Periodic excitation trajectories based on Fourier series [24] are used to generate data for dynamic model identification. These trajectories minimize the condition number of the regression matrix \mathbf{W}_b for the base parameters δ_b subject to the lower and upper boundary constraints of joint positions & velocities and the robot Cartesian positions.

$$\mathbf{W}_b = [\mathbf{H}_b(\mathbf{q}_1^m, \dot{\mathbf{q}}_1^m, \ddot{\mathbf{q}}_1^m), \dots, \mathbf{H}_b(\mathbf{q}_{n_s}^m, \dot{\mathbf{q}}_{n_s}^m, \ddot{\mathbf{q}}_{n_s}^m)]^T \quad (6)$$

where \mathbf{q}_i^m is the motor joint coordinate at i_{th} sampling point and n_s is the number of sampling points. The joint coordinate q_j^m of motor j can be calculated by

$$q_j^m(t) = q_{oj}^m + \sum_{k=1}^{n_H} \left[\frac{a_{jk}}{k\omega_f} \sin(k\omega_f t) - \frac{b_{jk}}{k\omega_f} \cos(k\omega_f t) \right] \quad (7)$$

where $\omega_f = 2\pi f_f$ is angular component of the fundamental frequency f_f , n_H is harmonic number of the Fourier series, a_{jk} & b_{jk} are amplitudes of the j_{th} -order sine and cosine function, q_{oj}^m is position offset, and t is time.

4) *Parameter Identification*: To identify the dynamic parameters, we move the robot along the excitation trajectories generated via the method described in the previous subsection. Data is collected at each sampling time to obtain the regression matrix \mathbf{W} and the dependent variable vector ω .

$$\begin{aligned} \mathbf{W} &= [\mathbf{H}(\mathbf{q}_1^m, \dot{\mathbf{q}}_1^m, \ddot{\mathbf{q}}_1^m), \dots, \mathbf{H}(\mathbf{q}_{n_s}^m, \dot{\mathbf{q}}_{n_s}^m, \ddot{\mathbf{q}}_{n_s}^m)]^T \\ \omega &= [\tau_1^m, \tau_2^m, \dots, \tau_{n_s}^m]^T \end{aligned} \quad (8)$$

τ_i^m is the motor torque at i_{th} sampling point.

Then, we convert the identification problem into an optimization problem [18], which minimizes the cost function Q with respect to the decision vector δ :

$$Q = \|\mathbf{W}\delta - \omega\|^2 \quad (9)$$

B. Learning-based Trocar Interaction Torque Correction

The second step of our method adapts the trocar correction proposed in [8] to the model-based dynamics identification outputs. We insert the robot instrument through a trocar into an abdominal phantom such that the RCM of the robot is at the incision port. The incision port applies interaction force to the instrument shaft and trocar, which can be regarded as an extra constraint to the robot and effectively changes the system dynamics. The free space torque identification in step one is no longer accurate in this case. However, this extra constraint is hard to model, since in the surgical scenario, the patient body wall is mechanically anisotropic. Furthermore, compliance, strain, and stiffness differ between patients. These factors are subject to the incision location, tissue property, patient age, etc. [11]. Therefore, to compensate for this additional interaction, as shown in Fig. 1, we train a trocar correction network for each of the six joints based on the residual error between the torque estimated by the free space dynamic model and the measured torque. This network accounts for the interaction between the instrument and the body wall but further work is needed to investigate collisions at other points. The input to each network is a window of concatenated measurements of the previous positions and velocities of all joints, and the current free space estimated torque for each joint. Each network is a multi-layer perceptron consisting of two linear layers with 256 hidden dimensions and a ReLU activation layer in between them. We follow the design in [8] to minimize the network size to ensure the intra-operative data collection time is clinically feasible. The corrected torque is the sum of the network output and free space network estimation.

C. Cartesian Force/Torque Estimation

To get the torque acting upon the external environment ($\hat{\tau}_{ext}$), we subtract the free space (FS) torque estimation with trocar correction ($\hat{\tau}_{FS} - \hat{\tau}_{corr}$) from the measured torque (τ) at every time point. Then, we multiply the inverse transpose of the spatial Jacobian J^{-T} with the external torque to derive the external Cartesian force. Note that the spatial Jacobian here is full-ranked.

$$\hat{F}_{ext} = J^{-T} \hat{\tau}_{ext} = J^{-T} (\tau - (\hat{\tau}_{FS} - \hat{\tau}_{corr})) \quad (10)$$

The force estimation scheme is shown in Fig. 3.

IV. EXPERIMENTAL SETUP

A. Dynamic Identification and Free Space Torque Estimation

1) *Excitation Trajectory Generation*: For each PSM, we generate two independent excitation trajectories for identification and testing purposes. To solve the Eq. 7, we select the harmonic number n_H as 6 and the fundamental

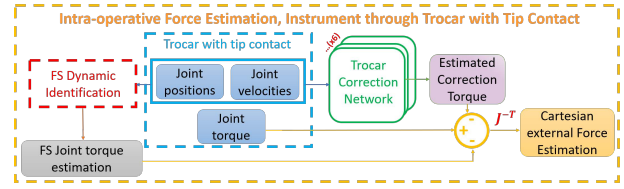


Fig. 3. The block diagram of Cartesian force estimation. We use the proposed model to estimate the torque required to achieve the measured joint positions and velocities with no contact. Then, we subtract it from the measured torque and use the Jacobian to convert the difference into force exerted on the environment.

frequency f_f of 0.18 Hz. Furthermore, given the pre-defined joint constraints, we can compute the optimized trajectory using pyOpt [25].

2) *Data Collection and Data Processing*: To collect the data, we run the excitation trajectory with a frequency of 200 Hz using the dVRK software framework on a dVRK PSM with a Large Needle Driver instrument. We collect the joint positions, velocities, and torques via subscribing to the dVRK Robot Operating System topics [26]. Then, we calculate the joint acceleration using the numerical differentiation of the velocities. Lastly, we implement a sixth-order low-pass filter to preprocess the data [18].

3) *Solve the model parameters*: To obtain uniformly precise identification results for all joints, we multiply a weight $w_i = \frac{1}{\max(\tau_i^m) - \min(\tau_i^m)}$ to the cost function Q_i of each motor joint i . We optimize the model parameters using the CVXPY package [27] with the SCS solver [28].

B. Trocar Correction and Cartesian Force Estimation

We collect about 4 minutes training dataset and split it into 80%/10%/10% as train/validation/test subsets, with a robot instrument inserted through a trocar in an incision port of the abdominal phantom, to learn the correction factor. The correction network is trained for 400 epochs with an initial learning rate of 0.0001, window size of 5, and batch size of 10000. We use Adam and decay learning rate on plateau. Training is done in 15.28 s on an Nvidia A4500 graphic card.

We then collect an in-trocar, with tip contact test dataset for Cartesian force estimation. We place a Gamma Force/Torque sensor (ATI Industrial Automation, Apex, NC, USA) in the workspace of the robot and mount a 3D-printed shaft on top. We teleoperate the instrument to poke and manipulate the shaft to log external six degrees-of-freedom Cartesian force data, as the ground truth, to compare with our method estimation. The experimental setup is shown in Fig. 4.

We use two metrics, normalized root-mean-square errors (NRMSE) and root-mean-square errors (RMSE) to evaluate the performance of the proposed method. The NRMSE between the actual and estimated joint forces/torques are calculated using the formula given in [15].

V. RESULTS AND DISCUSSION

We conduct experiments to compare the force estimation accuracy of our proposed hybrid method with the model-based-only and learning-based-only methods [8], [18]. We

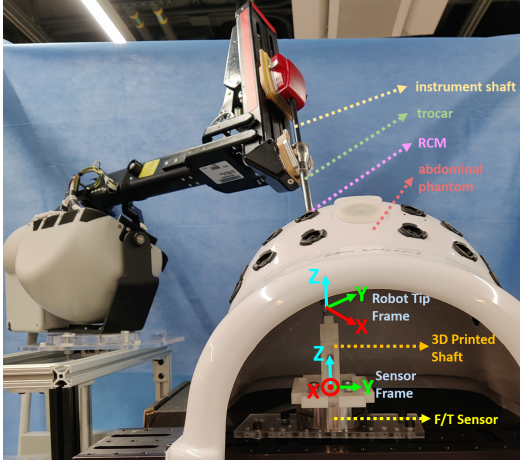


Fig. 4. Experimental Setup. The instrument shaft is inserted through a trocar to the abdominal phantom cavity. The trocar is plugged into an incision port on the phantom, which is also the RCM of the robot in this configuration. The instrument tip manipulates the 3D-printed shaft and the force/torque sensor beneath it logs the force that the gripper exerts. Note that there is a 45-degree angle bias between the robot tip frame and the sensor frame, with respect to the sensor frame Z-axis.

perform free space torque estimation when there is training-testing dataset mismatch, and both torque estimation and Cartesian force estimation when the instrument is in trocar.

A. Learning-based Method Suffers from Train, Test Data Mismatch

Our proposed hybrid method combines the robustness provided by the physical feasibility from the model-based part, and the adaptability provided by the data-driven compensation from the learning-based part. To validate this, we show that the pure learning-based method can be less reliable in certain conditions. Note that the pure learning-based method proposed in [8] relies highly on the input training data quality, i.e. whether the robot joint profile in the training dataset can generalize to and cover the workspace in the test dataset. We collect a 15-minute training dataset and partition it into 80%/20% train/validation sets. Next, we collect a test dataset in a different workspace.

Fig. 5 visualizes the different workspace of the joint positions in the training and testing datasets. As our step 2 network is too simple to capture the full workspace, we use the Long Short-Term Memory network proposed in [8] in this section for a fair comparison with the model-based method. Finally, we estimate torque for the test dataset using the learning-based and optimization-based models, respectively. Results are visualized in Fig. 6, and the RMSE is measured for estimation accuracy evaluation, shown in Table. II.

The results show that the learning-based method performs poorly when the training and test datasets are not in the same workspace. The training dataset needs to be well-crafted to guarantee test set torque estimation accuracy.

In contrast, the parameters identified through the model-based method are more generalizable than the learning-based method. Unlike the learning-based method, the model-based method considered the physical feasibility of the robot.

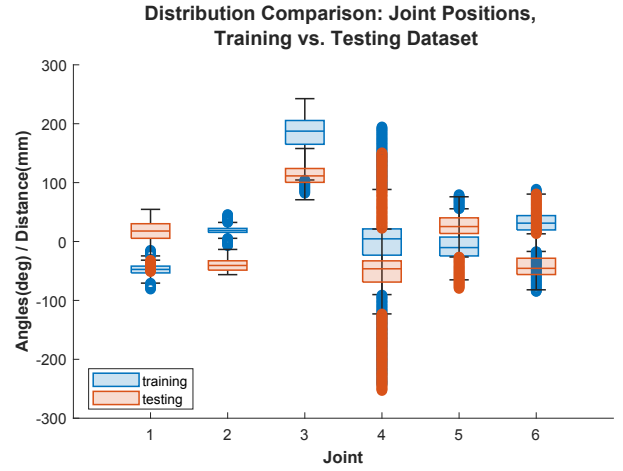


Fig. 5. Joint position profile distribution of training and testing dataset collected in different workspace, which causes a mismatch.

Free Space Torque Estimation, Mismatched Testing Data, Learning-based vs. Model-based

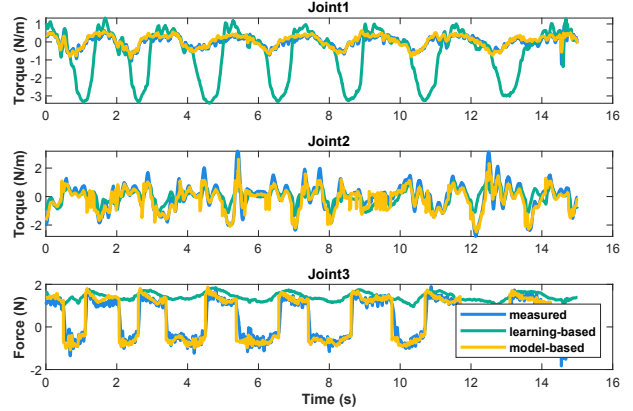


Fig. 6. Learning-based vs. model-based free space torque estimation. Since there is a mismatch between training and testing datasets, the learning-based method has high error. Model-based method estimates well, as the dynamic parameters are identified following the proposed scheme.

Therefore, compared to the uncertainty that the training dataset brings in, the performance of torque estimation by model-based, optimized excitation trajectory is more robust.

B. Phantom Force Estimation: Model-based vs. Hybrid

We use the experimental setup mentioned in the previous section to achieve trocar correction and Cartesian force estimation and compare the performance with the model-based-only method. We first compare the torque estimation of the in-trocar, no-tip-contact test dataset. Results are shown in Fig. 7. Next, we compare the Cartesian force estimation of the in-trocar, with-tip-contact test dataset, as shown in Fig. 8. The NRMSE and RMSE are presented in Table. III and Table. IV, respectively. The results depict that the model-based method fails to correct extra torque caused by trocar interaction, while the hybrid method does.

We observe from Fig. 7 that our proposed hybrid method estimates better than model-based-only method. The NRMSE

TABLE II

RMSE OF JOINT TORQUE($N \cdot m$)/FORCE(N) ESTIMATION:
LEARNING-BASED VS. MODEL-BASED WITH MISMATCHED TRAINING
AND TEST DATA

Method	τ_1	τ_2	f_3
Learning-based	0.84	0.49	1.38
Model-based	0.22	0.39	0.42

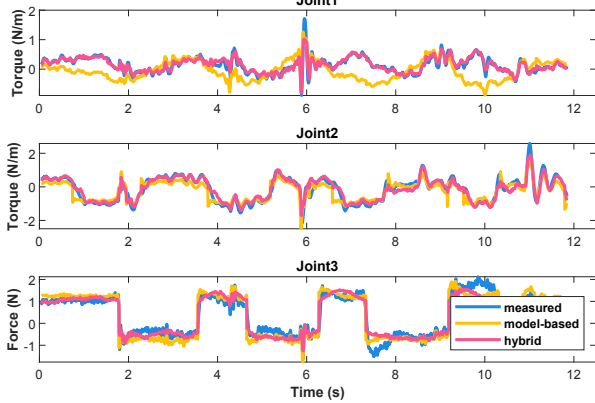
Joint Torque Estimation, In Trocar, Without Tip Contact,
Model-based vs. Hybrid

Fig. 7. Model-based vs. Hybrid torque estimation with trocar-only contact. Note that the hybrid method contributes the most significant correction on joints 1 and 2. That is because the trocar interaction force mainly exists on the tangent plane of the incision port. Therefore, along the norm of the port, which is the force direction of joint 3, the correction works not as effective as the other two.

TABLE III

NRMSE \dagger AND RMSE* OF IN TROCAR, NO TIP CONTACT JOINT
TORQUE($N \cdot m$)/FORCE(N) ESTIMATION: MODEL-BASED VS. HYBRID

Method	τ_1	τ_2	f_3	τ_4	τ_5	τ_6
Model-based \dagger	0.14	0.07	0.10	0.12	0.35	0.43
Hybrid \dagger	0.067	0.04	0.09	0.09	0.092	0.13
[12] \dagger	0.062	0.10	0.16	0.06	0.040	0.048
Model-based*	0.38	0.35	0.45	0.0049	0.07	0.066
Hybrid*	0.20	0.29	0.44	0.0067	0.03	0.025

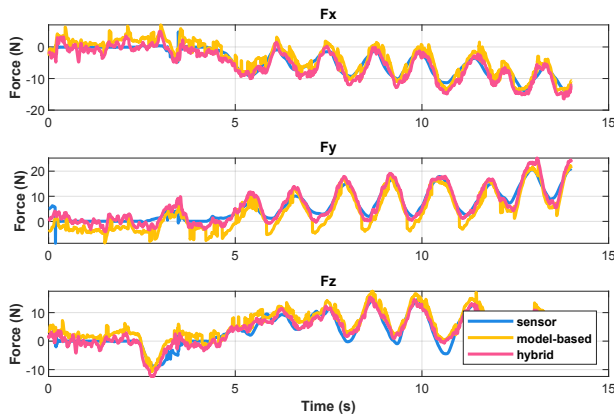
Cartesian Force Estimation, In Trocar, With Tip Contact,
Model-based-only vs. Hybrid

Fig. 8. Model-based vs. Hybrid Cartesian Force Estimation, with trocar-abdominal phantom contact and robot tip-F/T sensor contact.

TABLE IV

NRMSE \dagger AND RMSE* OF IN TROCAR, WITH TIP CONTACT
CARTESIAN FORCE ESTIMATION: MODEL-BASED VS. HYBRID

Method	$F_x(N)$	$F_y(N)$	$F_z(N)$
Model-based \dagger	0.086	0.098	0.098
Hybrid \dagger	0.062	0.064	0.069
Model-based*	2.7	3.5	3.3
Hybrid*	1.9	2.2	2.3
[8]*	2.3	1.5	3.3

of our method is in general within 10%, which is comparable to the results presented in [12]. We also observe from Fig. 8 that our method effectively estimates Cartesian force with tip contact. The RMSE for all directions are around 2 N, which is comparable to the results presented in [8]. As [8], [12] used pure learning-based methods that rely heavily on the quality of training data, these results reflect that the proposed method combines the advantages of both the learning-based and model-based methods. Inflection points are difficult to model for all tested methods. We observe that both the model and NN underestimate large changes and tend to optimize for average result. Lastly, the NN in the correction network may suffer from the same generalization issues that we observe in the full workspace. We posit that it is easier to collect high quality data for the correction network as the workspace is restricted compared to the other pure learning-based methods in [8], [12]. As patient body wall interactions are difficult to model, particularly as data must be collected intraoperatively, the NN is a reasonable trade-off. Additionally, patient body wall interactions may change throughout surgery as insufflation level changes and the NN has the potential to adapt through life-long learning.

VI. CONCLUSION

In this paper, we introduce a hybrid force estimation framework, combining model-based free space torque estimation, learning-based trocar interaction torque correction, and Cartesian force estimation. We show that this method can compensate for the interaction force between the trocar and the patient body, which the pure model-based method cannot address. We also show that this method can be more robust than the pure learning-based method since the model-based trajectory optimization and parameter fitting do not depend on sampling the entire workspace of the robot. In the future, we plan to further investigate the framework universality, such as deploying the methods on other robots. Our hybrid method for sensorless force combines the advantages of model and learning-based methods to provide generalizability and also account for environmental conditions.

ACKNOWLEDGEMENT

Thanks to Dale Bergman and Alessandro Gozzi for all the hardware support, to Peter Kazanzides and Anton Deguet for dVRK guidance. This work was supported in part by an Intuitive Technology Research Grant.

REFERENCES

- [1] T. Williamson and S.-E. Song, "Robotic surgery techniques to improve traditional laparoscopy," *JSL: Journal of the Society of Laparoscopic & Robotic Surgeons*, vol. 26, no. 2, 2022.
- [2] A. M. Okamura, "Methods for haptic feedback in teleoperated robot-assisted surgery," *Industrial Robot: An International Journal*, vol. 31, no. 6, pp. 499–508, 2004.
- [3] A. M. Okamura, L. N. Verner, C. E. Reiley, and M. Mahvash, "Haptics for robot-assisted minimally invasive surgery," in *Robotics Research: The 13th International Symposium ISRR*. Springer, 2011, pp. 361–372.
- [4] G. S. Fischer, T. Akinbiyi, S. Saha, J. Zand, M. Talamini, M. Marohn, and R. Taylor, "Ischemia and force sensing surgical instruments for augmenting available surgeon information," in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob*. IEEE, 2006, pp. 1030–1035.
- [5] Z. Chua and A. M. Okamura, "A modular 3-degrees-of-freedom force sensor for robot-assisted minimally invasive surgery research," *Sensors*, vol. 23, no. 11, p. 5230, 2023.
- [6] G. A. Fontanelli, L. R. Buonocore, F. Ficuciello, L. Villani, and B. Siciliano, "An external force sensing system for minimally invasive robotic surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1543–1554, 2020.
- [7] N. Yilmaz, J. Y. Wu, P. Kazanzides, and U. Tumerdem, "Neural network based inverse dynamics identification and external force estimation on the da vinci research kit," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1387–1393.
- [8] J. Y. Wu, N. Yilmaz, U. Tumerdem, and P. Kazanzides, "Robot force estimation with learned intraoperative correction," in *International Symposium on Medical Robotics (ISMR)*. IEEE, 2021, pp. 1–7.
- [9] S. Liu, L. Wang, and X. V. Wang, "Sensorless force estimation for industrial robots using disturbance observer and neural learning of friction approximation," *Robotics Comput. Integr. Manuf.*, vol. 71, p. 102168, 2021.
- [10] B. Eldridge, K. Gruben, D. LaRose, J. Funda, S. Gomory, J. Karidis, G. McVicker, R. Taylor, and J. Anderson, "A remote center of motion robotic arm for computer assisted surgery," *Robotica*, vol. 14, no. 1, pp. 103–109, 1996.
- [11] C. R. Deeken and S. P. Lake, "Mechanical properties of the abdominal wall and biomaterials utilized for hernia repair," *Journal of the mechanical behavior of biomedical materials*, vol. 74, pp. 411–427, 2017.
- [12] N. Yilmaz, J. Zhang, P. Kazanzides, and U. Tumerdem, "Transfer of learned dynamics between different surgical robots and operative configurations," *International Journal of Computer Assisted Radiology and Surgery*, vol. 17, no. 5, pp. 903–910, 2022.
- [13] 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 *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 6434–6439.
- [14] H. Sang, J. Yun, R. Monfaredi, E. Wilson, H. Fooladi, and K. R. Cleary, "External force estimation and implementation in robotically assisted minimally invasive surgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 13, no. 2, 2017.
- [15] F. Piqué, M. N. Boushaki, M. Brancadoro, E. De Momi, and A. Menciassi, "Dynamic modeling of the da vinci research kit arm for the estimation of interaction wrench," in *2019 International Symposium on Medical Robotics (ISMR)*. IEEE, 2019, pp. 1–7.
- [16] C. D. Sousa and R. Cortesao, "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.
- [17] G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Modelling and identification of the da vinci research kit robotic arms," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1464–1469.
- [18] Y. Wang, R. Gondokaryono, A. Munawar, and G. S. Fischer, "A convex optimization-based dynamic model identification package for the da vinci research kit," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3657–3664, 2019.
- [19] M. Ferro, A. Mirante, F. Ficuciello, and M. Vendittelli, "A coppeliasim dynamic simulator for the da vinci research kit," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 129–136, 2022.
- [20] W. Khalil and J. Kleinfinger, "A new geometric notation for open and closed-loop robots," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 1986, pp. 1174–1179.
- [21] 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.
- [22] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons, 2020.
- [23] 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.
- [24] M. Gautier, "Numerical calculation of the base inertial parameters of robots," *Journal of robotic systems*, vol. 8, no. 4, pp. 485–506, 1991.
- [25] 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, pp. 101–118, 2012.
- [26] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [27] S. Diamond and S. Boyd, "Cvxpy: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.
- [28] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Scs: Splitting conic solver, version 2.0. 2," 2, 2017.