Robotic Tool Tracking Under Partially Visible Kinematic Chain: A Unified Approach

Florian Richter, *Student Member, IEEE*, Jingpei Lu, Ryan K. Orosco, *Member, IEEE*, and Michael C. Yip, *Senior Member, IEEE*

***Abstract*—Anytime a robot manipulator is controlled via visual feedback, the transformation between the robot and camera frame must be known. However, in the case where cameras can only capture a portion of the robot manipulator in order to better perceive the environment being interacted with, there is greater sensitivity to errors in calibration of the base-to-camera transform. A secondary source of uncertainty during robotic control are inaccuracies in joint angle measurements which can be caused by biases in positioning and complex transmission effects such as backlash and cable stretch. In this work, we bring together these two sets of unknown parameters into a unified problem formulation when the kinematic chain is partially visible in the camera view. We prove that these parameters are nonidentifiable implying that explicit estimation of them is infeasible. To overcome this, we derive a smaller set of parameters we call lumped error since it lumps together the errors of calibration and joint angle measurements. A particle filter method is presented and tested in simulation and on two real world robots to estimate the lumped error and show the efficiency of this parameter reduction.**

***Index Terms*—Computer vision for automation, computer vision for medical robotics, perception for grasping and manipulation, visual tracking.**

I. Introduction

ANYTIME a robot manipulator is being controlled via visual feedback, an important coordinate transform must be known: the orientation and translation between the robot and the camera frame. This transforms positions and velocities of the robot defined by its kinematics, such as its end-effector position, into the camera's frame of reference where feedback and trajectories are often defined. Typically this relative transform is calibrated for by placing markers, such as ArUco [1], on the robot, identifying them in the image frame, and solving the homogeneous linear system for the *base-to-camera* transform [2]. However, in the case where cameras can only observe a portion of the robot manipulator, there is greater sensitivity to errors in calibration of the base-to-camera transform due to the limited range of motions the robot can take on when collecting data [3]. This situation arises when the camera is positioned to perceive the robotic tool (e.g., gripper) and the environment or objects being manipulated with rather than the entire kinematic chain. An example scenario is shown in Fig. 1. This comes up frequently in object grasping and manipulation tasks [4], [5] and small-scale manipulations such as robotic surgery with the da Vinci Surgical System.

A secondary source of uncertainty that can occur during robotic control is the inaccuracies in joint angle measurements. Errors in joint angle measurements are caused by biases in positioning, drifting in readings, and complex transmission effects such as cable stretch and backlash. Similar to finding the base-to-camera transform, this is typically solved through calibration where a separate sensor, such as a camera, collects

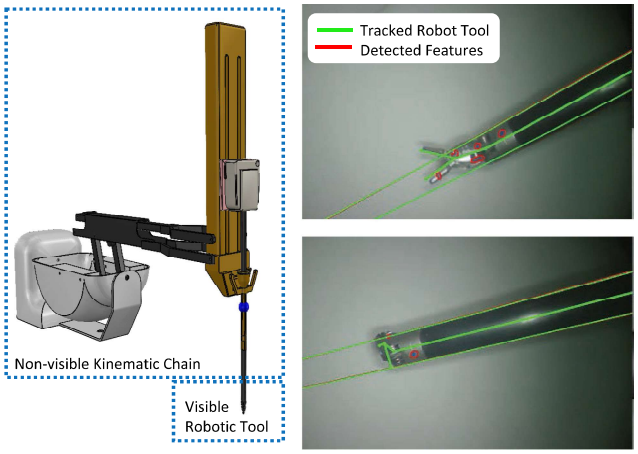

Fig. 1. Precise robotic manipulation utilizes visual information with the sensor positioned to observe the environment and objects of interest rather than the entire kinematic chain. As such, it is challenging to track the robotic tool due to partially visible kinematic chain. In this work we derive and track a smaller set of parameters called lumped error for effective robotic tool tracking in such scenarios. The right two images show the reprojected, tracked robot tool and its corresponding insertion shaft using the proposed lumped error parameter reduction technique.

Manuscript received April 13, 2021; revised July 29, 2021; accepted September 3, 2021. This work was supported in part by University of California San Diego's Galvanizing Engineering in Medicine (GEM) grant, in part by the Intuitive Surgical Technology Grant, in part by the National Science Foundation (NSF) under Grant 1935329 and Grant 2045803, and in part by the US Army Telemedicine and Advanced Technology Research Center (TATRC) under the Robotic Battlefield Medical Support System project. The work of Florian Richter is supported via the NSF Graduate Research Fellowships. This paper was recommended for publication by Associate Editor A. Krupa and Editor F. Chaumette upon evaluation of the reviewers' comments. (*Corresponding author: Florian Richter.)*

Florian Richter, Jingpei Lu, and Michael C. Yip are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093 USA (e-mail: frichter1995@gmail.com; jil360@eng.ucsd.edu; yip@ucsd.edu).

Ryan K. Orosco is with the Department of Surgery—Division of Head and Neck Surgery, University of California San Diego, La Jolla, CA 92093 USA (e-mail: ryanorosco@gmail.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TRO.2021.3111441.

Digital Object Identifier 10.1109/TRO.2021.3111441

ground truth measurements and compares against the joint angle readings [6]. For nonconstant errors, such as cable stretch, explicit dynamic modeling has been conducted [7] and data-driven approaches with neural networks [8]. These methods, however, are challenging to apply outside of a lab setting due to the need for additional sensors or calibration steps. Furthermore, the calibration parameters can degrade over time through irreversible effects from transmission wear-and-tear and mechanical creep.

A strong motivational example for scenarios with challenging base-to-camera calibration and errors in joint angle measurements are surgical robotic-endoscopic platforms [9],[10] such as Titan Medical's SPORT surgical system. The endoscopes are designed to only capture a small working space for higher operational precision. Surgical robotic platforms also typically use cable-drives to enable low-profile robotic tools hence resulting in joint angle measurement error. Furthermore, the bases of the surgical manipulators are adjusted regularly depending on the type of procedure and to fit each patients anatomy. There is a significant amount of previous literature from the surgical robotics community tackling these two problems separately, and we unify these two problems and present a solution which also generalizes to other robot manipulators.

A. Contributions

In this work, we demonstrate the ability to track robotic tools from visual observations that only show part of the kinematic chain under the conditions of uncertainty in base-to-camera transform and joint angle measurements. To this end, we present the following novel contributions.

1) A novel problem formulation which proves direct estimation of all the described parameters is infeasible since they are nonidentifiable.
2) The first approach to unify these uncertainties into a smaller set of parameters which are identifiable and compensates for all the uncertainties.
3) An extension of tracking under simultaneously moving robotic tools and cameras.

We coin the reduced set of parameters as *lumped error*. To track the lumped error, a tracking algorithm based on a particle filter is presented. The particle filter uses visual features from the tracked robotic tool to continuously update the belief of the lumped error. The visual features used in our implementation are detected using markers, edge detectors for geometric primitives such as cylinders, and learned point features. For experimentation, the presented particle filter was evaluated both in simulation and on real world robotic data using the da Vinci Research Kit (dVRK) [12], a widely used surgical robotics research platform with a total of 10 degrees-of-freedom (DoF) and a gripper across both the endoscopic (i.e., surgical robotic camera arm) and robotic manipulator kinematic chains, and the 7-DoF arm on Rethink Robotic's Baxter robot. From this set of experiments, the joint angle disturbances include simulated noise, cable stretch from the dVRK, and backlash from the Baxter arm. Lastly, we summarized our previous work and their results which tracked the lumped error for applications in control of surgical robotics, such as autonomous suction and suture needle manipulation, highlighting the impact this method already has in the surgical robotic community. This summary is written in Appendix A. Overall, these results show that estimation of the lumped error is efficient and yields precise and accurate robotic tool tracking.

B. Related Work

Integration of visual observations with robotic manipulators and handling inaccurate joint angle measurements is not a new concept. Therefore, this related work section is split into different categories to cover the wide-range of solutions presented by previous groups for robotic tracking. A special focus is given to surgical robotics as the challenge of surgical robotic tool tracking would be a direct application of the presented work.

1) Base-to-Camera Estimation: A common approach to calibrating the base-to-camera transform is rigidly attaching a marker whose pose can be directly estimated from visual data (e.g., ArUco [1], ARTag [13], AprilTag [14], and STag [15]), collect multiple images of the marker, and solve the homogeneous linear system [2], [16]. To relieve the heavy reliance on 3-D pose reconstruction, which can often be inaccurate from 2-D images, markers have been attached to robot manipulators to collect 2-D keypoints and the camera-to-base transform is estimated with Solve-PnP [17]. Deep learning approaches have been applied to detect 2-D keypoints on robotic manipulators to remove the need for markers [18]–[21]. However, these calibration methods do not consider errors in joint angle measurements and instead make the assumption that the robot kinematic chain is located exactly at the joint angle readings.

Zhong *et al.* [22] proposed an interactive method to maximize the accuracy in calibration for remote center of motion (RCM) robots which are typical for laparoscopic robots. Similarly, Zhao *et al.* [23] defined a kinematic remote-center coordinate system (KCS) which absorbs all the error in the transform from the camera frame to the base of an RCM robot. Tracking the KCS is a common technique in surgical robotics where the updates come from markers [24], learned features [25]–[27], silhouette matching [28], or online template matching [29]. All of these estimation methods do not explicitly consider the effects of joint angle errors. In fact, we show that the Lumped Error is mathematically equivalent to tracking the KCS implying that these methods are compensating for both the base-to-camera transform and joint angle errors. Furthermore, in this work we generalize Lumped Error to other robotic manipulators. In Appendix A, we summarize our previous work in control of surgical robotics which utilized tracking KCS, or equivalently lumped error, to highlight the impact this method already has on the surgical robotics community.

2) Joint Measurement Error Estimation: Using fiducial markers to collect data, Pastor *et al.* [30] applied a data-driven approach to estimating the joint angle error. Meanwhile Wang *et al.* [31] used markers to estimate the joint angle offsets in real-time via inverse kinematics. From the perspective of surgical robotics, errors of joint angle readings due to transmissions effects has largely been studied in the context of cable drives. Miyasaka *et al.* [7] explicitly modeled the physical effects of cable transmissions such as friction and hysteresis. Learning-based

approaches have been applied in the form of neural networks for direct estimation of cable stretch [8], [32] and Gaussian processes for compensation [33]. From visual data, unscented Kalman filter [34] and neural networks [35] were applied to estimate the effects of cable stretch. These techniques however are impractical to apply outside of a lab setting due to the need for additional sensors or calibration steps. In addition, calibration parameters can degrade over time due to mechanical effects such as cable creep which is when cable stretch varies irreversibly through usage.

3) Combined Base-to-Camera and Joint Estimation: Joint calibration techniques have been proposed which optimize for both joint angle offsets and base-to-camera transformations [6], [36]. To handle dynamic uncertainties, such as nonconstant joint angle errors, real-time estimation by combining iterative closest point from depth sensing and Kalman Filtering have been proposed [37]. A probabilistic approach has also been proposed where the observation models are grounded in physical parameters hence making it easier to tune the hyper parameters [38]. These works largely focus on integration of sensors into real-time estimation. Instead of this, we look into parameter reductions for the case of partially visible kinematic chains. Therefore, the proposed lumped error parameter reduction can aid these efforts by reducing the total number of parameters that need to be estimated in the case of a partially visible kinematic chain. Nonetheless, we propose a particle filter to estimate the lumped error which relies only on image data; meanwhile, the efforts above rely on depth sensing which is not as readily available on all robotic platforms such as the da Vinci Surgical System.

A separate and popular approach to controlling a robot from visual feedback without the base-to-camera transform is through online jacobian estimation [39]. These visual servoing techniques can even compensate for kinematic inaccuracies and joint angle measurement errors [40]. While these techniques are sufficient to control the end-effector in the camera frame, they do not describe the rest of the kinematic chain in the camera frame.

4) Eye-in-Hand Configuration: Another consideration in the case of a robotic camera arm is the problem of *eye-in-hand* calibration [41]. This particular visual-robotic challenge is not considered here but included for the sake of completeness. Zhang *et al.* [42] developed a computationally efficient methods using dual quaternions. Adjoint transformations from twist motions have also been applied to converge to solutions with high accuracy [43], [44]. In the case of RCM robots, reduction of the computational complexity has been found [45], [46]. Similar to the previously described visual servoing techniques, the robot camera arm can also be controlled through online jacobian estimation [47], [48].

II. Methods

The problem formulation for base-to-camera and joint angle measurement errors with camera interaction is first explained in this section. To overcome the described challenges for a partially observed robot from visual observations, a lumped error is derived and then extended to the eye-in-hand case. Finally, our proposed method for tracking the lumped error with a particle filter approach is described.

A. Problem Formulation

The 3-D geometry of a robotic tool can be fully described in the stationary camera frame through a base-to-camera transform and forward kinematics. A single point, $\mathbf{o}^j \in \mathbb{R}^3$, on the jth link of a robotic tool can be transformed to the stationary camera frame by

$$\overline{\mathbf{o}}_t^c = \mathbf{T}_b^c \prod_{i=1}^{j} \mathbf{T}_i^{i-1}(q_t^i)\overline{\mathbf{o}}^j \tag{1}$$

at time t where $\mathbf{T}_b^c \in SE(3)$ is the base-to-camera transform and $\mathbf{T}_i^{i-1}(q_t^i) \in SE(3)$ is the ith joint transform with joint angle q_t^i. The overline operator $\bar{(\cdot)}$ defines the homogeneous representation of a 3-D point (e.g $\overline{\mathbf{o}} = \begin{bmatrix}\mathbf{o} & 1\end{bmatrix}^\top$). Therefore, all that is necessary to describe a robotic manipulator in the camera frame is the base-to-camera transform and joint angles. Typically, the base-to-camera transform can be calibrated for, and the joint angles can be found from encoder readings. The issue with applying this approach directly to scenarios where the camera only captures images with a portion of the kinematic chain is that small errors in calibration or joint angles will be exacerbated in the image frame.

Therefore, let $\tilde{q}_t^1, \ldots, \tilde{q}_t^{n_j}$ be the joint angle measurements and $e_t^1, \ldots, e_t^{n_j}$ are the measurement errors, such that

$$q_t^i = \tilde{q}_t^i + e_t^i \tag{2}$$

for all $i = 1, \ldots, n_j$. No distribution is assumed for the errors, e_t^i. For example, the error could be constant bias from absolute position error or nonconstant with hysteresis effects from cable stretch. Combining with (1), the robotic tool can be described in the camera frame by

$$\overline{\mathbf{o}}_t^c = \mathbf{T}_{b-}^c \mathbf{T}_b^{b-} \prod_{i=1}^{j} \mathbf{T}_i^{i-1}(\tilde{q}_t^i + e_t^i)\overline{\mathbf{o}}^j \tag{3}$$

where the true base-to-camera transform is broken into $\mathbf{T}_{b-}^c \in SE(3)$ and $\mathbf{T}_b^{b-} \in SE(3)$ which are measured from an initial calibration and the error in the calibration respectively. Therefore, in order to correctly describe the robotic tool in the camera frame, both the joint angle errors, e_t^i, and the error in base-to-camera transform, $\mathbf{T}_b^{b-}$, need to be estimated. Let n_j be the total number of joint angles and the $SE(3)$ error transform, $\mathbf{T}_b^{b-}$, be estimated with an axis-angle and a translation vector, resulting a total of $n_j + 6$ parameters to estimate.

Explicit estimation for the joint angles and base-to-camera transform is not possible when only a portion of the kinematic chain is visible in the camera frame. This is since one cannot distinguish where the source of error is coming from, joint angles or the base-to-camera transform calibration. For example, a surgical tool is considered partially visible with regards to the endoscopic camera. The endoscopes narrow field only has visual information of the tool-tip and not the base nor joints preceding the articulated wrist resulting in multiple viable solutions to

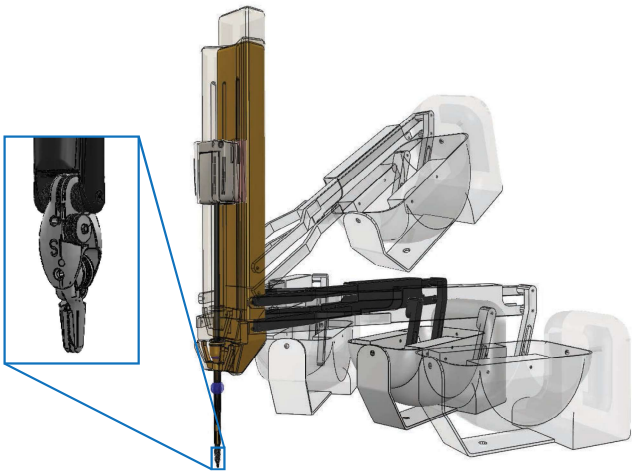

Fig. 2. Given an image of the robot tool, as shown in blue, and not the whole kinematic chain, multiple solutions exist for joint angles and base-to-camera transform errors. Examples of these solutions are shown with the transparent kinematic chains. This implies that it is infeasible to estimate these unknowns directly when the kinematic chain is partially visible.

estimating $\mathbf{T}_b^{b-}$ and e_t^i. An example of this is shown in Fig. 2 where the joints part of the RCM are not visible to the endoscopic camera, but the joints on the gripper are.

This relates to the concept of *identifiability* [49]. Identifiability is concerned with the existence of a unique inverse association with regard to the parameters estimated from observations. Fig. 2 shows examples of there not being a unique association from the image of the surgical tool from an endoscope to the base-to-camera transform and errors in joint angles. These instances are denoted as *observationally equivalent*. Parameters are only considered identifiable if there are no observational equivalences.

Claim 1: When only using the camera data for observations, then the error in base-to-camera transform, $\mathbf{T}_b^{b-}$, and errors in the first n_b joint angles, $e_t^1, \ldots, e_t^{n_b}$, are not identifiable if all the kinematic links preceding joint n_b are out of the camera frame.

Proof. Let *modified Denavit–Hartenberg parameters* be used to define each forward kinematic joint transform. Therefore, $\mathbf{T}_i^{i-1}(q_t^i) = \mathbf{T}_x(\alpha^i, a^i)\mathbf{T}_z(\theta^i, d^i)$ where

$$\mathbf{T}_x(\alpha^i, a^i) = \begin{bmatrix} 1 & 0 & 0 & a^i \\ 0 & cos(\alpha^i) & -sin(\alpha^i) & 0 \\ 0 & sin(\alpha^i) & cos(\alpha^i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_z(\theta^i, d^i) = \begin{bmatrix} cos(\theta^i) & -sin(\theta^i) & 0 & 0 \\ sin(\theta^i) & cos(\theta^i) & 0 & 0 \\ 0 & 0 & 1 & d^i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and q_t^i is plugged into θ^i or d^i for a revolute and prismatic joint, respectively. A revolute joint transform with a joint angle of $\omega + \psi \in \mathbb{R}$, $\mathbf{T}_i^{i-1}(\omega + \psi)$, can be expanded using the modified Denavit–Hartenberg parameters

$$\mathbf{T}_x(\alpha^i, a^i)\mathbf{T}_z(\omega + \psi, d^i) \tag{4}$$

$$\mathbf{T}_x(\alpha^i, a^i)\mathbf{T}_z(\omega, 0)\mathbf{T}_x^{-1}(\alpha^i, a^i)\mathbf{T}_x(\alpha^i, a^i)\mathbf{T}_z(\psi, d^i) \tag{5}$$

$$\mathbf{T}_i(\omega)\mathbf{T}_i^{i-1}(\psi) \tag{6}$$

where $\mathbf{T}_i(\omega) = \mathbf{T}_x(\alpha^i, a^i)\mathbf{T}_z(\omega, 0)\mathbf{T}_x^{-1}(\alpha^i, a^i)$. Likewise for a prismatic joint transform

$$\mathbf{T}_x(\alpha^i, a^i)\mathbf{T}_z(\theta^i, \omega + \psi) \tag{7}$$

$$\mathbf{T}_x(\alpha^i, a^i)\mathbf{T}_z(0, \omega)\mathbf{T}_x^{-1}(\alpha^i, a^i)\mathbf{T}_x(\alpha^i, a^i)\mathbf{T}_z(\theta^i, \psi) \tag{8}$$

$$\mathbf{T}_i(\omega)\mathbf{T}_i^{i-1}(\psi) \tag{9}$$

where $\mathbf{T}_i(\omega) = \mathbf{T}_x(\alpha^i, a^i)\mathbf{T}_z(0, \omega)\mathbf{T}_x^{-1}(\alpha^i, a^i)$. Note that the same notation, $\mathbf{T}_i(\omega)$, is used for rotational and prismatic joints for simplified notation in the coming equations.

Using these expansions, we will show using induction that portions of the joint angle errors can be expanded out as follows:

$$\prod_{i=1}^{n_b} \mathbf{T}_i^{i-1}(\tilde{q}_t^i + e_t^i) = \mathbf{T}^{n_b} \prod_{i=1}^{n_b} \mathbf{T}_i^{i-1}(\tilde{q}_t^i + \beta_i e_t^i) \tag{10}$$

where

$$\mathbf{T}^{n_b} = \prod_{k=1}^{n_b} \left(\prod_{i=1}^{k-1} \mathbf{T}_i^{i-1}(\tilde{q}_t^i + \beta_i e_t^i) \right) \mathbf{T}_k((1-\beta_i)e_t^k) \left(\prod_{i=1}^{k-1} \mathbf{T}_i^{i-1}(\tilde{q}_t^i + \beta_i e_t^i) \right)^{-1} \tag{11}$$

and $\beta_i \in \mathbb{R}$ for $i = 1, \ldots, n_b$ is an arbitrary portion of the joint angle error not to be lumped into $\mathbf{T}^{n_b}$. For the base case of $n_b = 1$ in (10), the error of the joint angle can be pulled out

$$\mathbf{T}_1^0(\tilde{q}_t^i + e_t^i) = \mathbf{T}_1((1-\beta_1)e_t^i)\mathbf{T}_1^0(\tilde{q}_t^i + \beta_1 e_t^i) \tag{12}$$

using (4)–(9).

Assuming (10) holds true for $n_b = m$, then for $n_b = m + 1$ the left-hand-expression from (10) can be rewritten as

$$\mathbf{T}^m \prod_{i=1}^{m} \mathbf{T}_i^{i-1}(\tilde{q}_t^i + \beta_i e_t^i)\mathbf{T}_{m+1}^m(\tilde{q}_t^{m+1} + e_t^{m+1}) \tag{13}$$

which expands to

$$\mathbf{T}^m \prod_{i=1}^{m} \mathbf{T}_i^{i-1}(\tilde{q}_t^i + \beta_i e_t^i)\mathbf{T}_{m+1}((1-\beta_{m+1})e_t^{m+1}) \mathbf{T}_{m+1}^m(\tilde{q}_t^{m+1} + \beta_{m+1} e_t^{m+1}) \tag{14}$$

using (4)–(9). Expanding the expression one more time

$$\mathbf{T}^m \prod_{i=1}^{m} \mathbf{T}_i^{i-1}(\tilde{q}_t^i + \beta_i e_t^i)\mathbf{T}_{m+1}((1-\beta_{m+1})e_t^{m+1}) \left(\prod_{i=1}^{m} \mathbf{T}_i^{i-1}(\tilde{q}_t^i + \beta_i e_t^i) \right)^{-1} \prod_{i=1}^{m} \mathbf{T}_i^{i-1}(\tilde{q}_t^i + \beta_i e_t^i) \mathbf{T}_{m+1}^m(\tilde{q}_t^{m+1} + \beta_{m+1} e_t^{m+1}) \tag{15}$$

which is equivalent to (10). Therefore (10) holds for $n_b = 1, 2, \ldots$ by mathematical induction.

Let $P(\mathbf{y}_t|\mathbf{T}_b^{b-}, e_t^1, \ldots, e_t^{n_j})$ be a proper probability distribution and the observation model of some feature $\mathbf{y}$ from the surgical tool in the camera frame parameterized by all the unknowns in the kinematic chain described in (3). Since $P(\mathbf{y}_t|\cdot)$ cannot describe a feature from the kinematic links preceding joint n_b, the observation model for some feature $\mathbf{y}$ can be reparameterized to

$$P\left(\mathbf{y}_t|\mathbf{T}_{b-}^{c}\mathbf{T}_{b}^{b-}\prod_{i=1}^{n_b}\mathbf{T}_i^{i-1}(\tilde{q}_t^i + e_t^i), e_t^{n_b+1}, \ldots, e_t^{n_j}\right). \quad (16)$$

The equality in (10) implies that the observation is not a one-to-one mapping from the parameter space $(\mathbf{T}_b^{b-}, e_t^1, \ldots, e_t^{n_j})$ to the camera observation (output of $P(\mathbf{y}_t|\cdot)$). In fact, for each observation generated by $P(\mathbf{y}_t|\cdot)$, there are an infinite solutions for the inverse mapping which are spanned by $\beta_1, \ldots, \beta_{n_b}$. Since there are infinite observational equivalencies, the parameters are not identifiable. ■

The equality in (10), which causes the lack of identifiability, can be interpreted as moving the joint errors from the kinematic chain to the base transform of the robot. Lack of identifiability implies undesirable properties for parameter estimation such as rank deficiency in the Fischer information matrix [49]. Furthermore, it shows the inability to estimate both errors in joint angles and base-to-camera transform.

B. Lumped Error Transform for Estimation

Due to Claim 1, it is infeasible to estimate all of the error parameters described in (3) when only using camera data. Therefore, we propose a parameter reduction technique where all the errors of the first n_b joints are lumped together with the error in base-to-camera transform. Hence, we call it the *lumped error* transform.

Using (10), (3) can be rewritten as

$$\overline{\mathbf{o}}_t^c = \mathbf{T}_{b-}^{c}\mathbf{T}_{n_b}^{b-}(\mathbf{w}_t, \mathbf{b}_t)\prod_{i=1}^{n_b}\mathbf{T}_i^{i-1}(\tilde{q}_t^i)\prod_{i=n_b+1}^{j}\mathbf{T}_i^{i-1}(\tilde{q}_t^i + e_t^i)\overline{\mathbf{o}}^j \quad (17)$$

where $\mathbf{T}_{n_b}^{b-}(\mathbf{w}_t, \mathbf{b}_t) \in SE(3)$ is the lumped error transform of all the first n_b joint errors and the base-to-camera transform calibration error $\mathbf{T}_b^{b-}$ and it is parameterized by an orientation $\mathbf{w}_t \in \mathbb{R}^3$ and translation $\mathbf{b}_t \in \mathbb{R}^3$. The lumped error analytical solution from the joint angle errors and error in base-to-camera transform is $\mathbf{T}_{n_b}^{b-}(\mathbf{w}_t, \mathbf{b}_t) = \mathbf{T}_b^{b-}\mathbf{T}^{n_b}$ where $\mathbf{T}^{n_b}$ is defined in (11) with $\beta_i = 0$ for $i = 1, \ldots, n_b$.

Intuitively, the lumped error transform is virtually adjusting the base of the kinematic chain for the robot in the camera frame. The virtual adjustments are done to fit the error of the first n_b joint angles and any error in base-to-camera transform. The lumped error transform removes the many to one mapping shown in (16). Furthermore, it is a significant reduction of the parameters that need to be estimated for robotic tool tracking. With a total of n_j joints and the $SE(3)$ error transforms, $\mathbf{T}_{b-}^{b}$ and $\mathbf{T}_{n_b}^{b-}(\mathbf{w}_t, \mathbf{b}_t)$, being estimated using axis-angle and a translation vector, then (3) has $n_j + 6$ parameters to estimate while (17) has $n_j - n_b + 6$ parameters.

Even with this parameter reduction, it can still be challenging to constrain all of the parameters with image observations. For example, from a single image frame, 4 pixel point detections are required to constrain the lumped error transform [50] and additional point detections would be needed for the joint errors $e_t^{n_b+1}, \ldots, e_t^{n_j}$. Therefore, we propose the following simplification to (17) if there is not an abundance of features:

$$e_t^i \approx 0 \text{ for } i = n_b + 1, \ldots, n_j. \quad (18)$$

With this simplification, only the lumped error transform needs to be estimated. This simplification can be done in situations where the error from joints $n_b + 1, \ldots, n_j$ does not propagate through the kinematic chain dramatically. In cases where the camera focuses on an articulated wrist or gripper as shown in Fig. 2, this is an acceptable assumption as their link lengths are short hence reducing their sensitivity to error.

The resulting expression when combining the simplification in (18) with (17) is equivalent to what previous literature in robotic surgical tool tracking described as the KCS which was developed for RCM based robots [23]. Therefore, the KCS tracking formulation not only corrects for the error in base-to-camera transform, but also joint angle errors.

The lumped error can also be moved to the right hand-side of the first n_b joint transforms in (17) giving the following expression:

$$\overline{\mathbf{o}}_t^c = \mathbf{T}_{b-}^{c}\prod_{i=1}^{n_b}\mathbf{T}_i^{i-1}(\tilde{q}_t^i)\mathbf{T}_{n_b+1}^{n_b,b-}(\mathbf{w}_t, \mathbf{b}_t)\prod_{i=n_b+1}^{j}\mathbf{T}_i^{i-1}(\tilde{q}_t^i + e_t^i)\overline{\mathbf{o}}^j \quad (19)$$

where the right hand-side lumped error is

$$\mathbf{T}_{n_b+1}^{n_b,b-}(\mathbf{w}_t, \mathbf{b}_t) = \left(\prod_{i=1}^{n_b}\mathbf{T}_i^{i-1}(\tilde{q}_t^i)\right)^{-1}\mathbf{T}_{n_b}^{b-}(\mathbf{w}_t, \mathbf{b}_t)\prod_{i=1}^{n_b}\mathbf{T}_i^{i-1}(\tilde{q}_t^i) \quad (20)$$

which is equivalent to the tracking method proposed by Hao *et al.* [28] and shows their method compensates for both errors in base-to-camera transform and joint angle errors.

C. Extension to Robotic Camera Arm

In the case of eye-in-hand, the constant true base-to-camera transform, $\mathbf{T}_b^c = \mathbf{T}_{b-}^{c}\mathbf{T}_b^{b-}$, described in (3) is replaced with a kinematic chain as follows:

$$\overline{\mathbf{o}}_t^c = \mathbf{T}_{c_n}^{c}\left(\prod_{i=1}^{n}\mathbf{T}_{c_i}^{c_{i-1}}(q_t^{c_i})\right)^{-1}\mathbf{T}_b^{c_b}\prod_{i=1}^{j}\mathbf{T}_i^{i-1}(\tilde{q}_t^i + e_t^i)\overline{\mathbf{o}}^j \quad (21)$$

where $\mathbf{T}_{c_n}^{c} \in SE(3)$ is the static transform from the final joint to the camera frame, $\mathbf{T}_{c_i}^{c_{i-1}}(q_t^{c_i}) \in SE(3)$ is the ith joint transform of the camera arm with joint angle $q_t^{c_i}$, and $\mathbf{T}_b^{c_b} \in SE(3)$ is the base-to-base transform (i.e., transform from the base of the robotic tool to the base of the robotic camera arm).

Calibration of the base-to-base transform is even more challenging than calibrating the base-to-camera transform since

the kinematic chain is extended by the camera arm. Joint angle errors are also still assumed. Let $\tilde{q}_t^{c_i}$ and $e_t^{c_i}$ be the joint angle measurement and measurement error respectively for joint angle c_i on the camera arm. The base to base transform $\mathbf{T}_b^{c_b}$ is split into the calibrated base to base transform $\mathbf{T}_{b-}^{c_b}$ and the error in calibration $\mathbf{T}_b^{b-}$. Therefore, (21) is rewritten as

$$\overline{\mathbf{o}}_t^c = \mathbf{T}_{c_n}^c \left(\prod_{i=1}^{n} \mathbf{T}_{c_i}^{c_{i-1}} (\tilde{q}_t^{c_i} + e_t^{c_i}) \right)^{-1} \mathbf{T}_{b-}^{c_b} \mathbf{T}_b^{b-} \prod_{i=1}^{j} \mathbf{T}_i^{i-1} (\tilde{q}_t^i + e_t^i) \overline{\mathbf{o}}^j. \quad (22)$$

The kinematic links from the camera arm are typically not visible in the camera frame. Therefore, the same nonidentifiability issue from Claim 1 extends to joint angle errors $e_t^{c_i}$ for $i = 1, \dots c_n$. To solve this issue, the lumped error from (17) is applied to the camera arm's kinematic chain. This results in

$$\overline{\mathbf{o}}_t^c = \mathbf{T}_{c_n}^c \left(\prod_{i=1}^{n} \mathbf{T}_{c_i}^{c_{i-1}} (\tilde{q}_t^{c_i}) \right)^{-1} \mathbf{T}_{c_n}^{c_b} (\mathbf{w}_t^c, \mathbf{b}_t^c)^{-1} \mathbf{T}_{b-}^{c_b} \mathbf{T}_{n_b}^{b-} (\mathbf{w}_t, \mathbf{b}_t) \prod_{i=1}^{n_b} \mathbf{T}_i^{i-1} (\tilde{q}_t^i) \prod_{i=n_b+1}^{j} \mathbf{T}_i^{i-1} (\tilde{q}_t^i + e_t^i) \overline{\mathbf{o}}^j \quad (23)$$

where $\mathbf{T}_{c_n}^{c_b} (\mathbf{w}_t^c, \mathbf{b}_t^c)$ analytical expression from joint angles is described in (10) with $\beta_i = 0$ for $i = 1, \dots, n$. Continuing further, (23) can be reduced to a single unknown pose parameterized by orientation and translation vectors $\mathbf{w}_t^l, \mathbf{b}_t^l \in \mathbb{R}^3$, respectively, and unknown joint errors e_t^i for $i = n_b + 1, \dots, n_j$. The new expression is

$$\overline{\mathbf{o}}_t^c = \mathbf{T}_{c_n}^c \left(\prod_{i=1}^{n} \mathbf{T}_{c_i}^{c_{i-1}} (\tilde{q}_t^{c_i}) \right)^{-1} \mathbf{T}_{b-}^{c_b} \mathbf{T}_{n_b}^{c_n} (\mathbf{w}_t^l, \mathbf{b}_t^l) \prod_{i=1}^{n_b} \mathbf{T}_i^{i-1} (\tilde{q}_t^i) \prod_{i=n_b+1}^{j} \mathbf{T}_i^{i-1} (\tilde{q}_t^i + e_t^i) \overline{\mathbf{o}}^j \quad (24)$$

where

$$\mathbf{T}_{n_b}^{c_n} (\mathbf{w}_t^l, \mathbf{b}_t^l) = \left(\mathbf{T}_{b-}^{c_b} \right)^{-1} \mathbf{T}_{c_n}^{c_b} (\mathbf{w}_t^c, \mathbf{b}_t^c)^{-1} \mathbf{T}_{b-}^{c_b} \mathbf{T}_{n_b}^{b-} (\mathbf{w}_t, \mathbf{b}_t). \quad (25)$$

The lumped error that would be estimated in this case, $\mathbf{T}_{n_b}^{c_n} (\mathbf{w}_t^l, \mathbf{b}_t^l)$, holds similar properties to the previous case of the stationary camera lumped error. The base of the robotic manipulator relative to the camera arm base is virtually adjusted to compensate for the error in the first n_b joint readings in it and all the joint readings in the robotic camera arm. This lumped error also reduces the number of parameters that need to be estimated to $n_j - n_b + 6$ while in (22) there are $n_j + n + 6$ unknown parameters. For even fewer parameters to estimate, the simplification in (18) can be applied resulting in only six parameters.

Algorithm 1: Particle Filter to Track Lumped Error.

Input : Initial base-to-camera transform $\mathbf{T}_{b-}^c$
Output: Estimated Lumped Error and Observable Joint Errors $\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t$

Initialize particle list $P_{0|0} = \{\alpha_{0|0}^{(p)}, \hat{\mathbf{w}}_{0|0}^{(p)}, \hat{\mathbf{b}}_{0|0}^{(p)}, \hat{\mathbf{e}}_{0|0}^{(p)}\}_{p=1}^N$
// Initialize particle distribution
for *particle* $p \in P_{0|0}$ **do**
  $\left[\hat{\mathbf{w}}_{0|0}^{(p)}, \hat{\mathbf{b}}_{0|0}^{(p)}\right]^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{w},\mathbf{b},0})$
  $\alpha_{0|0}^{(p)} \leftarrow \mathcal{G}\left(\left[\hat{\mathbf{w}}_{0|0}^{(p)}, \hat{\mathbf{b}}_{0|0}^{(p)}\right]^\top, \mathbf{\Sigma}_{\mathbf{w},\mathbf{b},0}\right)$
  $\hat{\mathbf{e}}_{0|0}^{(p)} \sim \mathcal{U}(-\mathbf{a}_{\hat{\mathbf{e}}}, \mathbf{a}_{\hat{\mathbf{e}}})$
$\{\alpha_{0|0}^{(p)}\}_{p=1}^N \leftarrow normalizeWeights\left(\{\alpha_{0|0}^{(p)}\}_{k=1}^N\right)$
// Main Loop
while *image and joint readings,* $(\mathbf{I}_t,\ \tilde{\mathbf{q}}_t)$*, arrive* **do**
  // Predict
  Initialize new particle list $P_{t|t-1}$
  for *particle* $p \in P_{t|t-1}$ **do**
    $q \sim P_{t-1|t-1}$ weights $\{\alpha_{t-1|t-1}^{(1)}, \dots, \alpha_{t-1|t-1}^{(N)}\}$
    $\left[\hat{\mathbf{w}}_{t|t-1}^{(p)}, \hat{\mathbf{b}}_{t|t-1}^{(p)}\right]^\top \sim \mathcal{N}\left(\left[\hat{\mathbf{w}}_{t-1|t-1}^{(q)}, \hat{\mathbf{b}}_{t-1|t-1}^{(q)}\right]^\top, \mathbf{\Sigma}_{\mathbf{w},\mathbf{b},t}\right)$
    $\alpha_{t|t-1}^{(p)} \leftarrow \mathcal{G}\left(\left[\hat{\mathbf{w}}_{t|t-1}^{(p)}, \hat{\mathbf{b}}_{t|t-1}^{(p)}\right]^\top, \mathbf{\Sigma}_{\mathbf{w},\mathbf{b},t}\right)$
    $\hat{\mathbf{e}}_{t|t-1}^{(p)} \sim \mathcal{N}\left(\hat{\mathbf{e}}_{t-1|t-1}^{(q)}, \mathbf{\Sigma}_{\hat{\mathbf{e}},t}\right)$
    $\alpha_{t|t-1}^{(p)} \leftarrow \alpha_{t|t-1}^{(p)} \cdot \mathcal{G}\left(\hat{\mathbf{e}}_{t|t-1}^{(p)}, \mathbf{\Sigma}_{\hat{\mathbf{e}},t}\right)$
  // Update
  $\mathbf{m}_t \leftarrow detectRobotPointFeatures(\mathbf{I}_t)$
  $\boldsymbol{\rho}_t, \boldsymbol{\phi}_t \leftarrow detectRobotEdgeFeatures(\mathbf{I}_t)$
  for *particle* $p \in P_{t|t-1}$ **do**
    $\hat{\mathbf{m}}_t \leftarrow projPoints(\hat{\mathbf{w}}_{t|t-1}^{(p)}, \hat{\mathbf{b}}_{t|t-1}^{(p)}, \hat{\mathbf{e}}_{t|t-1}^{(p)}, \tilde{\mathbf{q}}_t)$
    $A_m, \mathbf{C}^m \leftarrow associatePoints(\mathbf{m}_t, \hat{\mathbf{m}}_t)$
    $\alpha_{t|t-1}^{(p)} \leftarrow \alpha_{t|t-1}^{(p)} \cdot pointObsModel(A_m, \mathbf{C}^m)$
    $\hat{\boldsymbol{\rho}}_t, \hat{\boldsymbol{\phi}}_t \leftarrow projEdges(\hat{\mathbf{w}}_{t|t-1}^{(p)}, \hat{\mathbf{b}}_{t|t-1}^{(p)}, \hat{\mathbf{e}}_{t|t-1}^{(p)}, \tilde{\mathbf{q}}_t)$
    $A_l, \mathbf{C}^l \leftarrow associateEdges([\boldsymbol{\rho}_t, \boldsymbol{\phi}_t], [\hat{\boldsymbol{\rho}}_t, \hat{\boldsymbol{\phi}}_t])$
    $\alpha_{t|t-1}^{(p)} \leftarrow \alpha_{t|t-1}^{(p)} \cdot edgeObsModel(A_l, \mathbf{C}^l)$
  $P_{t|t} \leftarrow P_{t|t-1}$
  $\{\alpha_{t|t}^{(p)}\}_{k=1}^N \leftarrow normalizeWeights\left(\{\alpha_{t|t}^{(p)}\}_{k=1}^N\right)$
  if $numEffectiveParticles(P_{t|t}) > N_{eff}$ **then**
    $P_{t|t} \leftarrow stratifyResampling(P_{t|t})$
  $\left[\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t\right]^\top = \sum_{p=1}^{N} \alpha_{t|t}^{(p)} \left[\hat{\mathbf{w}}_{t|t}^{(p)}, \hat{\mathbf{b}}_{t|t}^{(p)}, \hat{\mathbf{e}}_{t|t}^{(p)}\right]^\top$

D. Particle Filter for Tracking of Lumped Error

The result in (17) reduced the number of parameters that are required to be estimated to the Lumped Error transform, $\mathbf{T}_{n_b}^{b-} (\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t)$, and joint errors: $\hat{\mathbf{e}}_t := \begin{bmatrix} \hat{e}_t^{n_b+1} & \dots & \hat{e}_t^{n_j} \end{bmatrix}^\top$. By these reductions, one can use previously developed methods of parameter estimation to track it such as the extended Kalman

filter, unscented Kalman filter, or a particle filter with updates from camera images. For our approach, we utilized a particle filter because of its flexibility to model the posterior probability density function with a finite-number of samples [51] rather than using parametric model such as a Kalman filter. It also has found recent success in estimating poses [52] which is needed here for the lumped error transform. The coming sections describe tracking of the parameters by defining the motion models and observation models. The last section covers the few modifications necessary for the eye-in-hand case. An outline of the proposed particle filter is shown in Algorithm 1 and specific parameter values used in the experiments are described in Appendix B.

1) Motion Model: The joint angle errors are initialized from a uniform distribution and have a motion model of additive zero mean Gaussian noise

$$\hat{\mathbf{e}}_0 \sim \mathcal{U}(-\mathbf{a}_{\hat{\mathbf{e}}}, \mathbf{a}_{\hat{\mathbf{e}}}) \qquad \hat{\mathbf{e}}_{t+1} \sim \mathcal{N}(\hat{\mathbf{e}}_t, \mathbf{\Sigma}_{\hat{\mathbf{e}},t+1}) \tag{26}$$

where $\mathbf{a}_{\hat{\mathbf{e}}} \in \mathbb{R}^{n_j - n_b}$ describes the bounds of constant joint angle error and $\mathbf{\Sigma}_{\hat{\mathbf{e}},t+1} \in \mathbb{R}^{(n_j-n_b)\times(n_j-n_b)}$ is a covariance matrix. The initialization is done to capture joint angle biases, and a Weiner process is chosen for the motion model due to its ability to generalize over a large number of random processes.

Let the tracked lumped error, $\mathbf{T}^{b-}_{n_b}(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t)$, be represented by an axis angle vector, $\hat{\mathbf{w}}_t \in \mathbb{R}^3$, and translation vector, $\hat{\mathbf{b}}_t \in \mathbb{R}^3$. Their initialization and motions are defined as

$$\begin{aligned} \left[\hat{\mathbf{w}}_0, \hat{\mathbf{b}}_0\right]^\top &\sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{w},\mathbf{b},0}) \\ \left[\hat{\mathbf{w}}_{t+1}, \hat{\mathbf{b}}_{t+1}\right]^\top &\sim \mathcal{N}(\left[\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t\right]^\top, \mathbf{\Sigma}_{\mathbf{w},\mathbf{b},t+1}) \end{aligned} \tag{27}$$

where $\mathbf{\Sigma}_{\mathbf{w},\mathbf{b},t} \in \mathbb{R}^{6\times6}$ is a covariance matrix. A Weiner process is once again chosen for the same reason as the joint angle error motion model. Integration of the initial distribution and motion model in the particle filter is shown in lines 1 to 6 and 8 to 9, respectively, in Algorithm 1.

2) Observation Model: To update the lumped error from images, features need to be detected and a corresponding observation model for them must be defined. The coming observation models will generalize for any point or edge features. Let $\mathbf{m}_t$ be a list of detected point features in the image frame from the projected robot tool. By following the standard camera pin-hole model combined with (17), the camera projection equation for the kth point is

$$\hat{\mathbf{m}}_k(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t) = \frac{1}{s}\mathbf{K}\mathbf{T}^c_{b-}\mathbf{T}^{b-}_{n_b}(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t)\prod_{i=1}^{n_b}\mathbf{T}^{i-1}_i(\tilde{q}^i_t) \prod_{i=n_b+1}^{j_k}\mathbf{T}^{i-1}_i(\tilde{q}^i_t + \hat{e}^i_t)\overline{\mathbf{p}}^{j_k} \tag{28}$$

where $\frac{1}{s}\mathbf{K}$ is the camera projection operator with intrinsic matrix $\mathbf{K}$ and known location $\mathbf{p}^{j_k}$ on joint link j_k.

Similarly, let the paired lists $\boldsymbol{\rho}_t, \boldsymbol{\phi}_t$ be the parameters describing the detected edges in the image from the projected robot tool. The parameters describe an edge in the image frame using the Hough Transform [53], so the kth pair, ρ^k_t and ϕ^k_t parameterize the kth detected edge with the following:

$$\rho^k_t = u\cos(\phi^k_t) + v\sin(\phi^k_t) \tag{29}$$

where (u, v) are pixel coordinates. Let the projection equations for the ith edge be $(\hat{\rho}^i(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t), \hat{\phi}^i(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t))$. These projection equations will need to be defined based on the geometry of the robot. An example of a cylindrical shape is shown in Appendix C. Furthermore, Chaumette [54] derived the projection equations for a multitude of geometric primitives and can be referred to for additional shapes. The point and edge projections are computed on lines 18 and 21, respectively, in Algorithm 1.

From the lists of detected features, there may be false detections, and they need to be associated with the correct point position ($\mathbf{p}^{j_i}$) or edge on the robot. To accomplish this, a cost matrix $\mathbf{C}^m$ is generated between the detected and projected features. For the kth detected point feature and ith projected point, the cost is

$$C^m_{k,i} = \gamma_m||\mathbf{m}^k_t - \hat{\mathbf{m}}_i(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t)||^2 \tag{30}$$

where γ_m is a tuned parameter. Likewise a cost matrix $\mathbf{C}^l$ is computed for the edges, and the kth detected edge and ith projected edge the cost is

$$C^l_{k,i} = \gamma_\rho|\rho^k_t - \hat{\rho}_i(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t)| + \gamma_\phi|\phi^k_t - \hat{\phi}_i(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t)| \tag{31}$$

where γ_ρ and γ_ϕ are tuned parameters.

A greedy matching technique is used to make associations between the detected and projected features because of the computational efficiency. The costs are sorted from lowest to highest, and the first (k, i) pair is matched and added to the set A_m or A_l for points and edges, respectively. All subsequent costs associated with either detection k or projection i are removed from the sorted list. This is repeated until a maximum cost of C^m_{max} or C^l_{max} is reached for points and edges, respectively. By limiting the maximum cost for association, false detections can be filtered out. This association technique is conducted on lines 19 and 22 in Algorithm 1 for points and edges, respectively.

The observation model wraps the associations and their costs into a probability function dependent on the state, so the filter can update the states properly. For the list of point features, the probability is

$$P(\mathbf{m}_t|\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t) \propto (n_m - |A_m|)e^{-C^m_{max}} + \sum_{k,i\in A_m} e^{-C^m_{k,i}} \tag{32}$$

where there are a total of n_m detectable point features on the robot. Similarly, the probability of the list of detected edges is

$$P(\boldsymbol{\rho}_t, \boldsymbol{\phi}_t|\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t) \propto (n_l - |A_l|)e^{-C^l_{max}} + \sum_{k,i\in A_l} e^{-C^l_{k,i}} \tag{33}$$

where there are a total of n_l detectable edge features on the robot. The probability distributions can be viewed as a summation of Gaussian centered about the projected features. The individual Gaussian probabilities are bounded and clipped by the maximum cost for association. Clipping the Gaussian is preferred since in the cases of missed feature detections, the posterior probability from the filter does not go to zero. An additional advantage

of using a particle filter for tracking the lumped error is that these observation models do not need to be normalized. In this case, finding the normalization factor would be challenging due to the matching complexity and clipping of Gaussians. These observation models update the particle filter on lines 20 and 23 in Algorithm 1 for points and edges, respectively.

3) Modifications for Eye-in-Hand Configuration: The explicitly tracked joint errors, $\hat{\mathbf{e}}_t$, remains the same since they are still the only joints visible in the camera frame. However, the tracked pose is now $\mathbf{T}^{c_n}_{n_b}(\hat{\mathbf{w}}^l_t, \hat{\mathbf{b}}^l_t)$, described in (24). The tracked parameters $\hat{\mathbf{w}}^l_t, \hat{\mathbf{b}}^l_t \in \mathbb{R}^3$ represent the lumped error as axis-angle and translation vectors, respectively, and have the same additive zero mean Gaussian noise as described in (27). The feature detection, association, and observation models all remain the same. The only change required is modifying the camera projection equations. The camera projection equation for the ith marker is changed from (28) to

$$\hat{\mathbf{m}}_k(\hat{\mathbf{w}}^l_t, \hat{\mathbf{b}}^l_t, \hat{\mathbf{e}}_t) = \frac{1}{s}\mathbf{K}\mathbf{T}^c_{c_n}\left(\prod_{i=1}^{n}\mathbf{T}^{c_i}_{c_{i-1}}(\tilde{q}^{c_i}_t)\right)^{-1}\mathbf{T}^{c_b}_{b-}$$
$$\mathbf{T}^{c_n}_{n_b}(\hat{\mathbf{w}}^l_t, \hat{\mathbf{b}}^l_t)\prod_{i=1}^{n_b}\mathbf{T}^{i-1}_{i}(\tilde{q}^i_t)\prod_{i=n_b+1}^{j_k}\mathbf{T}^{i-1}_{i}(\tilde{q}^i_t + \hat{e}^i_t)\overline{\mathbf{p}}^{j_k} \quad (34)$$

by combining (24) with the camera pin-hole model. A similarly simple modification is required for the projected edges $(\hat{\rho}^i(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t), \hat{\phi}^i(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t))$. The example shown in Appendix C for cylindrical shapes includes the modifications necessary.

III. Experiments and Results

Since surgical robotic tool tracking is a direct application of this work, the proposed particle filter was used to track the lumped error in a simulated scene of a da Vinci Surgical System and on a real world dVRK [12]. The uncertainties of joint angles on the dVRK system are so prevalent that results relying only on base-to-camera calibration and not accounting for joint angle error were intentionally omitted in previous work due to such poor results [35]. Furthermore, in our own previously equivalent work, we experimented using either calibrated base-to-camera transform or active tracking to grasp chicken tissue detected in the camera frame [24]. Using the calibrated base-to-camera transform, the surgical tool was unable to grasp the chicken tissue. Meanwhile with active tracking, the surgical tool was able to repeatedly grasp the chicken tissue. The last experiment is tracking a partially visible Baxter robot arm which has significant backlash transmission effects. These set of tests show the effectiveness of the proposed parameter reduction technique by comparing against different parameter sets.

A. Da Vinci Simulated Scene Setup

A simulated scene in V-REP [55] was developed based on the da Vinci robot model constructed in Fontanelli *et al.* [56]. The robotic tool and camera arm simulated were a patient side manipulator (PSM) with a large needle driver and an endoscopic camera manipulator (ECM), respectively, from the da Vinci Surgical System. The PSM has 6 DoF and an additional gripper joint. The ECM is stereoscopic and has 4 DoF. The first joint link visible in the endoscopic camera frame was after the $n_b = 4$ joint as expected when operating with a da Vinci Surgical System.

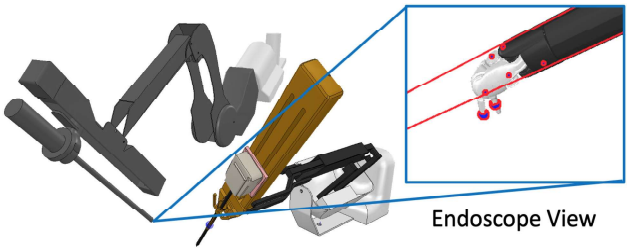

Fig. 3. Simulated scene in V-REP [56] of a patient side manipulator (PSM) and endoscopic camera manipulator (ECM) from a da Vinci Surgical System. Blue markers are placed on PSM's gripper and detected for the particle filter as shown in red in the endoscopic view. Similarly, the detected edges of the insertion shaft are highlighted with red lines and are also used by the particle filter.

Small blue spheres were placed as markers along the kinematic links near the gripper to be used as point features to update the particle filter. The blue markers were detected using standard color segmentation from OpenCV [57]. Each camera image was first converted to the hue, saturation, and value (HSV) color space. Hand-tuned lower and upper bounds for each HSV channel was then applied to the image resulting in a segmented binary image. The segmented binary image was then clustered into distinct contours from which the centroids are estimated. The list of centroids $\mathbf{m}_t$ were considered detected pixel coordinate features potentially from projected points on the surgical tool. The edges of the projected cylindrical insertion shaft of the PSM tool were also used to update the particle filter and detected using standard OpenCV functionality [57]. Each pixel potentially associated with the edges were detected using Canny edge detector [58]. The pixels were further classified into distinct edges using the Hough Transform [53] with parameters ρ^k_t and ϕ^k_t to fit (29). The simulated scene and a corresponding camera image with the detected features is shown in Fig. 3.

The error in calibration, $\mathbf{T}^{b-}_b$ was done by sampling from zero mean Gaussian in its axis angle and translation vector representations

$$\left[\mathbf{w}^{b-}, \mathbf{b}^{b-}\right]^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}^{b-}_{\mathbf{w},\mathbf{b}}). \quad (35)$$

Therefore, the initial calibration given to the filter was set to

$$\mathbf{T}^c_{b-} = \mathbf{T}^c_b\left(\mathbf{T}^{b-}_b(\mathbf{w}^{b-}, \mathbf{b}^{b-})\right)^{-1}$$
$$\mathbf{T}^{c_b}_{b-} = \mathbf{T}^{c_b}_b\left(\mathbf{T}^{b-}_b(\mathbf{w}^{b-}, \mathbf{b}^{b-})\right)^{-1} \quad (36)$$

for the stationary camera and eye-in-hand cases, respectively, where $\mathbf{T}^c_b$ and $\mathbf{T}^{c_b}_b$ were given by the simulator.

The joint error for the PSM was simulated as a summation between a uniformly sampled bias at the start of each trial and linear cable stretch. Written explicitly, the error for joint angle i was defined as

$$e^i_t = e^i_b + e^i_c q^i_t \quad (37)$$

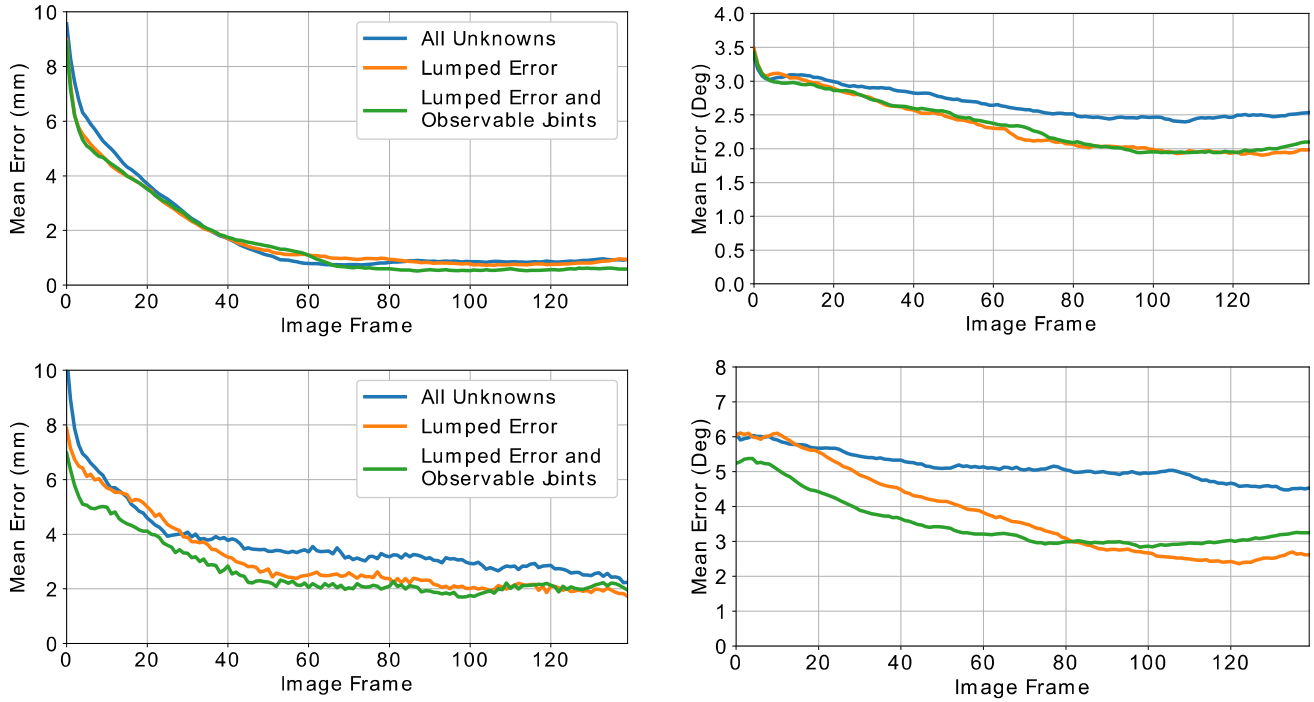

Fig. 4. Mean end-effector pose error in the camera frame over time under various tracking configurations from simulated da Vinci scene. The top and bottom row of plots are measured from the stationary and eye-in-hand cases, respectively. These mean error trends are calculated with 50 trials and shows that tracking the lumped error results in a lower end-effector orientation error compared to tracking all unknowns.

where $e_b^i \sim \mathcal{U}(-a_e^{i,b}, a_e^{i,b})$, e_c^i is the linear cable stretch coefficient, and q_t^i is the correct joint angle from the PSM. Similarly, joint error c_i for the ECM was defined as

$$e_t^{c_i} = e_b^{c_i} + e_{l,t}^{c_i} \tag{38}$$

where $e_b^{c_i} \sim \mathcal{U}(-a_e^{c_i,b}, a_e^{c_i,b})$ was sampled once at the start of each trial and $e_{l,t}^{c_i} \sim \mathcal{N}(0, \sigma_{c_i,l}^2)$ sampled at every time step to simulate the uncertainties in the robotic endoscopes joint angles.

The PSM arms configuration was set via V-REP's inverse kinematics. Its position moved along a preset cyclical trajectory added with a small, random sample from a zero mean Gaussian with standard deviation of 1 mm. The gripper joint opened and closed at a similar cyclical rate. Likewise, the four joint angles of the ECM were set to move in a cyclical pattern in the eye-in-hand case. The orientation of PSM end-effector instead takes a random walk starting at a preset value by rotating an additional uniformly sampled rotation at every time step. Note that this represents the most complex scenario where every part of the robot (manipulator, gripper, camera) are continuously moving on independent paths hence testing the proposed tracking method in a larger variety of scenarios including occlusions of features. Additional details and parameter values are described in Appendix D.

B. Tracking Lumped Error in Da Vinci Simulated Scene

The particle filter configurations evaluated were the following.

1) *All unknowns:* tracking all joint angle errors and base-to-camera transform or base-to-base in the stationary and eye-in-hand case, respectively. Done by setting $n_b = 0$ in the particle filter.

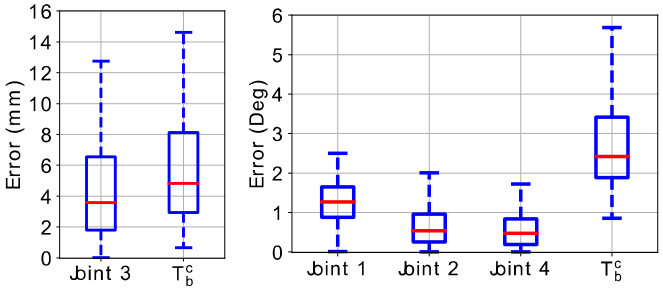

Fig. 5. Distribution of first 4 joint angle errors, whose preceding kinematic links are never in the camera frame, and the stationary camera to base transform $\mathbf{T}_b^c$ error when explicitly estimating them in the simulated da Vinci scene. Errors up to 14 mm and 5° highlight the inability to estimate these unknown values explicitly due to the parameters being nonidentifiable as shown in Claim 1.

2) *Lumped error:* applying (18) to the particle filter.
3) *Lumped error and observable joints:* no modifications to the described particle filter.

Both stationary camera and moving camera arm scenarios were tested. Each configuration was repeated 50 times to test for consistent performance.

To evaluate the effectiveness of pose or transform estimation, the error was calculated at time t as

$$\epsilon_{\mathbf{b}} = ||\mathbf{b}_t - \hat{\mathbf{b}}_t|| \qquad \epsilon_{\mathbf{w}} = ||\mathbf{w}_t^r|| \tag{39}$$

where $\mathbf{w}_t^r$ is the axis angle representation of $\mathbf{R}_t(\hat{\mathbf{R}}_t)^{-1}$, $\mathbf{b}_t \in \mathbb{R}^3$ and $\mathbf{R}_t \in SO(3)$ are the ground truth translation vector and rotation matrix, respectively, and $\hat{\mathbf{b}}_t \in \mathbb{R}^3$ and $\hat{\mathbf{R}}_t \in SO(3)$ are the tracked translation vector and rotation matrix, respectively.

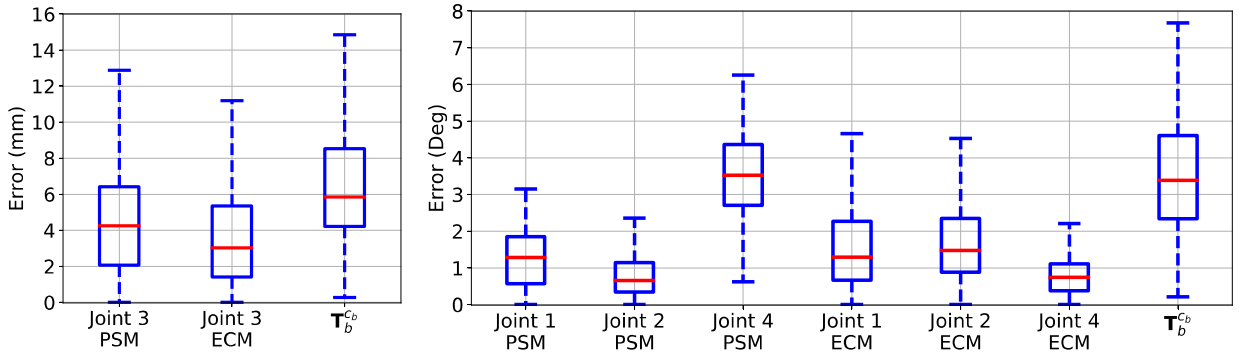

Fig. 6. Box plots of the tracked joint angle errors, whose preceding kinematic links are never in the camera frame, and the base of camera arm to base of robotic tool transform $\mathbf{T}_b^{c_b}$ error when explicitly estimating all unknowns in the simulated da Vinci scene. Errors up to 14 mm and 7° highlight the inability to estimate these unknown values explicitly due to the parameters being nonidentifiable as shown in Claim 1.

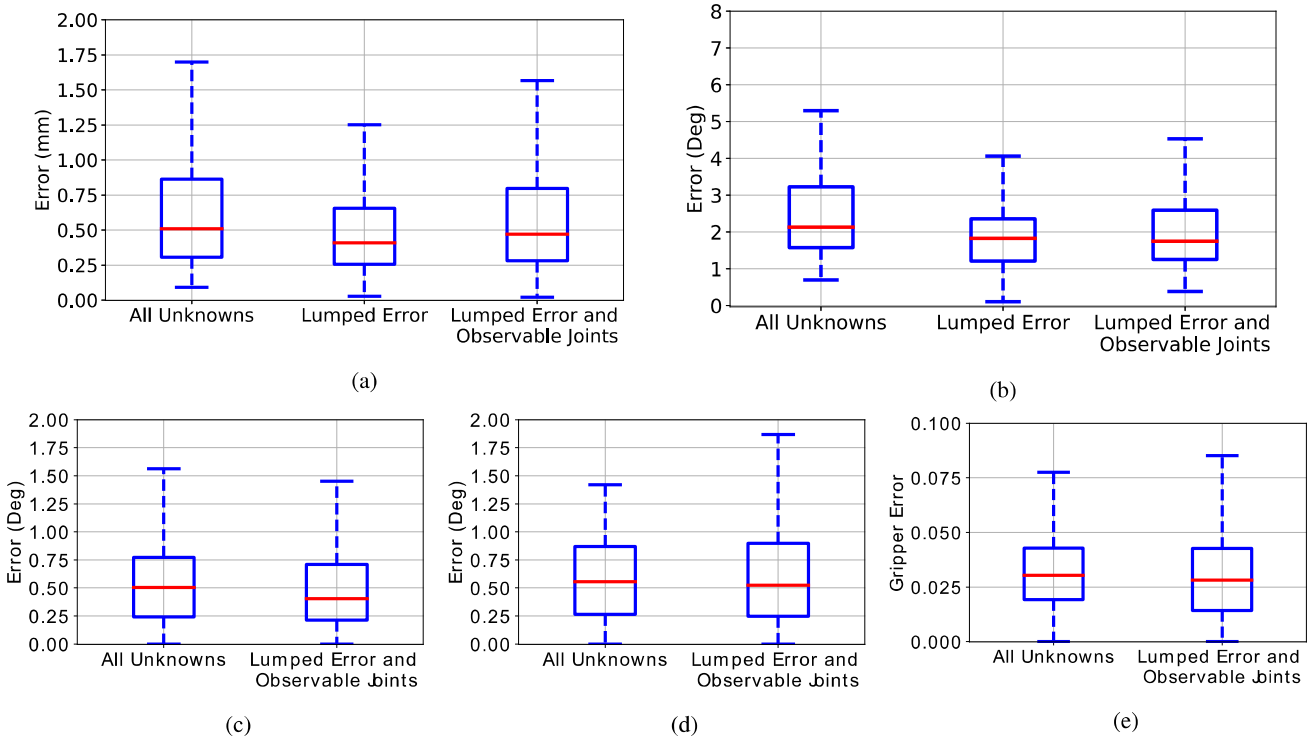

Fig. 7. Box plots of converged tracking performance under various configurations for the particle filter in simulated da Vinci for stationary camera. As is evident from the box plots, the lumped error results in better converged end-effector pose error compared to tracking all unknowns. Meanwhile, no significant difference in observable joint angle error is seen. (a) End-effector position error. (b) End-effector orientation error. (c) Joint 5 error. (d) Joint 6 error. (e) Joint 7 error.

The ith joint angle error was computed as

$$\epsilon_{q^i} = |\hat{q}_t^i - q_t^i| j. \tag{40}$$

The mean end-effector pose error plots are shown in Fig. 4 for both stationary camera and eye-in-hand cases. Fig. 5 shows the distributions of errors for the nonidentifiable joint angles and base-to-camera transform in the stationary camera case when explicitly tracking all unknown parameters. Fig. 6 shows the distributions of errors in the eye-in-hand configuration for the nonidentifiable joint angles and base-to-base transform when explicitly tracking all unknown parameters. These values were calculated across 40 time steps from all 50 trials after 100 time steps to give time for the particle filter to converge. These errors have a large spread even though the particle filter is still able to sufficiently track the end-effector as seen in the mean end-effector pose error plots in Fig. 4. This supports Claim 1 by showing that it is infeasible to explicitly track all the unknown parameters from partially visible robotic tools since they do not converge to their true values.

The distribution of end-effector tracking errors after giving the particle filter time to converge in the same manner as previously described are shown in Figs. 7 and 8 for stationary camera and moving camera arm, respectively. The converged distributions of error show little difference in end-effector positional error. However, end-effector orientation error was clearly improved by using the lumped error estimation for both the stationary camera and robotic camera arm cases. For the observable joints, joints $5, 6, 7$, the lumped error tracking method showed no significant

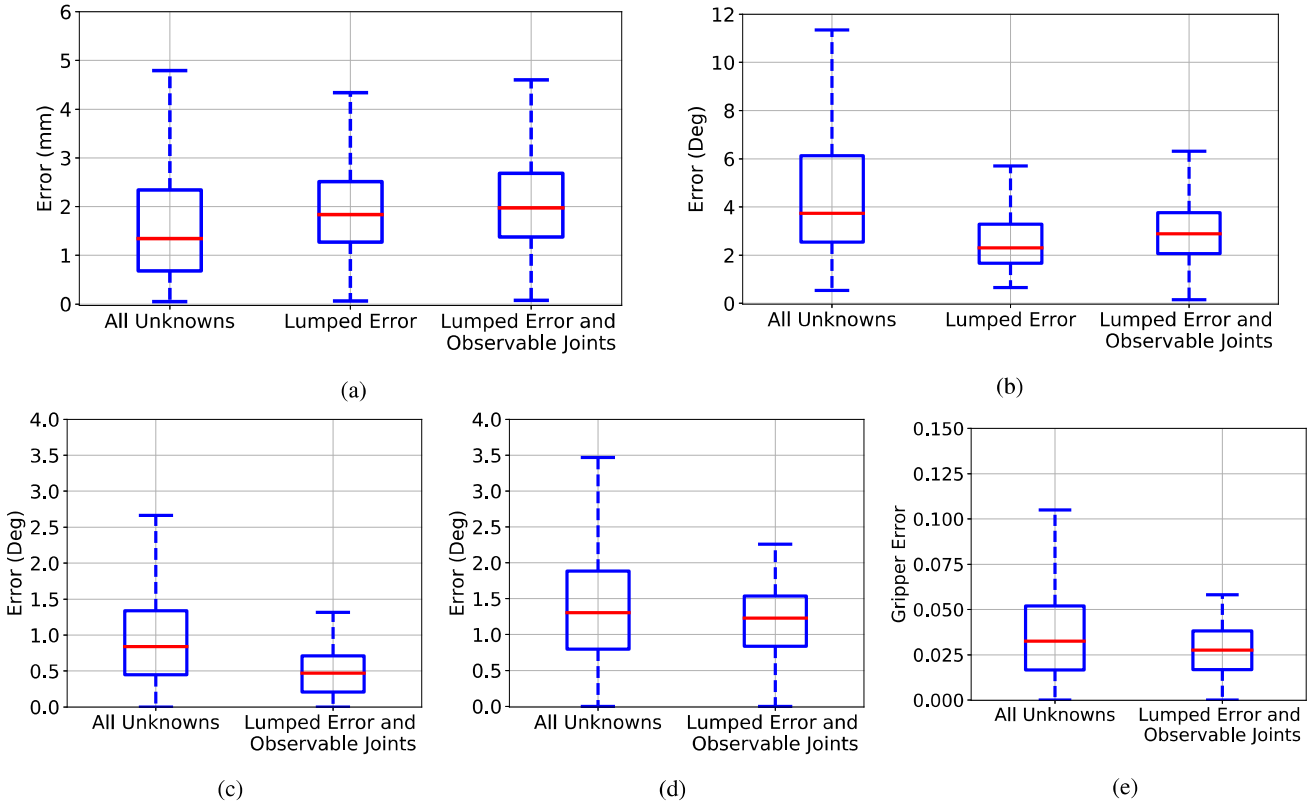

Fig. 8. Box plots of converged tracking performance under various configurations for the particle filter in simulated da Vinci scene for eye-in-hand configuration. As is evident from the box plots, the lumped error results in better converged end-effector orientation error and observable joint angle errors. (a) End-effector position error. (b) End-effector orientation error. (c) Joint 5 error. (d) Joint 6 error. (e) Joint 7 error.

difference in error between the stationary camera and eye-in-hand cases. Meanwhile, when explicitly tracking all unknowns, the error in observable joints was significantly worse in the eye-in-hand case.

C. Tracking Lumped Error on dVRK

Two one-minute segments of encoder readings and stereoscopic data from the endoscope on an ECM was captured from a dVRK [12]. The stereoscopic camera system used the standard dVRK endoscopic lens and has a resolution of 1920 by 1080 pixels at 30FPS. In both sequences, a single PSM arm was teloperated with gripper in full view of the stereoscopic camera. In the first sequence, the PSM arm travelled a total distance of 48 mm, and the ECM was stationary. In the second sequence, the PSM arm travelled a total distance of 49 mm, and the ECM arm joint angles were set to sinusoidal patterns similar to the previous simulation experiment resulting in 35 mm for a total distance travelled. The PSM arm had blue colored markers painted on. The markers and edges of the projected cylindrical insertion shaft were detected in the same manner as the simulated scene. The initial calibration $\mathbf{T}_{b-}^{c}$ and $\mathbf{T}_{b-}^{c_b}$ were computed using OpenCV's solvePnP [57] with manually set associations of the markers. On a deployed, and fully assembled da Vinci Surgical System, we envision that these initial transformations are computed from the set up joints, which connect the ECM with the PSM arm. However, the dVRK by default does not come with set up joints, which is why we elected to the sovlePnP with manually set associations for initialization.

From both sequences, 20 evenly distributed images were manually annotated using the VGG labeller [59]. These labels $\mathbf{I}_G$ were considered ground truth and intersection over union (IoU) was used as the metric in this experiment

$$\frac{\mathbf{I}_R \cap \mathbf{I}_G}{\mathbf{I}_R \cup \mathbf{I}_G} \tag{41}$$

where $\mathbf{I}_R$ was a generated mask using our previously developed rendering procedure [60]. The generated mask was rendered using the tracked parameters. The distributions of IoU are shown in Fig. 9 for both stationary and moving camera arm cases. Similar to the simulation results, the lumped error clearly performed the best. Examples of surgical tool renderings on top of the image feed are shown in Fig. 10.

D. Tracking Lumped Error on Baxter

A 77 s video segment was recorded of a 7-DoF arm from the Baxter robot with corresponding joint reading data. The first joint link consistently visible in the image frames is after the $n_b = 6$ joint, and the end-effector moved a total distance of 5.25 m during the segment. The video was captured on Microsoft's Azure Kinect camera which has and RGB camera and a depth camera. In this experiment, the particle filter which tracked this robotic arm only used the mono-RGB camera data.

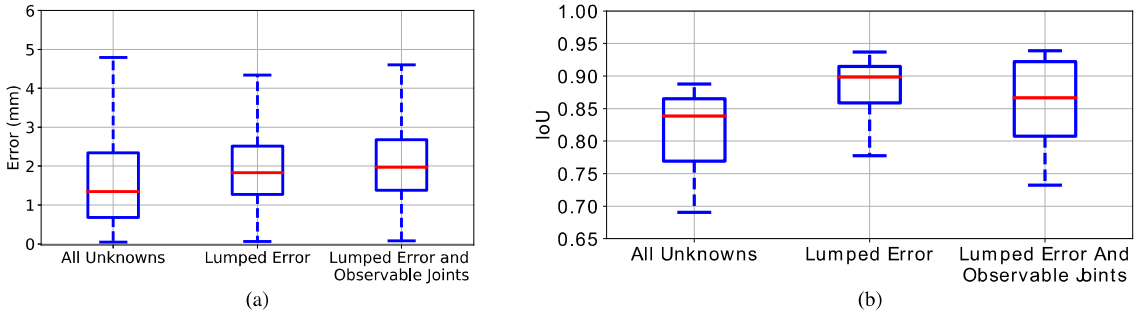

Fig. 9. Distribution of IoU between manual annotations and reprojected rendering from tracked values under various particle filter configurations on the dVRK [12]. The left and right plots correspond to stationary camera and eye-in-hand cases respectively. The plots show tracking the lumped error yields better tool tracking rather than explicitly tracking all unknowns.

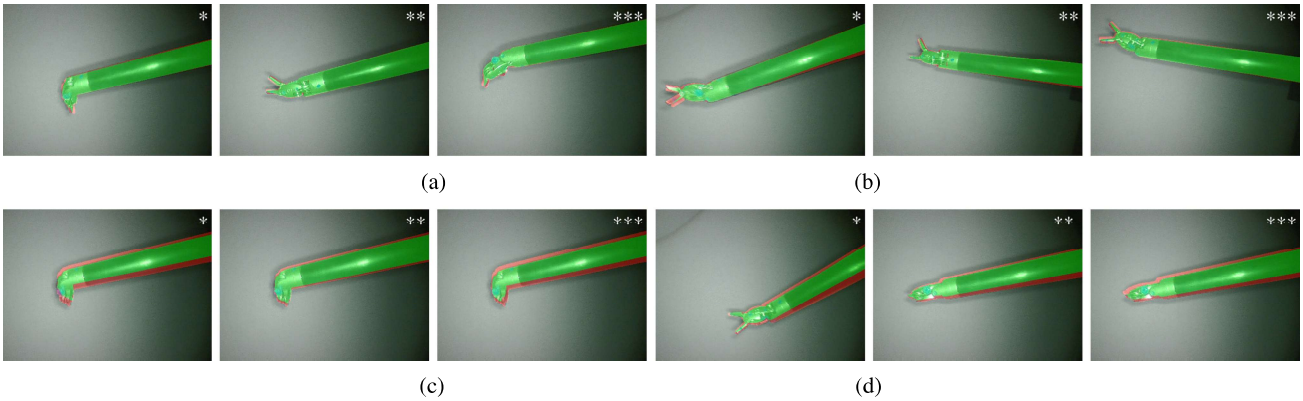

Fig. 10. Images from the best and worst IoU when reprojecting the tracked dVRK [12] surgical tool onto the image. The green and red regions are the intersection and error, union minus intersection, respectively between the re-projection and surgical tool (best seen in color). The tracking conditions for all sets of three, from left to right, are: All unknowns*, lumped error**, lumped error and observable joints***. The corresponding IoU values are also listed. This shows that the failures when using lumped error are substantially less severe than tracking all unknowns. (a) Best IoU for stationary camera: $0.94^{*}, 0.96^{**}, 0.96^{***}$. (b) Best IoU for moving camera arm: $0.89^{*}, 0.94^{**}, 0.94^{***}$. (c) Worst IoU for stationary camera: $0.69^{*}, 0.85^{**}, 0.70^{***}$. (d) Worst IoU for moving camera arm: $0.69^{*}, 0.76^{**}, 0.73^{***}$.

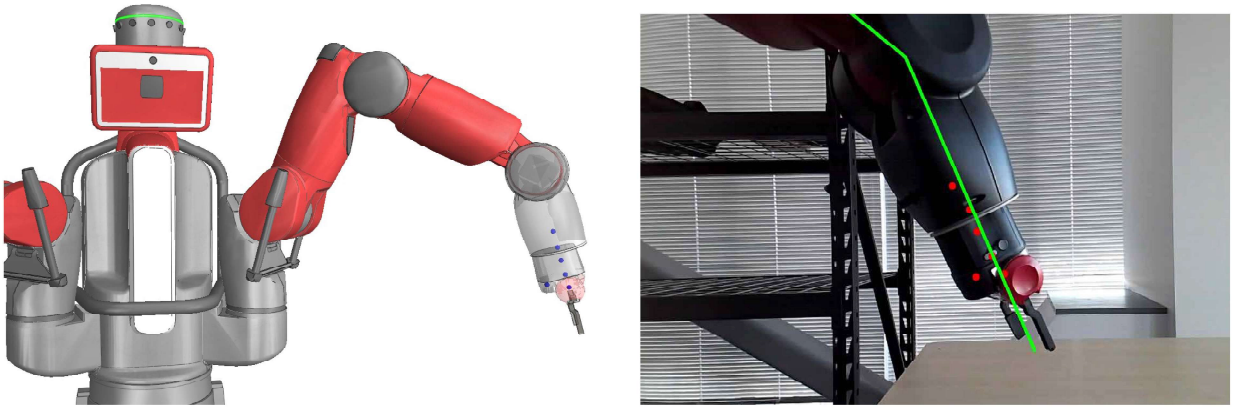

Fig. 11. DNN was trained in simulation to detect keypoints on a partially visible Baxter robot arm. The keypoint locations on the visible portion of the Baxter arm in this experiment were optimized for such that their detectability and accuracy is maximized [21], and the result is shown with blue circles in the left figure. An example of the detections from the Baxter experiment is shown in red on the right figure. These detections were used to update the particle filter tracking the Lumped Error of the partially visible Baxter arm. A reprojected skeleton of the Baxter using the particle filter is also shown in green on the right figure.

Meanwhile the depth images were used to evaluate performance of tracking the robotic tool.

The features used to update the particle filter for this experiment were detected using a deep neural network (DNN) rather than markers as used in the previous experiments. This was done to show the flexibility of the presented particle filter with regards to the features used to update it. The DNN chosen was DeepLabCut [61], and it was trained to detect and optimize feature points in simulation using our previously developed method [21]. The resulting feature points which were detected by the DNN are shown in Fig. 11. The DeepLabCut detections also provide direct associations for the feature points, A_m, and a confidence value, $\eta_t^k \in [0, 1]$, for each detected feature k. To integrate this with the lumped error tracking, the point feature

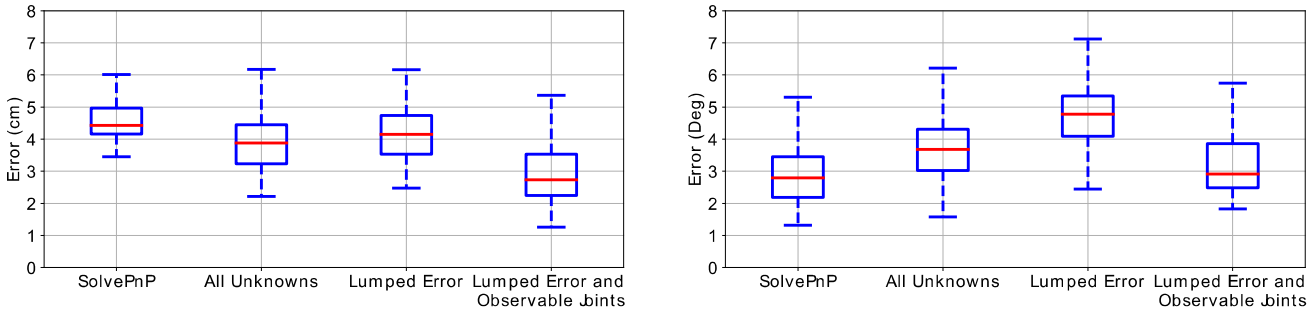

Fig. 12. Distribution of position and orientation errors when calibrating for the base-to-camera transform alone with solvePnP and various particle filter configurations compensating for errors in the base-to-camera transform and joint angles from the Baxter robot experiment. Out of the active tracking methods, the lumped error parameter reduction technique with the observable joints is the most effective. Meanwhile, solvePnP for the static base-to-camera transform performs similar to Lumped Error and Observable joints in orientation error, but performs significantly worse in positional error.

observation model in (32) was modified to

$$P(\mathbf{m}_t|\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t) \propto \sum_{k,i\in A_m} \eta_t^k e^{-\gamma_m||\mathbf{m}_t^k - \hat{\mathbf{m}}_i(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{e}}_t)||} \quad (42)$$

which removes the association step in line 19 from Algorithm 1. It is important to include the DNN's confidence η_t^k in the model because sometimes the detections can be poor and the corresponding update needs to be weighted lower. In the original observation model, (32), this was done in the association step where the maximum cost is thresholded at C_{max}^m.

To evaluate tracking performance, the depth images were compared against the reconstructed robotic arm using the tracked parameters. The reconstructed scene was rendered using a virtual camera and a Baxter robot model in V-REP [55]. After every image used to update the particle filter, the virtual camera captured a depth image of the reconstructed scene. The tracking error was defined as the relative transform, $\mathbf{T}(\mathbf{w}^\epsilon, \mathbf{b}^\epsilon) \in SE(3)$, between the rendered point cloud from the virtual camera $\mathbf{R}$ and the corresponding point cloud from Kinect Azure's depth camera $\mathbf{G}$. This relative transform is calculated by minimizing

$$\sum_{(\mathbf{r},\mathbf{g})\in\mathcal{K}} ||\bar{\mathbf{r}} - \mathbf{T}(\mathbf{w}^\epsilon, \mathbf{b}^\epsilon)\bar{\mathbf{g}}|| \quad (43)$$

where $\mathbf{r} \in \mathbf{R}$, $\mathbf{g} \in \mathbf{G}$, and $\mathcal{K}$ is the correspondence set between $\mathbf{R}$ and $\mathbf{G}$. The optimization was solved using Open3D's [62] implementation of the iterative closest point algorithm [63]. To filter out poorly converged values, only the results where the amount of corresponded points relative to the total rendered points, $|\mathcal{K}|/|\mathbf{R}|$, is greater than 0.7 were recorded. Similar to the dVRK experiments, the initial calibration, $\mathbf{T}_{b-}^c$, was computed using OpenCV's solvePnP [57] using the detected features and their associations on the first image frame.

For additional comparison, we ran OpenCV's solvePnP implementation [57] over the first 20 images of the dataset to solve for a static base-to-camera transform $\mathbf{T}_b^c$. The 2-D detections are the same used by the particle filter. Their corresponding 3-D positions in the base frame of the Baxter robot are generated using forward kinematics with the joint angle readings $\tilde{q}_t^i$. The resulting $\mathbf{T}_b^c$ and the joint angle readings are used to generate a rendered point cloud $\mathbf{R}$ in V-REP [56], and the same error metric as previously described is computed.

The distribution of translational errors $||\mathbf{b}^\epsilon||$ and orientation errors $||\mathbf{w}^\epsilon||$ for the three different particle filter configurations are plotted in Fig. 12. In this case, lumped error with observable joints performed the best. We believe this is since the kinematic links on the Baxter are much larger hence making the simplification from (18) no longer valid.

IV. Discussion

From the experimental results and their respective metrics, it is evident that using lumped error yields almost always better tool tracking than explicitly estimating all unknowns for both stationary camera and eye-in-hand cases. Only two experimental metrics, Figs. 8(a) and 12(b), performed marginally better when tracking all unknowns. Nevertheless, these metrics cannot be viewed in isolation, and their respectively paired metrics, Figs. 8(b)–(e) and 12(a), showed significant improvement when tracking the lumped error over all unknowns. Furthermore, the nonidentifiable values estimated when tracking all unknowns yielded nonrealistic results as shown in Figs. 5 and 6. This is due to there being a single solution to the parameters being estimated when tracking the lumped error rather than the infinite set of solutions when tracking all unknowns, as shown in Claim 1. Due to the infinite set of solutions, large distributions of the parameters being estimated occurred which is seen in Figs. 5 and 6 when tracking all unknowns. From the perspective of the particle filter, this is an inefficient usage of particles and detrimental to tracking because the particle filter estimates the posterior probability using a finite number of samples.

Tracking the observable joints when using the lumped error showed little difference in end-effector accuracy to not tracking them in the da Vinci simulation. This supports the validity of applying simplification in (18) to the da Vinci robot because the link lengths for the observable joints, a dexterous robotic gripper, are short. However, not tracking the observable joints on the real-world dVRK performed better. We believe this occurred due to the features not being detected as consistently in the real world as in simulation, hence highlighting the usefulness of the simplification in (18). Meanwhile for the Baxter experiment, tracking the observable joint angles gave better performance than using the simplification. In contrast to the da Vinci robot, the link lengths for the observable joints in the Baxter experiment

are long hence making errors in their joint angles more catastrophic. This matches with our motivation for the simplification in (18), and that it should only be used when the errors from the observable joints does not propagate through the kinematic chain drastically (e.g., a robotic gripper).

In addition to being efficient with respect to parameters to estimate, the lumped error modeled as a Weiner process experimentally was found to compensate various distributions of errors. In the stationary camera arm case in simulation, it captures both linear cable stretch and uniform bias error from joint angles. For the eye-in-hand case in simulation, tracking the lumped error additionally compensated for Gaussian noise from the camera arm. Meanwhile in the real world experiments, there are significant nonlinear cable stretch effects on the PSM arm [8] and backlash and hysteresis on the Baxter robot and ECM arm, all of which the lumped error successfully compensated.

The proposed particle filter was also shown to be effective with a variety of visual features to update the lumped error. The surgical robotic experiments used colored markers and edge detections of a cylindrical shaft as features which also needed to be associated. The Baxter robot experiment and previously equivalent work, SuPer Deep [27], instead used a DNN to extract features with association. The DNN feature extraction does perform better, as shown in our previous work [27], but the marker based approach is still sufficient for precise control as shown in the previous autonomous suction [64] and needle regrasping [65] works. This discrepancy in performance is due to the improved accuracy in feature detection, which directly improves the accuracy of the particle filters output. Furthermore, we observed better tracking performance when the visible robotic tool is closer to the camera as the features (learned or markers) are more accurately detected.

The particle filter in all of the experiments shown here and the previous ones just described ran on a Intel Core™ i9-7940X Processor and NVIDIA's GeForce RTX 2080 and yielded a loop rate of 24FPS and 10FPS when using the colored markers and DNN, respectively. Therefore, it is suitable for real-time applications which is further highlighted by our previous control experiments. These previous works in surgical robotic control and their usage of lumped error are discussed in Appendix V-A.

We also showed that the lumped error is mathematically equivalent to a previous, popular surgical tool tracking formulation which claimed to only track the error in the transform from the camera frame to the base of the surgical robot [26]. The equivalency implies that this previous formulation actually was compensating for both error in base-to-camera transform and joint angle errors. The surgical tool tracking experiments presented here focused on the parameter reduction technique. For performance of surgical tool tracking in surgical environments, refer to our previously developed work which utilized an equivalent tool tracking method [24], [27], [64], [65].

Lastly, we want to highlight that the tracking method presented in this work requires knowledge of the kinematic chain through the joint transforms $\mathbf{T}_{i-1}^{i}(\cdot)$ and camera intrinsics $\mathbf{K}$. This is a fair assumption as the kinematic chain for a robot is typically supplied by the manufacturer. Furthermore, the joint transforms can be calibrated for with high accuracy offline [66]. Camera calibration is also a well studied method [67] and even implemented in standard vision toolboxes such as OpenCV [57]. Nonetheless, accurate calibration of these parameters is required to implement the proposed tracking method in this work.

V. Conclusion

In this work, we described the challenges of tracking robotic tools from visual observations that only showed part of the kinematic chain and there was uncertainty in base-to-camera transform and joint angle measurements through a problem formulation that shows it was infeasible to directly estimate all of these unknowns. A smaller set of parameters, which we coined as lumped error, was derived and shown to compensate all the described uncertainties and was identifiable. Furthermore, lumped error was mathematically equivalent to a popular instrument tracking method [26] hence giving a deeper understanding of how it worked. Tracking the lumped error experimentally was shown to efficiently track robotic tools and even could be extended to eye-in-hand configurations. Through this extension, we successfully tracked for the first time a surgical robotic tool with a moving endoscope, which contained a total of 10 DoF and a gripper joint.

The proposed tracking method to estimate the lumped error used a Weiner Process to model the uncertainty and experimentally was found to be efficient. In future work, we intend to use the analytical derivation of the lumped error to more precisely describe the uncertainty. In particular, we will use cable stretch models to describe the joint angle error so the uncertainty of the motion model for the lumped error appropriately propagates the transmission errors for cable driven robots such as dVRK [12]. Additionally, we also intend to investigate controllers which utilize the lumped error parameter reduction in future work. The controllers presented in the previous autonomous suction [64] and suture needle regrasping [65] works show great promise to the capabilities of a lumped error controller. These controllers will be investigated from a theoretical perspective where criteria for stability will be defined.

Appendix

A. Previous Works Using Lumped Error Tracking for Surgical Robotic Control

The lumped error simplification has been used to great effect in previous surgical robotic work under the guise of tracking KCS [23]. The summary here gives insight into how this tracking method has enabled our previous work in surgical robotic control. In addition, minor adjustments are described which show how these previous works fit into the unified approach for tool tracking presented here.

1) Position Control: In order to regulate a robotic suction tool along a motion plan to clear the surgical field of blood, a controller which uses lumped error was implemented [64]. An example of this motion plan is shown in Fig. 13. In this automated suction work, the robotic suction tool's lumped error was tracked using painted markers for point features and the cylindrical insertion shaft, similar to the surgical tool tracking experiments. Since the motion plan generates goal positions

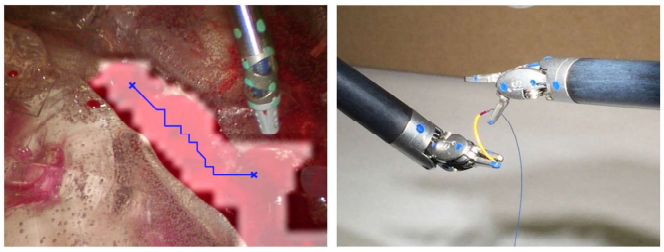

Fig. 13. From left to right respectively, the figures show autonomous suction to clear the surgical field of blood for hemostasis [64] and automated needle regrasping for suture throwing [65]. Both of these efforts used a controller to regulate the robotic surgical tools in the camera frame as the goals, blood and suture needles, are detected and tracked in it. The controllers utilize the lumped error to accomplish the regulation.

in the camera frame, which will be denoted as $\mathbf{p}_g^c \in \mathbb{R}^3$, the controller regulates the end-effector of the robotic suction tool in the camera frame. Let $\mathbf{b}_t^e \in \mathbb{R}^3$ be the incorrect position of the end-effector in the robot base frame which was computed through forward kinematics with the noisy joint angle measurements $\tilde{q}_t^i$. The controller iteratively transforms the goal $\mathbf{p}_g^c$ to the virtual base defined by the lumped error, and the error $\mathbf{d}_t^e \in \mathbb{R}^3$, in the virtual base was computed as

$$\mathbf{d}_t^e = \left(\mathbf{T}_{b-}^c \mathbf{T}_{n_b}^{b-}(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t)\right)^{-1} \overline{\mathbf{p}}_g^c - \overline{\mathbf{b}}_t^e. \tag{44}$$

This error was then used to update the end-effector position

$$\overline{\mathbf{b}}_{t+1}^e = \begin{cases} \gamma_s \frac{\mathbf{d}_t^e}{||\mathbf{d}_t^e||} + \overline{\mathbf{b}}_t^e & \text{if } ||\mathbf{d}_t^e|| > \gamma_s \\ \mathbf{d}_t^e + \overline{\mathbf{b}}_t^e & \text{if } ||\mathbf{d}_t^e|| \leq \gamma_s \end{cases} \tag{45}$$

where γ_s is the max step size. The updated end-effector position $\mathbf{b}_{t+1}^e$ was set on the robotic suction tool using inverse kinematics and joint level regulators which use the noisy joint angle readings $\tilde{q}_t^i$ as feedback. These operations were repeated until the error $||\mathbf{d}_t^e||$ was less than some threshold, and then a new goal position was set from the motion planner. The resulting motion was an effective controller used to automate clearing of the surgical field from blood for hemostasis.

2) Orientation Control: The lumped error parameter reduction was also used to regulate robotic large needle drivers along motion plans to conduct suture needle regrasping. Similar to the previously described autonomous suction task, the motion plan was generated in the camera frame which acts as a bridge between the tracked surgical robotic tools and reconstructed suture needle. The lumped errors of the two surgical robotic tools were tracked using the same point and edge features as described in the surgical tool tracking experiments. Their positions were regulated using the same controller as described in (44) and (45). The orientation was regulated in a similar fashion. Let $\mathbf{R}_g^c \in SO(3)$ and $\mathbf{R}_t^e \in SO(3)$ be the goal orientation in the camera frame and incorrect orientation of the end-effector in the robot base frame computed through forward kinematics with the noisy joint angle measurements $\tilde{q}_t^i$, respectively. The orientation of the end-effector is iteratively set to

$$\mathbf{R}_t^e = \left(\mathbf{R}_{b-}^c \mathbf{R}_{n_b}^{b-}(\hat{\mathbf{w}}_t)\right)^{-1} \mathbf{R}_g^c \tag{46}$$

where $\mathbf{R}_{b-}^c \in SO(3)$ and $\mathbf{R}_{n_b}^{b-}(\hat{\mathbf{w}}_t) \in SO(3)$ are the rotation matrix of $\mathbf{T}_{b-}^c$ and $\mathbf{T}_{n_b}^{b-}(\hat{\mathbf{w}}_t, \hat{\mathbf{b}}_t)$, respectively. Similar to the position of the end-effector, the orientation $\mathbf{R}_t^e$ is set using inverse kinematics and joint level regulators which use the noisy joint angle readings $\tilde{q}_t^i$ as feedback. The controller effectively regulates the surgical robotic tools along the generated motion plan to complete the task of suture needle regrasping as shown in Fig. 13.

TABLE I
PARAMETERS USED FOR PARTICLE FILTER TO TRACK DA VINCI TOOLS IN SIMULATION AND DVRK. ALL ANGLES AND DISTANCES ARE IN RADIANS AND MILLIMETERS, RESPECTIVELY

Parameter	Value
$\mathbf{a}_{\hat{\mathbf{e}}}$	$\begin{bmatrix} 0.004 & 0.004 & 2 & 0.004 \\ 0.004 & 0.004 & 0.01 \end{bmatrix}$
$\mathbf{\Sigma}_{\hat{\mathbf{e}},t}$	$\text{diag}\left(\begin{bmatrix} 0.0025 & 0.0025 & 1 \\ 0.0025 & 0.0025 & 0.0025 \\ 0.005 \end{bmatrix}\right)$
$\mathbf{a}_{\hat{\mathbf{e}}^c}$	$\begin{bmatrix} 0.004 & 0.004 & 2 & 0.004 \end{bmatrix}$
$\mathbf{\Sigma}_{\hat{\mathbf{e}}^c,t}$	$\text{diag}\left(\begin{bmatrix} 0.01 & 0.01 & 2.5 \\ 0.01 \end{bmatrix}\right)$
$\mathbf{\Sigma}_{\mathbf{w},\mathbf{b},t}$	$\text{diag}(\begin{bmatrix} \mathbf{\Sigma}_{\mathbf{w},t} & \mathbf{\Sigma}_{\mathbf{b},t} \end{bmatrix})$
$\mathbf{\Sigma}_{\mathbf{w},t}$	$\text{diag}(\begin{bmatrix} 0.005 & 0.005 & 005 \end{bmatrix})$
$\mathbf{\Sigma}_{\mathbf{b},t}$	$\text{diag}(\begin{bmatrix} 0.25 & 0.25 & 0.25 \end{bmatrix})$
$\mathbf{\Sigma}_{\mathbf{w},\mathbf{b},0}$	$10(\mathbf{\Sigma}_{\mathbf{w},\mathbf{b},t})$
$\begin{bmatrix} \gamma_m & \gamma_\phi & \gamma_\rho \end{bmatrix}$	$\begin{bmatrix} 0.15 & 40.0 & 0.1 \end{bmatrix}$
$\begin{bmatrix} C_{max}^m & C_{max}^l \end{bmatrix}$	$\begin{bmatrix} 25\gamma_m & 0.1\gamma_\phi + 25\gamma_\rho \end{bmatrix}$
$\begin{bmatrix} N & N_{eff} \end{bmatrix}$	$\begin{bmatrix} 1000 & 0.5 \end{bmatrix}$

B. Implementation Details for Particle Filter

The parameters used to describe the motion models and observation models for da Vinci in simulation and dVRK tracking experiments are shown in Table I. These values were chosen based on our previously equivalent work [24]. The only modification for the nonstationary robotic endoscope case is the covariance for the lumped errors, $\mathbf{\Sigma}_{\mathbf{w},t}$ and $\mathbf{\Sigma}_{\mathbf{b},t}$, are scaled by 2. Note that the relationship of $\mathbf{\Sigma}_{\mathbf{w},\mathbf{b},0} = 10(\mathbf{\Sigma}_{\mathbf{w},\mathbf{b},t})$ is still kept after the scaling. The markers are located in similar locations as the detected features from previous work by Ye *et al.* [29]. All marker locations relative to the joint coordinate frames, $\mathbf{p}^{j_i}$ from (28), were measured using calipers on the dVRK. For the Baxter experiment, the parameters used in the particle filter are listed in Table II. These values were chosen based on a previously developed report on Baxter's performance [68].

C. Camera Projection Equation for Cylinder

The camera projection of a cylinder is an adaptation of previous work by Chaumette [54]. A cylinder is described by three parameters: a radius $r \in \mathbb{R}$, a directional vector $\mathbf{d}^j \in \mathbb{R}^3$ of its center axis, and a position along its center axis $\mathbf{p}_0^j \in \mathbb{R}^3$. Let $\mathbf{d}^j$ and $\mathbf{p}_0^j$ be defined in joint frame j, which is the insertion shaft, and $||\mathbf{d}^j|| = 1$. Using (17) or (24) for

TABLE II
PARAMETERS USED FOR PARTICLE FILTER TO TRACK BAXTER ROBOT. ALL ANGLES AND DISTANCES ARE IN RADIANS AND MILLIMETERS, RESPECTIVELY

Parameter	Value
$\mathbf{a}_{\hat{\mathbf{e}}}$	$[0.01 \quad 0.01 \quad 0.01 \quad 0.01 \quad 0.01 \quad 0.01 \quad 0.01]$
$\mathbf{\Sigma}_{\hat{\mathbf{e}},t}$	$\text{diag}\begin{pmatrix}[0.001 & 0.001 & 0.001 \\ 0.001 & 0.001 & 0.001 \\ 0.001]\end{pmatrix}$
$\mathbf{\Sigma}_{\mathbf{w},\mathbf{b},t}$	$\text{diag}(\begin{bmatrix}\mathbf{\Sigma}_{\mathbf{w},t} & \mathbf{\Sigma}_{\mathbf{b},t}\end{bmatrix})$
$\mathbf{\Sigma}_{\mathbf{w},t}$	$\text{diag}(\begin{bmatrix}0.001 & 0.001 & 0.001\end{bmatrix})$
$\mathbf{\Sigma}_{\mathbf{b},t}$	$\text{diag}(\begin{bmatrix}0.25 & 0.25 & 0.25\end{bmatrix})$
$\mathbf{\Sigma}_{\mathbf{w},\mathbf{b},0}$	$10(\mathbf{\Sigma}_{\mathbf{w},\mathbf{b},t})$
$\begin{bmatrix}\gamma_m & N & N_{eff}\end{bmatrix}$	$\begin{bmatrix}5 & 200 & 0.5\end{bmatrix}$

the stationary endoscope or robotic endoscope cases, respectively, $\mathbf{d}^j$ and $\mathbf{p}_0^j$ are transformed to the camera frame and denoted as $\mathbf{d}^c(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t) = \begin{bmatrix}a^c & b^c & c^c\end{bmatrix}^\top$ and $\mathbf{p}_0^c(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t) = \begin{bmatrix}x_0^c & y_0^c & z_0^c\end{bmatrix}^\top$, respectively. Note that $(\mathbf{w}_t, \mathbf{b}_t)$ should be replaced with $(\mathbf{w}_t^l, \mathbf{b}_t^l)$ in the robotic endoscope case. The center axis of the cylinder in the camera frame can be described as

$$\mathbf{p}_a^c = \mathbf{p}_0^c(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t) + \lambda \mathbf{d}^c(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t) \tag{47}$$

where $\lambda \in \mathbb{R}$. The cross section of a cylinder that is normal to the center axis can be described as the intersection between the surface of a sphere with radius r centered along $\mathbf{p}_a^c$ and a plane with normal $\mathbf{d}^c$ that contains the point $\mathbf{p}_a^c$. This intersection is described as

$$\begin{cases}(\mathbf{p}_c^c - \mathbf{p}_a^c)^\top (\mathbf{p}_c^c - \mathbf{p}_a^c) - r^2 = 0 \\ \mathbf{d}^c(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t)^\top (\mathbf{p}_c^c - \mathbf{p}_a^c) = 0\end{cases} \tag{48}$$

where $\mathbf{p}_c^c \in \mathbb{R}^3$ is a point on the perimeter of the circle from the cross section of the cylinder in the camera frame.

By combining (47) and (48), an expression for the surface of a cylinder can be derived. The resulting expression of the cylinder is

$$\begin{aligned}&(\mathbf{p}_s^c - \mathbf{p}_0^c(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t))^\top (\mathbf{p}_s^c - \mathbf{p}_0^c(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t)) \\ &- \left(\mathbf{d}^c(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t)^\top (\mathbf{p}_s^c - \mathbf{p}_0^c(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t))\right)^2 - r^2 = 0\end{aligned} \tag{49}$$

where $\mathbf{p}_s^c = \begin{bmatrix}x_s^c & y_s^c & z_s^c\end{bmatrix}^\top$ is a point on the surface of the cylinder in the camera frame.

Without loss of generality, let (X, Y) be the projected pixel coordinates of the cylinder to a unit camera using the pin-hole model. This can be converted to the (u, v) pixel location on a different camera by setting

$$X = \frac{u - c_u}{f_x} \qquad Y = \frac{v - c_v}{f_y} \tag{50}$$

where f_x, f_y, and c_u, c_v are the focal lengths and principal point in pixel units, respectively, from the camera intrinsic matrix $\mathbf{K}$.

Applying the camera pin-hole model to the surface of the cylinder in the camera frame results in a quadratic

$$A + \frac{1}{z}B + \frac{1}{z^2}C = 0 \tag{51}$$

where

$$\begin{aligned}A &= X^2 + Y^2 + 1 - (a^c X + b^c Y + c^c)^2 \\ B &= -2\left((x_0^c - a^c\nu)X + (y_0^c - b^c\nu)Y + z_0^c - c^c\nu\right) \\ C &= (x_0^c)^2 + (y_0^c)^2 + (z_0^c)^2 - \nu^2 - r^2\end{aligned} \tag{52}$$

and

$$\nu = a^c x_0^c + b^c y_0^c + c^c z_0^c. \tag{53}$$

The quadratic expression with respect to the depth occurs because there can be at most two solutions to the depth for each (X, Y) when the cylinder is projected onto an image plane. One solution is the visible side of the cylinder, and the other is the obstructed side of the cylinder. The case of a single solution to depth would occur only at the two edges of the projected cylinder. This can be enforced by setting the determinant of the quadratic to zero ($B^2 - 4AC = 0$) resulting in

$$\begin{aligned}&\left(\frac{Br}{-2\sqrt{C}} - \alpha X - \beta Y - \kappa\right)\left(\frac{Br}{-2\sqrt{C}} + \alpha X + \beta Y + \kappa\right) \\ &= 0\end{aligned} \tag{54}$$

where

$$\alpha = c^c y_0^c - b^c z_0^c \qquad \beta = a^c z_0^c - c^c x_0^c \qquad \kappa = b^c x_0^c - a^c y_0^c \tag{55}$$

after simplification.

Therefore, it is evident that the two edges from the projection of the cylinder result in two lines

$$\begin{aligned}&\left(\frac{r(x_0^c - a^c\nu)}{\sqrt{C}} - \alpha\right)X + \left(\frac{r(y_0^c - b^c\nu)}{\sqrt{C}} - \beta\right)Y \\ &\qquad + \left(\frac{r(z_0^c - c^c\nu)}{\sqrt{C}} - \kappa\right) = 0\end{aligned} \tag{56}$$

and

$$\begin{aligned}&\left(\frac{r(x_0^c - a^c\nu)}{\sqrt{C}} + \alpha\right)X + \left(\frac{r(y_0^c - b^c\nu)}{\sqrt{C}} + \beta\right)Y \\ &\qquad + \left(\frac{r(z_0^c - c^c\nu)}{\sqrt{C}} + \kappa\right) = 0.\end{aligned} \tag{57}$$

Through simple arithmetic and (50), both of these edges can be converted to the normal form described in (29) for any camera. In the normal form, let the resulting two edges be parameterized by $(\hat{\rho}_1(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t), \hat{\phi}_1(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t))$ and $(\hat{\rho}_2(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t), \hat{\phi}_2(\mathbf{w}_t, \mathbf{b}_t, \mathbf{e}_t))$.

D. Da Vinci Simulation Details

The stereoscopic endoscope's virtual cameras are set to render 540 by 432 images with a field of view of 60°. The baseline

TABLE III
PARAMETERS USED FOR SIMULATED EXPERIMENTS. ALL ANGLES AND DISTANCES ARE IN RADIANS AND MILLIMETERS, RESPECTIVELY

Parameter	Value
$\mathbf{\Sigma}^{b-}_{\mathbf{w},\mathbf{b}}$	$\text{diag}\begin{pmatrix}[0.005 & 0.005 & 0.005 \\ 5 & 5 & 5]\end{pmatrix}$
$\mathbf{a}^b_e$	$[0.004 \quad 0.004 \quad 2 \quad 0.004 \\ 0.004 \quad 0.004 \quad 0.01]$
$\mathbf{e}_c$	$[0.02 \quad 0.02 \quad 0.0025 \quad 0.02 \\ 0.02 \quad 0.02 \quad 0.05]$
$\mathbf{a}^c_e$	$[0.004 \quad 0.004 \quad 2 \quad 0.004]$
$\sigma_{c,l}$	$[0.0075 \quad 0.0075 \quad 0.75 \quad 0.0075]$

distance for the stereo cameras is set to 5 mm to give similar depth challenges in real stereoscopic endoscopes. The random walk for the orientation, $\mathbf{w}^e_t$ a quaternion vector, of the end-effector is

$$\mathbf{w}^e_{t+1} = \mathbf{w}^e_t \mathbf{w}^n_t \tag{58}$$

where $\mathbf{w}^n_t$ is the quaternion representation of axis-angle vector whose angle is sampled from $\mathcal{U}(0, 0.07)$ radians and axis is uniformly sampled in spherical coordinates

$$\begin{bmatrix} \sin(\phi^n_t)\cos(\theta^n_t) \\ \sin(\phi^n_t)\sin(\theta^n_t) \\ \cos(\phi^n_t) \end{bmatrix} \tag{59}$$

where $\theta^n_t = \arccos(u_t)$, $u_t \sim \mathcal{U}(-1, 1)$, and $\phi^n_t \sim \mathcal{U}(0, 2\pi)$. The trajectory per trial is ran for 140 time steps. Additional parameters are given in Table III.

ACKNOWLEDGMENT

The authors would like to thank Intuitive Surgical Inc. for instrument donations, and S. DiMiao, O. Maherari, D. Bergman, and A. Deguet for their support with the dVRK.

REFERENCES

[1] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.

[2] I. Fassi and G. Legnani, "Hand to sensor calibration: A geometrical interpretation of the matrix equation ax= xb," *J. Robot. Syst.*, vol. 22, no. 9, pp. 497–506, 2005.

[3] R. Y. Tsai, and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, Jun. 1989.

[4] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, no. 4/5, pp. 421–436, 2018.

[5] J. Mahler *et al.*, "Learning ambidextrous robot grasping policies," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaau4984.

[6] V. Pradeep, K. Konolige, and E. Berger, "Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach," in *Experimental. Robotics*. New York, NY, USA: Springer, 2014, pp. 211–225.

[7] M. Miyasaka, M. Haghighipanah, Y. Li, J. Matheson, A. Lewis, and B. Hannaford, "Modeling cable driven robot with hysteresis and cable-pulley network friction," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 2, pp. 1095–1104, Apr. 2020.

[8] M. Hwang *et al.*, "Efficiently Calibrating Cable-Driven Surgical Robots With RGBD Fiducial Sensing and Recurrent Neural Networks," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5937–5944, 2020, *arXiv:2003.08520*.

[9] U. Hagn *et al.*, "DLR Mirosurge: A versatile system for research in endoscopic telesurgery," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 5, no. 2, pp. 183–193, 2010.

[10] S. DiMaio, M. Hanuschik, and U. Kreaden, "The da vinci surgical system," in *Surgical Robotics*.New York, NY, USA: Springer, 2011, pp. 199–217.

[11] *Titan Medical*, "Sport surgical system,"

[12] 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 *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 6434–6439.

[13] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 2, pp. 590–596.

[14] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3400–3407.

[15] B. Benligiray, C. Topal, and C. Akinlar, "Stag: A stable fiducial marker system," *Image Vis. Comput.*, vol. 89, pp. 158–169, 2019.

[16] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving ax= xb on the euclidean group," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 717–721, Oct. 1994.

[17] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, 2009, Art. no. 155.

[18] J. Lambrecht and L. Kästner, "Towards the usage of synthetic data for marker-less pose estimation of articulated robots in RGB images," in *Proc. 19th Int. Conf. Adv. Robot.*, 2019, pp. 240–247.

[19] J. Lambrecht, "Robust few-shot pose estimation of articulated robots using monocular cameras and deep-learning-based keypoint detection," in *Proc. 7th Int. Conf. Robot Intell. Technol. Appl.*, 2019, pp. 136–141.

[20] T. E. Lee *et al.*, "Camera-to-robot pose estimation from a single image," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 9426–9432.

[21] J. Lu, F. Richter, and M. Yip, "Robust keypoint detection and pose estimation of robot manipulators with self-occlusions via sim-to-real transfer," 2020, *arXiv:2010.08054*.

[22] F. Zhong, Z. Wang, W. Chen, K. He, Y. Wang, and Y.-H. Liu, "Hand-eye calibration of surgical instrument for robotic surgery using interactive manipulation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1540–1547, Apr. 2020.

[23] T. Zhao, W. Zhao, B. D. Hoffman, W. C. Nowlin, and H. Hui, "Efficient vision and kinematic data fusion for robotic surgical instruments and other applications," Mar. 3, 2015, U.S. Patent 8 971 597.

[24] Y. Li *et al.*, "Super: A surgical perception framework for endoscopic tissue manipulation with surgical robotics," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2294–2301, Apr. 2020.

[25] A. Reiter, P. K. Allen, and T. Zhao, "Feature classification for tracking articulated surgical tools," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2012, pp. 592–600.

[26] A. Reiter, P. K. Allen, and T. Zhao, "Appearance learning for 3D tracking of robotic surgical tools," *Int. J. Robot. Res.*, vol. 33, no. 2, pp. 342–356, 2014.

[27] J. Lu, A. Jayakumari, F. Richter, Y. Li, and M. C. Yip, "Super deep: A surgical perception framework for robotic tissue manipulation using deep learning for feature extraction," 2020, *arXiv:2003.03472*.

[28] R. Hao, O. Özgüner, and M. C. Çavuşoğlu, "Vision-based surgical tool pose estimation for the da vinci robotic surgical system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1298–1305.

[29] M. Ye, L. Zhang, S. Giannarou, and G.-Z. Yang, "Real-time 3D tracking of articulated tools for robotic surgery," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2016, pp. 386–394.

[30] P. Pastor *et al.*, "Learning task error models for manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 2612–2618.

[31] F. Wang, K. Chen, and X. Chen, "An online calibration method for manipulator with joint clearance," *Robot*, vol. 35, pp. 521–526, 2013.

[32] H. Peng, X. Yang, Y.-H. Su, and B. Hannaford, "Real-time data driven precision estimator for RAVEN-II Surgical Robot end effector position," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 350–356.

[33] J. Mahler *et al.*, "Learning accurate kinematic control of cable-driven surgical robots using data cleaning and Gaussian process regression," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, 2014, pp. 532–539.

[34] M. Haghighipanah, M. Miyasaka, Y. Li, and B. Hannaford, "Unscented kalman filter and 3D vision to improve cable driven surgical robot joint angle estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 4135–4142.

[35] D. Seita, S. Krishnan, R. Fox, S. McKinley, J. Canny, and K. Goldberg, "Fast and reliable autonomous surgical debridement with cable-driven robots using a two-phase calibration procedure," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 6651–6658.

[36] Q. V. Le and A. Y. Ng, "Joint calibration of multiple sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 3651–3658.

[37] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in hand model acquisition," in *Proc., IEEE Int. Conf. Robots Autom.*, 2010, pp. 1817–1824.

[38] C. G. Cifuentes, J. Issac, M. Wüthrich, S. Schaal, and J. Bohg, "Probabilistic articulated real-time tracking for robot manipulation," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 577–584, Apr. 2017.

[39] Y. Shen, D. Sun, Y.-H. Liu, and K. Li, "Asymptotic trajectory tracking of manipulators using uncalibrated visual feedback," *IEEE/ASME Trans. Mechatronics*, vol. 8, no. 1, pp. 87–98, Mar. 2003.

[40] C.-C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, "Approximate jacobian control for robots with uncertain kinematics and dynamics," *IEEE Trans. Robot. Autom.*, vol. 19, no. 4, pp. 692–702, Aug. 2003.

[41] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax= xb," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 16–29, Feb. 1989.

[42] Z. Zhang, L. Zhang, and G.-Z. Yang, "A computationally efficient method for hand-eye calibration," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 10, pp. 1775–1787, 2017.

[43] K. Pachtrachai, M. Allan, V. Pawar, S. Hailes, and D. Stoyanov, "Hand-eye calibration for robotic assisted minimally invasive surgery without a calibration object," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 2485–2491.

[44] K. Pachtrachai *et al.*, "Adjoint transformation algorithm for hand-eye calibration with applications in robotic assisted surgery," *Ann. Biomed. Eng.*, vol. 46, no. 10, pp. 1606–1620, 2018.

[45] K. Pachtrachai, F. Vasconcelos, G. Dwyer, S. Hailes, and D. Stoyanov, "Hand-eye calibration with a remote centre of motion," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3121–3128, Oct. 2019.

[46] Z. Wang *et al.*, "Vision-based calibration of dual RCM-based robot arms in human-robot collaborative minimally invasive surgery," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 672–679, Apr. 2018.

[47] J. A. Piepmeier and H. Lipkin, "Uncalibrated eye-in-hand visual servoing," *Int. J. Robot. Res.*, vol. 22, no. 10/11, pp. 805–819, 2003.

[48] M. Verghese, F. Richter, A. Gunn, P. Weissbrod, and M. Yip, "Model-free visual control for continuum robot manipulators via orientation adaptation," 2019, *arXiv:1909.00450*.

[49] T. J. Rothenberg, "Identification in parametric models," *Econometrica*, vol. 39, no. 3, pp. 577–591, 1971.

[50] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 930–943, Aug. 2003.

[51] S. Thrun, "Particle filters in robotics," in *Proc. 18th Conf. Uncertainty Artif. Intell.*, 2002, pp. 511–518.

[52] C. Choi and H. I. Christensen, "Robust 3D visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features," *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 498–519, 2012.

[53] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic hough transform," *Comput. Vis. Image Understanding*, vol. 78, no. 1, pp. 119–137, 2000.

[54] F. Chaumette, La relation vision-commande: Théorie et application á des tâches robotiques, Ph.D. dissertation, Dept. Comput. Sci., L'Université de Rennes I., Rennes, France, 1990.

[55] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1321–1326.

[56] G. A. Fontanelli*et al.*, "A V-Rep simulator for the da vinci research kit robotic platform," in *Proc. 7th IEEE Int. Conf. Biomed. Robot. Biomechatronics*, 2018, pp. 1056–1061.

[57] G. Bradski, "The OpenCV library," *Dr Dobb's J. Softw. Tools*, 2000.

[58] J. Canny, "A computational approach to edge detection," *Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 679–698, 1986.

[59] A. Dutta, A. Gupta, and A. Zissermann, "VGG Image Annotator (VIA)" (2016). Accessed: Jun. 30, 2018. [Online]. Available: http://www.robots.ox.ac.uk / vgg/software/via/

[60] F. Richter, Y. Zhang, Y. Zhi, R. K. Orosco, and M. C. Yip, "Augmented reality predictive displays to help mitigate the effects of delayed telesurgery," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 444–450.

[61] A. Mathis *et al.*, "DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning," *Nat. Neurosci.*, vol. 21, no. 9, pp. 1281–1289, 2018.

[62] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.

[63] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611. Bellingham, WA, USA: SPIE, 1992, pp. 586–606.

[64] F. Richter *et al.*, "Autonomous robotic suction to clear the surgical field for hemostasis using image-based blood flow detection," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1383–1390, Apr. 2021.

[65] Z.-Y. Chiu, F. Richter, E. K. Funk, R. K. Orosco, and M. C. Yip, "Bimanual regrasping for suture needles using reinforcement learning for rapid motion planning," 2020, *arXiv:2011.04813*.

[66] K. Okamura and F. C. Park, "Kinematic calibration using the product of exponentials formula," *Robotica*, vol. 14, no. 4, pp. 415–421, 1996.

[67] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[68] S. Cremer, L. Mastromoro, and D. O. Popa, "On the performance of the Baxter research robot," in *Proc. IEEE Int. Symp. Assem. Manuf.*, 2016, pp. 106–111.

Florian Richter (Student Member, IEEE) received the B.Sc. degree in electrical engineering in 2017 from the University of Illinois at Chicago (UIC), Chicago, IL, USA and the M.S. degree in electrical engineering in 2020 from the University of California San Diego (UCSD), La Jolla, CA, USA. He is currently working toward the Ph.D. degree in electrical engineering with the Department of Electrical and Computer Engineering, UCSD.

His research interests include surgical robotics, robot perception, and snake-like robots.

Mr. Richter is a Fellow in the National Science Foundation Graduate Research Fellowship Program since 2017. He was the recipient of the UIC Bell Honors Award in 2017.

Jingpei Lu received the B.Sc. and M.S. degrees in electrical engineering in 2018 and 2020, respectively, from the University of California San Diego, La Jolla, CA, USA, where he is currently working toward the Ph.D. degree in electrical engineering with the Department of Electrical and Computer Engineering.

His research interests include computer vision and machine learning for robot perception and automation.

Ryan K. Orosco (Member, IEEE) received the B.Sc. degree in electrical engineering in 2006 from New Mexico State University, Las Cruces, NM, USA, and the M.D. degree from Johns Hopkins University, Baltimore, MD, USA, in 2010.

He is an Assistant Professor of Surgery with the University of California San Diego, La Jolla, CA, USA, and is a practicing Otolaryngologist with subspecialty training in head and neck robotic surgery. His research interests in robotic surgery include tissue modeling, novel techniques and platforms, safety, and semiautonomy.

Michael C. Yip (Senior Member, IEEE) received the B.Sc. degree in mechatronics engineering from the University of Waterloo,Waterloo, ON, Canada, in 2009, the M.S. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 2011, and the Ph.D. degree in bioengineering from Stanford University, Stanford, CA, USA, in 2015.

He is currently an Assistant Professor of Electrical and Computer Engineering, Director of the Advanced Robotics and Controls Lab with the University of California San Diego (UCSD), La Jolla, CA, USA, and the Director of the Medical Robotics Collaboratory with the UCSD Contextual Robotics Institute. His research interests include robotic surgery, machine learning-based control, haptics, soft robotics, and computer vision.

Dr. Yip was the recipient of several best paper awards at the International Conference on Robotics and Automation, including the Inaugural Best Paper Award for the IEEE Robotics and Automation Letters in 2016. He is an IEEE Robotics and Automation Society Distinguished Lecturer, NSF Career Award and NIH Trailblazer award recipient, and a Hellman Fellow.