

Robot Force Estimation with Learned Intraoperative Correction

Jie Ying Wu¹, Nural Yilmaz², Ugur Tumerdem², Peter Kazanzides¹

Abstract—Measurement of environment interaction forces during robotic minimally-invasive surgery would enable haptic feedback to the surgeon, thereby solving one long-standing limitation. Estimating this force from existing sensor data avoids the challenge of retrofitting systems with force sensors, but is difficult due to mechanical effects such as friction and compliance in the robot mechanism. We have previously shown that neural networks can be trained to estimate the internal robot joint torques, thereby enabling estimation of external forces on the da Vinci Research Kit (dVRK). In this work, we extend the method to estimate external Cartesian forces and torques, and also present a two-step approach to adapt to the specific surgical setup by compensating for forces due to the interactions between the instrument shaft and cannula seal and between the trocar and patient body. Experiments show that this approach provides estimates of external forces and torques within a mean root-mean-square error (RMSE) of 1.8 N and 0.1 Nm, respectively. Furthermore, the two-step approach can add as little as 5 minutes to the surgery setup time, with about 4 minutes to collect intraoperative training data and 1 minute to train the second-step network.

I. INTRODUCTION

Robotic surgery has enabled better surgeon ergonomics and patient care. Yet this comes at the cost of increased distance between surgeon and patient, which also results in a lack of direct haptic feedback to the surgeon [1]. Although surgeons can learn to use visual feedback to adjust for the lack of haptic feedback, restoring it can make telerobotic surgery easier to learn, more intuitive, and precise.

Adding force sensing capability to surgical robots has been an active research subject and multiple alternative solutions have been suggested. Madhani et al. performed force estimation with joint current/torque measurements of the Black Falcon surgical robot, but it has been reported that inertial forces of the robot are dominant in torque measurements, especially in free motion [2]. When reflected to the operator they can be annoying. In addition to the free motion inertial forces, trocar interactions introduce further forces that should not be reflected in instrument tip interaction feedback. Fig. 1 shows potential contributors, the trocar and the cannula seal, that interact with the instrument along its length.

In our previous works, we developed a self-supervised force estimation approach [3] on the da Vinci Research Kit (dVRK) [4], an open-source surgical robotic research platform based on the da Vinci Surgical System (Intuitive

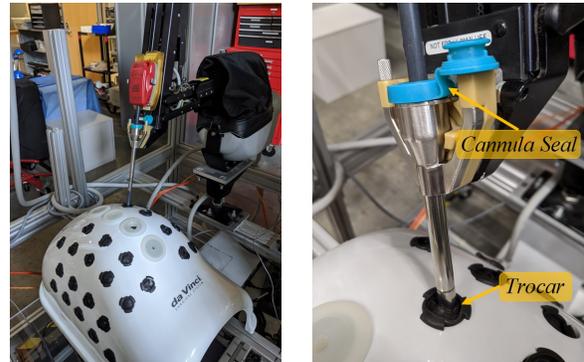


Fig. 1. Left: Abdominal phantom with an instrument inserted through the trocar; Right: Closeup view of the trocar and cannula seal.

Surgical Inc., Sunnyvale CA, USA). The method estimated forces in free space and may not accurately account for trocar interactions in patients. While one could train the networks with surgery-specific data that includes trocar interactions, the time to collect sufficient training data and train the network may be clinically infeasible.

In this paper, we extend our use of neural networks to identify and eliminate trocar interaction forces. As the trocar interaction forces are surgery-specific, we use two-step deep learning techniques to make this identification feasible in a clinical scenario. The proposed technique is self-supervised to not require additional setup in the clinic. Furthermore, we extend our prior work to estimate Cartesian torques in addition to forces. We test the proposed method on the dVRK system with an abdominal phantom.

II. RELATED WORKS

One way to provide force feedback is to place force sensors in the patient side manipulators. Berkelman et al. [5] and Siebold et al. [6] developed three and six axis miniature force sensors, respectively, and attached them to the tips of specially developed surgical instruments. Due to the difficulties in manufacturing such sensors, other researchers have attached flexible capacitive [7], strain gauge based [8], or fiber-optics based [9] sensors on surgical grippers. However, the force/torque sensing degrees of freedom are limited in these approaches. Shahzada et al. place four fiber Bragg grating sensors on the shaft of the instrument to measure lateral forces [10]. Since these do not measure force in the insertion direction, they show limited benefit for palpation tasks. Intra-corporeal sensor based approaches often present difficulties in manufacturing small, accurate, sterilizable, and robust multi-degrees of freedom (DOF) sensors [11].

¹Dept. of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA (email: {jyeying, pkaz}@jhu.edu)

²Dept. of Mechanical Engineering, Marmara University, Istanbul, Turkey (email: nural.yilmaz@marun.edu.tr, ugur.tumerdem@marmara.edu.tr)

To overcome these problems, some researchers have proposed the use of force sensors placed at the noninvasive (extra-corporeal) part of the robot. The drawback is that since the robot interacts with the patient body at the trocar, these forces would also be registered by the sensor. In order to filter these forces, the “overcoat” method was proposed by Shimachi et al. [12] by introducing a second trocar sensor. This method has been further developed by Willaert et al. [11] and Schwab et al. [13]. Since these methods require bulky force sensors placed at the trocar, Fontanelli et al. consider the use of fiber optic sensors in the trocar [14]. These methods can provide accurate estimates of forces in the Cartesian XYZ directions of the tool tip. However, the wrist bending torques cannot be estimated/measured.

Another approach is to estimate forces based on robot dynamics identification. Mahvash et al. performed friction compensation on a da Vinci system [15]. Haghhighipناه et al. estimated cable tension with dynamic identification on the Raven II system [16]. These papers have reported force estimation in 1 DOF. Sang et al. [17] and Piqué et al. [18] used more comprehensive explicit model-based dynamic identification in the estimation of external forces on the dVRK in 3 and 6 DOF, respectively. Although these model-based approaches have good performance, they do not take trocar and cannula seal interactions into consideration.

With the recent advances in machine learning methods, some researchers have applied deep learning to force estimation in robotic surgery. O’Neill et al. [19] and Guo et al. [20] trained deep neural networks to estimate gripper forces on 1 DOF instruments, by using joint and force sensor measurements in training. Yu et al. [21] and Tran et al. [22], use a similar approach for Cartesian force estimation in three axes. However, these approaches require the use of a force sensor for ground truth. The use of a force sensor for training in a patient would be difficult, limiting the applications of these approaches in a clinical scenario. Additionally, the placement of the force sensor may introduce biases [22].

This paper focuses on adding a self-supervised correction step to account for trocar interactions. Our implementation adds the proposed correction step to our prior neural network [3], but it could be used with any of the above model or learning-based torque estimation methods.

III. METHODS

To account for trocar interaction forces, we propose a two-step learning scheme. Fig. 2 shows the proposed setup. The first network, described in Section III-A, follows our previous work in estimating free space dynamics [3]. It is trained extensively for robot joint torque identification in the full robot workspace. It is followed by another network, described in Section III-B, that learns patient and setup specific effects. The second network is designed to be trained with a subset of the workspace that is relevant to the procedure after the robot has been docked and the instruments have been placed through the ports.

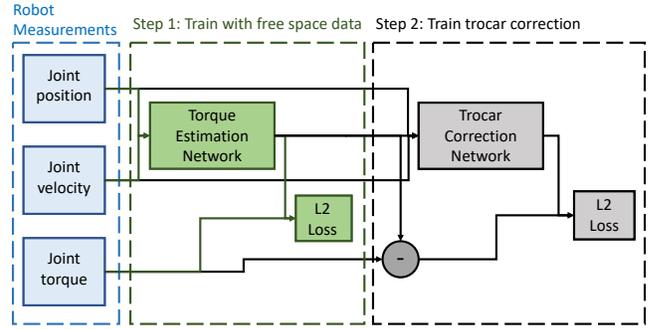


Fig. 2. Block diagram of two-step correction network. Step 1 estimates the torque for robot motion in free space. Step 2 adds a correction factor for the torque to account for the cannula seal and trocar interaction. It is trained from motion inside the trocar and learns the difference between the free space estimation and the measured torque.

A. Step 1: Joint Torque Identification Network

The goal of the step 1 network is to learn the joint torque during free space motion. We train one network for each joint. Following the comparison in [23], we adopt an LSTM as our base network for torque identification as it was able to better capture the hysteresis in the system. The input to the network consists of a sequence of positions and velocities of all the joints. Joint positions are read from encoders and joint velocities are calculated according to [24]. The network architecture, shown in Fig. 3, consists of an LSTM layer with 128 hidden dimensions followed by two fully connected layers with RELU activation.

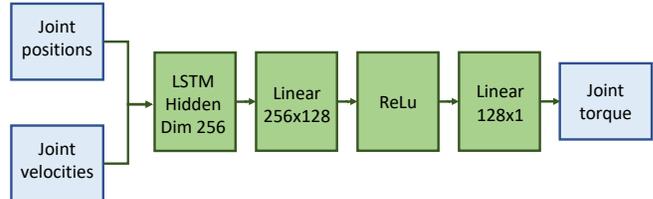


Fig. 3. Free space network architecture. The input consists of a sequence of the positions and velocities for all joints. This goes to an LSTM layer, followed by two linear layers.

B. Step 2: Compensation for Trocar Interaction Forces

We insert the da Vinci Patient Side Manipulator (PSM) instrument inside an abdominal phantom such that its remote center of motion is at the insertion port. We also add a cannula seal to hold the instrument in the middle of the cannula. The setup is shown in Fig. 4. Since the trocar and the cannula seal introduce additional points of contact with the external environment, the torque identification learned on free space data may no longer be accurate. To correct for this additional interaction, we train another network for each joint based on the residual error between the measured torque and the torque predicted by the free space network, as shown in Fig. 2.

The second network has competing goals of making accurate predictions and learning from a small training set. While

neural networks typically require a lot of data to train, the trocar interactions are unique for each surgery and can only be collected after surgical instruments have been inserted into a patient. We have to limit the surgery-specific training data set to data on the order of minutes, so that it is feasible to collect in the operating room. The training time must also be suitably short to fit into the clinical workflow.

Since LSTMs are generally slow to train and require more data, we use a feed-forward network for learning the surgery-specific trocar correction. To capture some time-series information without the heavy computation cost of LSTMs, we use a window of previous position and velocity measurements of all joints and the predicted free space torques as input. The network has two linear layers, with hidden dimension of 256 nodes. The small capacity is sufficient to learn the limited workspace for a particular patient setup and avoids overfitting to small training sets. We use ReLU activation between the hidden and output layers. The output of the network is added to the free space network prediction as a correction factor.

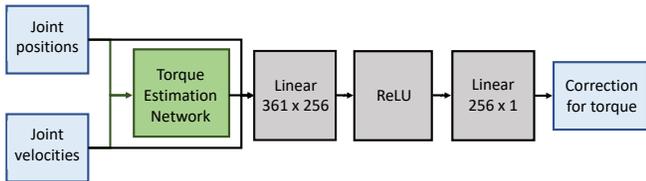


Fig. 4. Trocar correction network architecture. The input consists of a sequence of the positions and velocities for every joint, and the output of the free space torque estimation network (Fig. 3) for the specific joint. This passes through two linear layers.

C. Force and Torque Estimation

We refer readers to our previous work [3] for a detailed description of the torque identification task. Briefly, once the free space and correction networks have learned the torque to move the robot without contact, we can subtract their estimates ($\hat{\tau}_{step1} + \hat{\tau}_{step2}$) from the measured torque (τ) at every time point to get the torque acting upon the external environment ($\hat{\tau}_{ext}$).

$$\hat{\tau}_{ext} = \tau - \hat{\tau}_{step1} - \hat{\tau}_{step2} \quad (1)$$

We multiply the external torque with the spatial Jacobian at every time step to calculate the force the robot exerts.

$$\hat{F} = J^{-T} \hat{\tau}_{ext} \quad (2)$$

IV. EXPERIMENTAL SETUP

A. Free Space Torque Identification Network Training

Similar to the setup in [3], we move the robot instrument in free space to collect training data. We collect about an hour of data without any contact, for both with and without

the cannula seal. We use 80% of the data as our training set and 10% each for the test and validation sets. Extending our previous work, which only explored a limited workspace, we collected as full a workspace as possible and the workspace analysis is shown in Fig. 5. We record the data at 1 kHz using Rosbag and down-sample the data to 200 Hz. We implement the networks in PyTorch [25] and use the Adam optimizer with initial learning rate set to 0.001. We train the networks for 1000 epochs with a scheduler to decay on plateau.

B. Trocar and Cannula Seal

To learn the trocar effects, we collect about 20 min of interactions inside the cannula and split it as 80%/10%/10% into train, validation, and test sets. We insert the cannula into a trocar in the abdominal phantom and place a cannula seal on top. Then, we attach the cannula to the dVRK and insert the instrument through the cannula. The abdominal phantom limits the workspace of the robot. A comparison of the free space and the trocar workspace is shown in Fig 5.

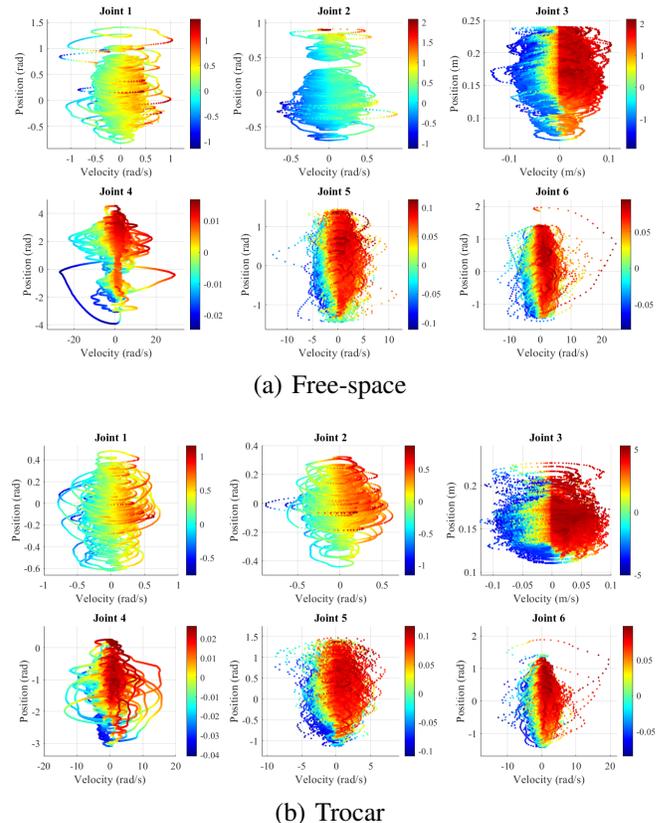


Fig. 5. Joint states and torques (shown by colors) used in training for (a) free-space and (b) trocar. The trocar workspace is confined by the phantom.

Since the free space network trained with the cannula seal matches the test conditions better, we expect that it will perform better than the free space network trained without the cannula seal. However, correcting the latter free space network enables us to estimate how well the correction schemes can generalize to test setups that are extremely different from training. The correction network is trained

for 400 epochs with an initial learning rate of 0.0001. The window size was set to be 30 empirically through grid search. At 200Hz, this gives 0.15s of context for each torque prediction. For comparison, we also train an LSTM on patient specific data (**Troc**), even though this may not be clinically viable due to the longer data collection and training times. These cases are described in Table I.

TABLE I
TESTED NETWORK CONFIGURATIONS

Troc	LSTM trained directly from trocar data using the network described in III-A.
Base	LSTM trained on data collected without the trocar and without the cannula seal as described in III-A
Seal	LSTM trained on data collected without the trocar but with a cannula seal using the network described in III-A
Corr	Correction net trained from trocar data using second network as described in III-B

C. 6 Degrees of Freedom 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 structure on top as shown in Fig. 6. This structure enables us to use the da Vinci instrument to grasp a handle and apply both forces and torques to evaluate 6 degrees of freedom.

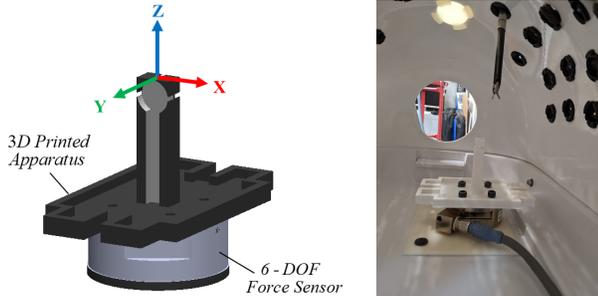


Fig. 6. CAD model (left) and test setup (right) used to collect 6 degree of freedom forces and torques. The test setup shows the sensor mounted inside the abdominal phantom shown in Fig. 1.

The coordinate systems of the force sensor and the robot are aligned, with just an offset in the Z direction. Thus, we can use a simplified adjoint matrix to compare the Cartesian forces and torques estimated at the robot instrument (F_{ri}) vs. the sensor (F_{fs}). The forces are the same and the Cartesian torques (moments) are given by

$$M_{fs} = M_{ri} + P \times F_{ri} \quad (3)$$

where P is the vector offset from the force sensor origin to the robot instrument origin.

V. RESULTS

A. Training Time vs. Training Dataset

First, we analyze the runtime required to train the LSTM vs. the correction network given training data of a certain

length. The training was done on a Titan V (Nvidia, Santa Clara, CA, USA). Since the training can be parallelized, we train the networks for each joint in parallel and report the longest of the run times. We exclude the time to read the data from disk as we assume that the small amount of data can be collected and stored in memory when actually deployed.

For training dataset lengths of up to 4 min, the LSTM can be trained within 4 min and the feed-forward network can be trained within 40 s. For larger training sets, the LSTM exceeds the GPU memory and we reduce the batch size to 32. This increased the training time to up to 14 min for 18 min of data, indicating that the training data collection time is the main bottleneck for adapting to patient setups. With more GPUs or larger GPU memory, the training could be further parallelized and the training time could become negligible. Since training the feed-forward network requires less memory use, the correction network can be trained within 2 min for up to 18 min of training data. To demonstrate that training the correction network is feasible even without a high-end GPU, we also trained it on an Intel Xeon Processor E5-1630 v4 (Intel, Santa Clara, CA, USA). Since the memory was not limited on the CPU setup, the feed-forward network and the LSTM had similar training times. All networks finished training within 60 s for up to 18 min of training data, indicating that memory was the main bottleneck for the small networks used in this work.

B. No-contact Case

One of the key problems with not accounting for trocar forces is that they contribute falsely to forces in no-contact motion, so we consider this case first. Table II shows the results from no contact experiments performed in the trocar. The **Troc** and **Corr** networks are trained with 10 min of data. In this condition, we see that the **Troc** case achieves the best performance. This could be because the LSTM learns smoother behavior than the feed-forward network and is less sensitive to the friction forces as the robot moves around with no contact. The correction network does improve upon the free space estimation, mainly in the F_x and F_y , which are most affected by the body wall (trocar). The correction network improves error in the insertion axis more in the **Seal** case than in the **Base** case. This is interesting as the cannula seal should have the most effect on that axis.

C. Force Estimation

Lastly, we test contact force estimation inside the trocar by interacting with the force sensor and comparing its measurements to our network predictions. We collect several long interactions and separate them into fifteen 15 s segments. We calculate the RMS error for each trial and report the mean and standard deviation of the error over all trials. Table III shows the results for the **Base** and **Seal** cases, which do not consider surgery-specific training. Table IV shows the results for the cases that do consider trocar data for different lengths of the training dataset.

TABLE II

RMSE MEAN AND (STANDARD DEVIATION) OF CARTESIAN FORCE (f , IN N) AND TORQUE (τ , IN NM) IN FREE-MOTION OVER 40 S.

Method	Troc	Base	Seal	Base + Corr	Seal + Corr
F_x	0.480 (0.322)	1.629 (0.924)	1.632 (0.926)	0.884 (0.593)	0.740 (0.503)
F_y	0.463 (0.313)	1.383 (0.870)	1.417 (0.870)	0.964 (0.632)	1.128 (0.765)
F_z	1.367 (0.815)	2.887 (1.327)	2.886 (1.330)	2.327 (1.017)	1.767 (1.201)
τ_x	0.036 (0.023)	0.125 (0.079)	0.125 (0.079)	0.086 (0.058)	0.109 (0.079)
τ_y	0.046 (0.033)	0.145 (0.079)	0.143 (0.080)	0.076 (0.051)	0.063 (0.042)
τ_z	0.016 (0.012)	0.053 (0.034)	0.057 (0.037)	0.030 (0.022)	0.032 (0.025)

TABLE III

RMSE MEAN AND (STANDARD DEVIATION) OF CARTESIAN FORCE (F , IN N) OR TORQUE (τ , IN NM) FOR THE TWO FREE SPACE NETWORKS WITHOUT CORRECTION WHEN INTERACTING WITH THE PHANTOM THROUGH THE TROCAR (15 TRIALS FOR EACH NETWORK).

	Base	Seal
F_x	2.363 (1.020)	2.363 (1.026)
F_y	1.485 (0.476)	1.479 (0.476)
F_z	3.295 (0.674)	3.300 (0.674)
τ_x	0.113 (0.011)	0.114 (0.010)
τ_y	0.146 (0.072)	0.152 (0.069)
τ_z	0.078 (0.018)	0.080 (0.020)

It can be seen that the correction networks have lower mean than the other approaches for the shorter training data sets. With more data, the LSTM trained directly on patient data (**Troc**) is able to get more accurate torque estimations compared to the smaller feed-forward networks. Fig. 7 shows a sample of the network predictions compared to the sensor readings. The training dataset length for the plots in Fig. 7 is 10 min.

Lastly, we evaluate the RMSE over the length of the training dataset used to train each method. Fig. 8 shows the accuracy in force estimation to illustrate how the RMSE changes as a function of dataset length.

VI. DISCUSSION

In our experiments, we observed that the correction network performed better when the original (free space) network was trained with a setup that mimicked the actual use scenario as closely as possible (i.e., the **Seal** case), even though both the **Seal** and **Base** networks performed similarly without correction. We also observe that the improvement in Cartesian torques is limited. This may reflect a limited effect of the trocar on the wrist motions. The present experiments were performed with a single da Vinci instrument but future work aims to extend to more instrument types. We may see more differences in Cartesian torques between instruments. Since the free space network accounts for the robot dynamics, while the correction network should be mainly driven by the instrument shaft's interaction with the trocar and cannula

TABLE IV

RMSE MEAN AND (STANDARD DEVIATION) OF CARTESIAN FORCE (F , IN N) OR TORQUE (τ , IN NM) FOR DIFFERENT LENGTHS OF TROCAR TRAINING DATA WHEN INTERACTING WITH THE PHANTOM THROUGH THE TROCAR (15 TRIALS FOR EACH NETWORK).

	Time	Troc	Base + Corr	Seal + Corr
F_x	4 min	1.366 (1.284)	1.232 (1.264)	1.166 (1.236)
	8 min	1.277 (1.323)	1.231 (1.307)	1.148 (1.284)
	12 min	1.196 (1.304)	1.223 (1.306)	1.134 (1.286)
	16 min	1.172 (1.294)	1.222 (1.302)	1.138 (1.280)
F_y	4 min	1.189 (0.598)	1.025 (0.670)	1.235 (0.630)
	8 min	1.007 (0.459)	0.924 (0.387)	0.977 (0.368)
	12 min	1.033 (0.391)	0.947 (0.414)	0.994 (0.402)
	16 min	0.907 (0.363)	0.940 (0.461)	1.024 (0.444)
F_z	4 min	3.311 (0.782)	2.924 (0.716)	2.510 (1.006)
	8 min	3.317 (0.844)	2.903 (0.741)	2.468 (1.037)
	12 min	2.454 (1.109)	2.875 (0.758)	2.473 (1.053)
	16 min	2.382 (1.126)	2.878 (0.752)	2.472 (1.058)
τ_x	4 min	0.091 (0.023)	0.088 (0.026)	0.095 (0.024)
	8 min	0.083 (0.020)	0.081 (0.023)	0.084 (0.021)
	12 min	0.082 (0.023)	0.081 (0.023)	0.084 (0.021)
	16 min	0.077 (0.022)	0.082 (0.022)	0.083 (0.019)
τ_y	4 min	0.109 (0.086)	0.106 (0.087)	0.102 (0.087)
	8 min	0.106 (0.088)	0.104 (0.089)	0.101 (0.089)
	12 min	0.103 (0.090)	0.104 (0.091)	0.101 (0.091)
	16 min	0.104 (0.091)	0.103 (0.089)	0.100 (0.089)
τ_z	4 min	0.059 (0.025)	0.058 (0.023)	0.061 (0.025)
	8 min	0.056 (0.025)	0.054 (0.023)	0.057 (0.026)
	12 min	0.056 (0.027)	0.054 (0.024)	0.057 (0.026)
	16 min	0.055 (0.027)	0.054 (0.023)	0.057 (0.026)

seal, it is possible that the same correction network may work across different types of instruments. Because the system can detect which instrument is installed, it is feasible to have a different free space network trained for each instrument type. There could even be different correction networks targeting different sources of error. Different instruments of the same type could behave differently. While we have shown a correction network for the trocar, we can have a separate correction scheme to adapt to the instrument.

We observe from Fig. 8 that the error in **Troc** is higher than **Seal+Corr** with less than 12 min of training data, while the **Seal+Corr** has a low error throughout. This suggests that using a free space network with a correction step could reduce the amount of training data needed, making surgery-specific force estimation more clinically feasible. It is interesting to note that while the error of **Troc** reduces with more training data, as expected, the errors of **Seal+Corr** and **Base+Corr** do not show clear trends. This could be due to the estimates being noisier, as seen in the **Seal+Corr** case in Fig. 7. This noise may be due to the lack of memory between predictions since each torque estimate in the feed-forward network is made independently of previous estimates.

Unlike our phantom, patient body musculature and fat can show greater resistance and cause larger interaction forces at the trocar. Future work can add foam or skin-like materials around the trocar to collect more realistic and more varied data. Furthermore, the patient abdominal cavities are insufflated at the beginning of the operation and then the insufflation level changes during the procedure. These

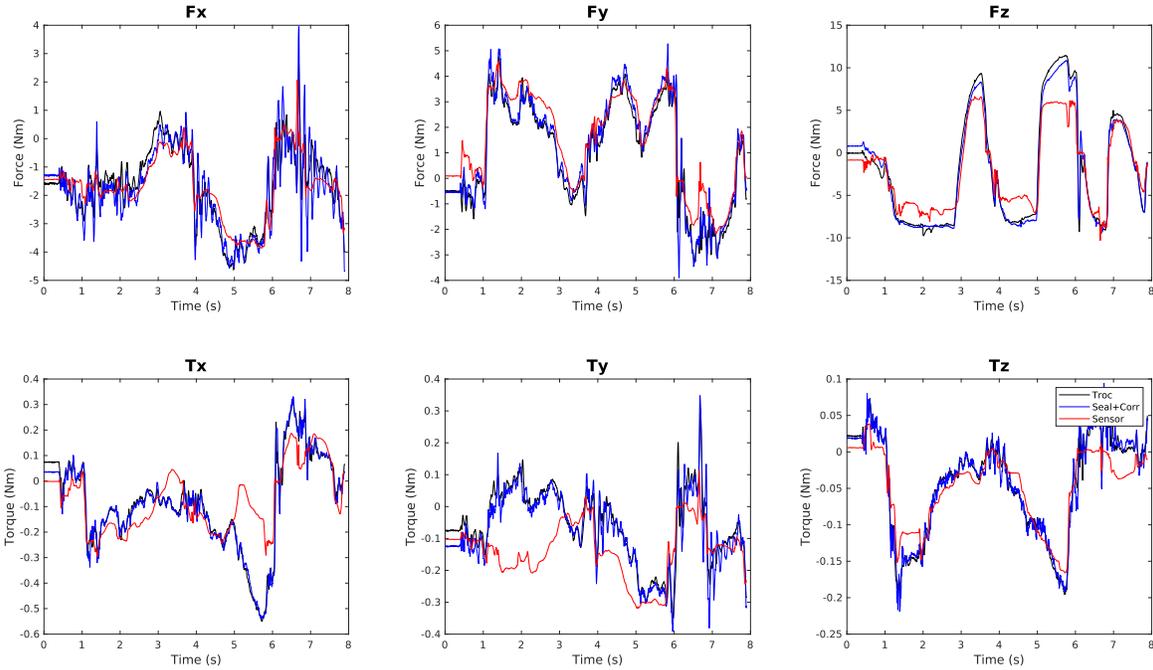


Fig. 7. Cartesian forces and torques for directly-trained trocar network (**Troc**), the corrected torque when the free space network was trained with a cannula seal (**Seal+Corr**), and the force sensor measurements.

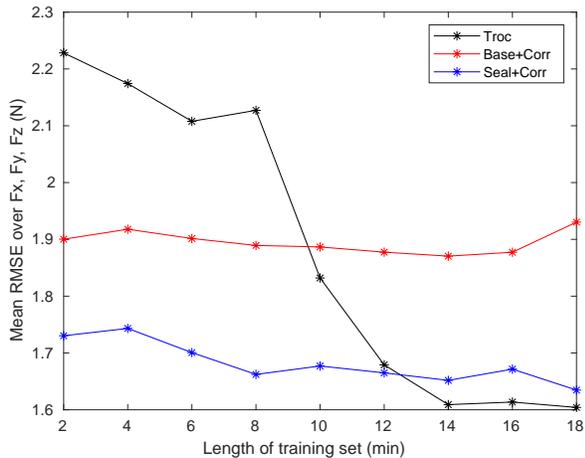


Fig. 8. Training set length vs. mean error (RMSE) of estimated forces over F_x , F_y , and F_z .

factors can cause changes in the interaction forces during the operation. Since the proposed method is self-supervised, it has the potential for lifelong learning to adjust for changes in insufflation levels. If the measured torques differ slightly from predicted, we can assume the motion is in free space and use it to update the network. Otherwise, if the measured torques differ greatly from the predicted, the change is more likely caused by contact.

Another limitation of this work is that due to equipment limitation, we used the same cannula seal for training and testing. This gives an advantage to the **Seal** setup as the third joint is generally responsible for much of the applied force and is not affected by the trocar. However, in the absence

of the cannula seal, our correction approach was still able to improve the performance. This indicates that it could correct for changes in the cannula seal and account for more significant body wall forces in a patient as well.

VII. CONCLUSION

Previous work demonstrated that neural networks can predict internal torques of the da Vinci PSM, thereby enabling estimation of tool-to-tissue interaction forces, at least in a laboratory setting. But, this does not necessarily translate to clinical conditions, which are subject to additional external forces, due to the cannula seal and trocar, that can vary significantly based on the patient and clinical setup. It is not practical, however, to collect a large amount of surgery-specific training data or to wait for a large network to complete training prior to starting the surgical procedure. Therefore, in this work, we propose a self-supervised, two-step training method to account for surgery-specific interactions. The proposed scheme uses an LSTM pre-trained on large amounts of free space data in conjunction with a small feed-forward correction network that can be quickly trained and deployed in the operating room. With a few minutes of surgery-specific data or less, and 1 min training time, we estimate forces inside a phantom that are similar to those estimated in free space [22]. This suggests that the proposed method may be feasible to deploy in clinical use.

VIII. ACKNOWLEDGEMENTS

This work was supported in part by NSF OISE 1927354. The Titan V GPU was donated by the Nvidia Corporation.

REFERENCES

- [1] A. M. Okamura, "Methods for haptic feedback in teleoperated robot-assisted surgery." *The Industrial Robot*, vol. 31, no. 6, pp. 499–508, Dec 2004.
- [2] A. Madhani, G. Niemeyer, and J. Salisbury, "The Black Falcon: a teleoperated surgical instrument for minimally invasive surgery," in *Intl. Conf. on Intelligent Robots and Systems*, vol. 2. IEEE, 1998, pp. 936–944.
- [3] 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 *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1387–1393.
- [4] 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 *Intl. Conf. on Robotics and Auto.* Hong Kong, China: IEEE, 2014, pp. 6434–6439.
- [5] P. J. Berkelman, L. L. Whitcomb, R. H. Taylor, and P. Jensen, "A miniature microsurgical instrument tip force sensor for enhanced force feedback during robot-assisted manipulation," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 917–921, 2003.
- [6] U. Seibold, B. Kubler, and G. Hirzinger, "Prototype of instrument for minimally invasive surgery with 6-axis force sensing capability," in *IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 496–501.
- [7] U. Kim, D.-H. Lee, W. J. Yoon, B. Hannaford, and H. R. Choi, "Force sensor integrated surgical forceps for minimally invasive robotic surgery," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1214–1224, 2015.
- [8] R. Pena, M. J. Smith, N. P. Ontiveros, F. L. Hammond, and R. J. Wood, "Printing strain gauges on Intuitive Surgical da Vinci robot end effectors," in *Intl. Conf. on Intelligent Robots and Systems*. IEEE, Oct 2018, pp. 806–812.
- [9] J. Peirs, J. Clijnen, D. Reynaerts, H. Van Brussel, P. Herijgers, B. Corteville, and S. Boone, "A micro optical force sensor for force feedback during minimally invasive robotic surgery," *Sensors and Actuators A: Physical*, vol. 115, no. 2-3, pp. 447–455, 2004.
- [10] K. S. Shahzada, A. Yurkewich, R. Xu, and R. V. Patel, "Sensorization of a surgical robotic instrument for force sensing," in *Optical Fibers and Sensors for Medical Diagnostics and Treatment Applications XVI*, vol. 9702. International Society for Optics and Photonics, 2016, p. 97020U.
- [11] B. Willaert, N. Famaey, P. Verbrugge, D. Reynaerts, and H. Van Brussel, "Design and in vivo validation of a force-measuring manipulator for MIS providing synchronized video, motion and force data," in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4857–4862.
- [12] S. Shimachi, Y. Hakozaiki, T. Tada, and Y. Fujiwara, "Measurement of force acting on surgical instrument for force-feedback to master robot console," in *International Congress Series*, vol. 1256. Elsevier, 2003, pp. 538–546.
- [13] W. Schwalb, B. Shirinzadeh, and J. Smith, "A force-sensing surgical tool with a proximally located force/torque sensor," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 13, no. 1, p. e1737, 2017.
- [14] G. A. Fontanelli, L. R. Buonocore, F. Ficuciello, L. Villani, and B. Siciliano, "A novel force sensing integrated into the trocar for minimally invasive robotic surgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 131–136.
- [15] M. Mahvash, J. Gwilliam, R. Agarwal, B. Vagvolgyi, L.-M. Su, D. D. Yuh, and A. M. Okamura, "Force-feedback surgical teleoperator: Controller design and palpation experiments," in *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE, 2008, pp. 465–471.
- [16] M. Haghhighpanah, M. Miyasaka, and B. Hannaford, "Utilizing elasticity of cable-driven surgical robot to estimate cable tension and external force," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1593–1600, 2017.
- [17] H. Sang, J. Yun, R. Monfaredi, E. Wilson, H. Fooladi, and K. 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, p. e1824, 2017.
- [18] F. Piqué, M. N. Boushaki, M. Brancadoro, E. De Momi, and A. Men-ciassi, "Dynamic modeling of the da Vinci research kit arm for the estimation of interaction wrench," in *International Symposium on Medical Robotics (ISMR)*. IEEE, 2019, pp. 1–7.
- [19] J. J. O'Neill, T. K. Stephens, and T. M. Kowalewski, "Evaluation of torque measurement surrogates as applied to grip torque and jaw angle estimation of robotic surgical tools," *Robotics and Automation Letters*, vol. 3, no. 4, p. 3027, 2018.
- [20] Y. Guo, B. Pan, Y. Fu, and M. Q.-H. Meng, "Grip force perception based on dAENN for minimally invasive surgery robot," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 1216–1221.
- [21] L. Yu, X. Yu, and Y. Zhang, "Microinstrument contact force sensing based on cable tension using BLSTM-MLP network," *Intelligent Service Robotics*, vol. 13, no. 1, pp. 123–135, 2020.
- [22] N. Tran, J. Y. Wu, A. Deguet, and P. Kazanzides, "A deep learning approach to intrinsic force sensing on the da Vinci surgical robot," in *IEEE International Conference on Robotic Computing*, 2020, pp. 25–32.
- [23] M. Hwang, B. Thananjeyan, S. Paradis, D. Seita, J. Ichnowski, D. Fer, T. Low, and K. Goldberg, "Efficiently calibrating cable-driven surgical robots with RGBD sensing, temporal windowing, and linear and recurrent neural network compensation," *arXiv preprint arXiv:2003.08520*, 2020.
- [24] J. Y. Wu, Z. Chen, A. Deguet, and P. Kazanzides, "FPGA-based velocity estimation for control of robots with low-resolution encoders," in *Intl. Conf. on Intelligent Robots and Systems*. IEEE, Oct 2018, pp. 6384–6389.
- [25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>