MOPS

A Modular and Open Platform for Surgical Robotics Research

Schwaner, Kim Lindberg; Iturrate, Iñigo; Andersen, Jakob Kristian Holm; Dam, Christian Rosendahl; Jensen, Pernille Tine; Savarimuthu, Thiusius Rajeeth

Published in:
2021 International Symposium on Medical Robotics (ISMR)

DOI:
10.1109/ISMR48346.2021.9661539

Publication date:
2021

Document version:
Accepted manuscript

MOPS: A Modular and Open Platform for Surgical Robotics Research

Kim Lindberg Schwaner* Iñigo Iturrate* Jakob Kristian Holm Andersen*
Christian Rosendahl Dam* Pernille Tine Jensen† Thiusius Rajeeth Savarimuthu*

Abstract

MOPS is a platform for surgical robotics research. It consists of hardware and software components for building surgical robot systems based on existing robot arms. The reference system presented in this paper has two robot arms, each with an adapter for mounting and actuating different types of surgical instruments. Additionally, the system has a high-resolution stereo camera and a simple operator console with haptic devices and foot pedals for teleoperating the manipulators. Software components are based on Robot Operating System (ROS) to keep them modular and reusable for different types of hardware. We evaluate the precision of our system for vision-based tasks and demonstrate its potential for surgical task automation. The mean end-to-end error, including that of the vision system, for a trajectory following task was 1.2 mm / 0.3° for one manipulator and 2.7 mm / 0.7° for the other. MOPS software and hardware components are made available under an open-source license to encourage collaboration and facilitate the advancement of surgical robotics research.

1 Introduction

An increasing number of surgical procedures is performed with robotic minimally invasive technique. There is evidence that this paradigm shift has had clear patient and societal benefits [1]. Currently, robotic surgery is performed through teleoperation and quality is thus dependent of the surgeon's skills. We envision that future autonomous execution of robotic surgery will enable surgeons to plan and oversee one or several surgical procedures while the robot performs autonomously. This will make robotic surgery highly scalable, more consistent and increase the availability of advanced surgical procedures.

To date, experimental research on surgical task automation is carried out either in simulation only, using custom in-house robot systems, or with the da Vinci Research Kit (dVRK) [2] or Raven-II [3] platforms. Unfortunately, the availability of the dVRK is limited and using the Raven-II for autonomous operation is very challenging because of large uncertainties in state estimation.

*K. L. Schwaner, I. Iturrate, J. K. H. Andersen, C. R. Dam and T. R. Savarimuthu are with SDU Robotics, The Mærsk Mc-Kinney Møller Institute, Faculty of Engineering, University of Southern Denmark (SDU), 5230 Odense, Denmark; {kils,inju,jkha,trs}@mmmi.sdu.dk.

†P. T. Jensen is with the Department of Gynaecology and Obstetrics, Aarhus University Hospital, 8200 Aarhus, Denmark; petije@rm.dk.

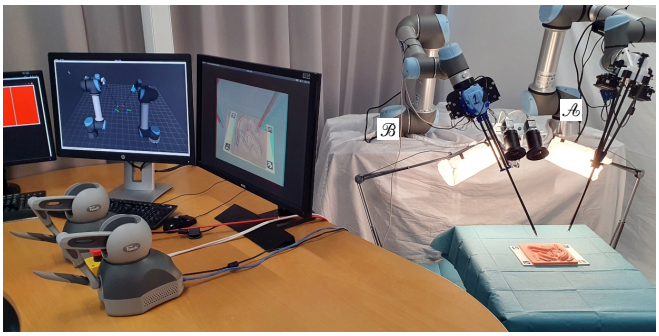


Figure 1: Our surgical robot research platform set up in the lab: The image shows the haptic devices and screens for teleoperating and monitoring the robot to the left, and the manipulators and workspace to the right.

In this paper, we present MOPS – an open platform for surgical robotics research with modular hardware and software components. The modular architecture allows using different types of robot arms, surgical instruments, sensors and master input devices interchangeably with relative ease. Our platform provides building blocks for setting up surgical robot systems based on commercially available robot arms which already exist in many research laboratories. We believe such a system will benefit future research in surgical robotics and, in particular, surgical task automation where high precision and good sensing capabilities are required.

For the reference system (see Fig. 1) described in this paper, we use UR5 and UR5e robot arms (Universal Robots A/S, Denmark) with custom instrument adapters for attaching da Vinci EndoWrist surgical instruments (Intuitive Surgical, Inc.) and a custom high-resolution stereo vision system. The UR5e robot has an integrated force-torque sensor at the tool flange, adding an additional sensing capability to the system. Recently, we have used this specific system for a study on autonomous suturing [4].

The software for our system is developed following a component-based design approach. Components are implemented and interconnected utilizing the Robot Operating System (ROS). Robot models are provided in Unified Robot Description Format (URDF). Changing a hardware component requires only changing the robot model and the corresponding software driver component.

To strengthen collaboration among researchers we have set up a project page[1] where we will make hardware designs and

[1] https://gitlab.com/sdurobotics/medical/mops.

source code for our platform available to the community under an open source license.

A wide variety of systems have been used for research into automation of surgical tasks. Some examples of in-house surgical robot systems are given in [5], [6]. Industrial robot manipulators have constituted the basis of a number of systems, which have been used for tasks like surgical knot tying [7] and suturing [8]–[10]. The most prevalent systems used for surgical robotics research are the dVRK [2] and Raven-II [3]. To the best of our knowledge, these are also the only *open* (in the sense of an open robot interface) systems.

The Raven-II is an open surgical robot platform which features two cable-driven 7 Degrees of Freedom (DOF) manipulators and open-source control software. It has been used for research into automation of surgical tasks like debridement [11], tumor ablation [12], human-robot collaborative myomectomy [13], motion compensation [14], soft tissue manipulation [15] and suturing [16]. However, task automation using the Raven-II is challenging because of elasticity, tensioning and friction of the cables used for actuating its joints. This causes end-effector state estimation errors in the range of 10 to 25 mm at the end-effector [17]–[19]. Approaches have been suggested for reducing trajectory tracking errors [18] and improving the state estimates [17], [19], [20]. However, incorporating such methods in the system increases complexity and overhead for set up and use.

The dVRK is an open and well-supported research platform based on first-generation da Vinci robot hardware. It has enabled a lot of research into automation of surgical tasks such as cutting [21], [22], suturing [23], [24], debridement [25] and training tasks like peg transfer [26], [27]. Other researchers have investigated the use of modified instruments to perform autonomous tumor localization and resection [28], [29]. The dVRK is, similar to the Raven-II, cable-driven and displays some of the same related issues, although they are much less pronounced: the state estimation error is in the range of 1 to 4 mm [25], [26], [30], [31]. However, the dissemination of the dVRK is limited by the availability of first-generation da Vinci robot hardware.

While our system is similar in scope to systems like the dVRK and Raven-II it offers benefits like high modularity and flexibility, potentially higher precision and the possibility of using additional sensors. Example robot arms that can be used with our platform are the UR5 and UR5e, LBR iiwa, LBR Med (KUKA) and Panda (Franka Emika GmbH). The listed robots all have actuators and encoders placed at the joints, avoiding some of the issues inherent to cable-driven systems. Accordingly, they have datasheet ISO 9283 pose repeatability of ± 0.1 mm for most, and ± 0.03 mm for the UR5e. Our system can be used either as an alternative to, or in conjunction with, other surgical robot systems for research.

2 Platform Hardware

This section gives an overview of the hardware of our reference system as it is currently set up in our laboratory (Fig. 1).

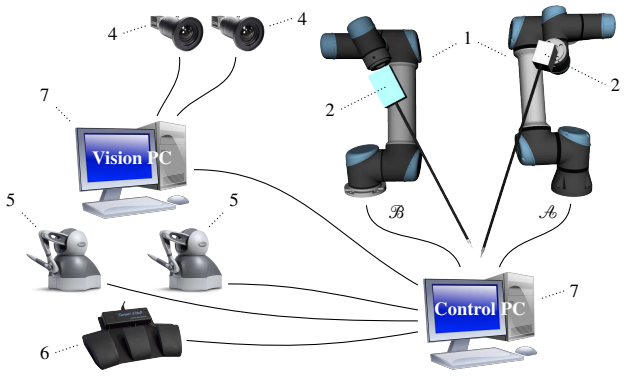


Figure 2: Overview of the hardware components of our reference surgical robot system.

An overview of the system is diagrammed in Fig. 2. The system comprises the following elements, each of which are exchangeable (numbers correspond to those in Fig. 2):

1) UR5 (CB-Series) and UR5e (e-Series) robot arms;
2) da Vinci EndoWrist S/Si and X/Xi surgical instruments;
3) adapters for mounting and actuating the instruments;
4) Basler acA2500-20gc GigE cameras with Tamron M111FM08 8 mm lenses;
5) 3DSystems Touch haptic devices;
6) Kinesis Savant Elite2 foot pedals; and
7) desktop PCs.

An interface with the workspace exists at the *manipulator site* and there is a human-machine interface at the *operator site*. The following will describe each in more detail.

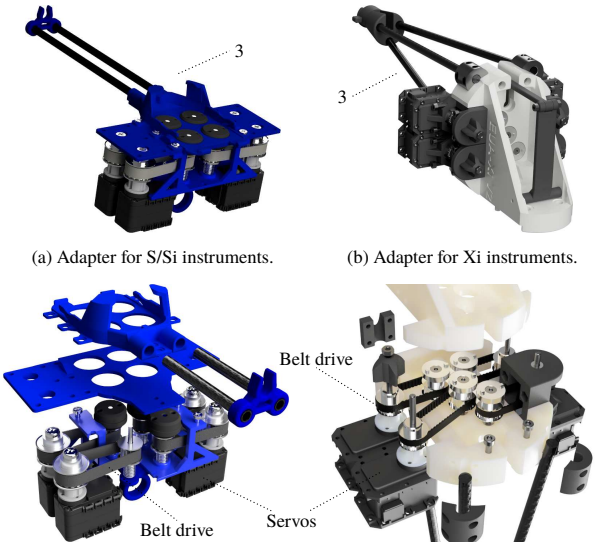
2.1 Manipulator Site

At the manipulator site, our system currently has two manipulators, each comprised of 1) a robot arm for positioning the surgical instruments; 2) an adapter for mounting and actuating the surgical instrument; and 3) the actual surgical instrument.

In our reference system, UR5 and UR5e robots act as positioning arms for surgical instruments. The URs are commercially available 6 DOF collaborative robots, i.e. they are approved for use in close proximity of humans. They are mounted rigidly to a stand which is elevated above the workspace to properly position the instrument end-effectors.

We use da Vinci EndoWrist surgical instruments to keep the system close to a realistic clinical setting. To mount the instruments on the positioning arms, we use a pair of custom-made adapters: one for da Vinci S/Si instruments and one for da Vinci Xi instruments. Fig. 3 shows renderings of the adapters, which are fastened to the tool flange of a positioning arm.

The instrument adapters consist of 3D-printed sections combined with off-the-shelf mechanical parts and actuators. We use Dynamixel AX-12A and MX-28T servo modules for actuating the instrument joints due to compactness, high torque, ease of use and torque feedback for sensorless homing. Each servo is coupled to a cylindrical disc with knobs for



(a) Adapter for S/Si instruments. (b) Adapter for Xi instruments.

(c) Exploded view of S/Si adapter. (d) Exploded view of Xi adapter.

Figure 3: Renderings of our custom-made adapters for actuating da Vinci EndoWrist surgical instruments and mounting them on the tool flange of a robot arm.

engaging with the instrument via toothed pulleys and belts. Rolled carbon fiber rods are used for stiff, lightweight support of the instrument shaft. The adapters are made to fit an ISO 9409-1-50-4-M6 conforming robot tool flange.

In our reference system, as shown in Figs. 1 and 2, we name the UR5e robot holding an Xi type instrument $\mathcal{A}$, and the UR5 robot holding an S/Si type instrument $\mathcal{B}$.

Most da Vinci instruments have each 4 DOF, bringing the total number of DOF for each manipulator to 10. In practice we combine the last two DOF (grasper jaws) of the instrument into a single "grasper opening angle" DOF, similar to the dVRK and Raven-II.

The vision system of our platform is based on a custom stereo camera rig with Basler acA2500-20gc GigE cameras and Tamron M111FM08 8 mm lenses. This configuration acts as a substitute for an actual stereo endoscope and allows focusing down to 30 cm and a 76° horizontal field of view with less than −1.9 % lens TV distortion. The stereo rig is rigidly attached to the stand on which the manipulators are also mounted, i.e. the cameras are fixed with respect to the robot bases. The desired workspace area was approximately 40 cm from the focal point of the cameras.

2.2 Operator Site

At the operator site, we have a simple console made up by 1) 3DSystems Touch haptic devices; 2) a set of foot pedals; 3) monitors; and 4) red/cyan glasses. The operator controls the robot via haptic devices and foot pedals. To engage teleoperation, the operator engages one of the pedals. Another pedal engages the free-drive mode of both UR robots, allowing them to be moved by hand.

We facilitate operator depth perception with a simple stereo viewer that displays a red/cyan anaglyph, composed from synchronized stereo frame pairs.

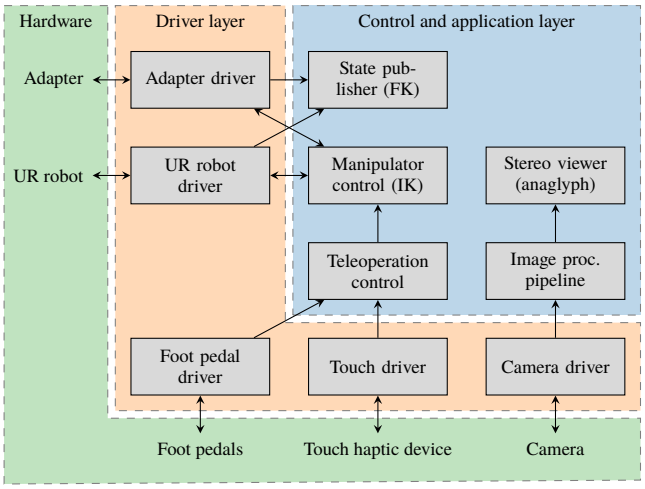


Figure 4: Overview of core software components of our surgical robot platform. Note that the diagram is simplified: in practice many components are instantiated multiple times.

3 Software Components

The software for our platform is developed with a component-based approach. Fig. 4 shows the core components of our system. We distinguish between two software layers: a *driver* layer and a *control and application* layer.

Components are distributed across two PCs which we denote the *control PC* and *vision PC*. The control PC runs high-rate haptic device, robot and instrument control loops, and reads input from foot pedals. The vision PC transfers frames from the stereo camera and runs a machine vision pipeline and visualization programs. All software components are implemented as ROS *nodes* except for vision-related components which are implemented as *nodelets* for zero-copy image data passing. Utilizing the ROS framework gives users the possibility to easily take advantage of the various tools of the ROS ecosystem, like RViz for 3D visualization and RQt (plugins) for interacting with the system in different ways.

Our software currently targets Ubuntu 20.04, ROS Noetic, C++17 and Python 3.8. For the case of the control PC, we use a `PREEMPT_RT`[2]-patched Linux kernel.

The software interfaces with a number of hardware units, described in Section 2, through driver components. In this paper, we will not go into detail with the specific drivers. Note, however, that each hardware device has a separate driver component associated with it. This means that if a hardware component is changed, only the one driver component associated with it will have to be changed, while the rest of the system can remain the same.

In the remainder of this section we will describe some software components that are distinctive to our platform in more

[2] `https://wiki.linuxfoundation.org/realtime/start`

detail.

3.1 Manipulator Control

The manipulators of our platform do not have mechanically constrained motion with respect to a Remote Center of Motion (RCM), as the da Vinci and Raven-II robots do. We use two different model-based schemes for solving the inverse kinematics problem and controlling the manipulators, depending on whether or not we wish motion to be constrained by an RCM. Common to both methods is the fact that we must resolve the redundancy of each $n = 9$ DOF manipulator with respect to the $m = 6$ DOF task space.

At the velocity level, the mapping between the joint velocities $\dot{\boldsymbol{q}}$ and task space velocity $\dot{\boldsymbol{x}}$ is

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{q}}, \tag{1}$$

where $\boldsymbol{J} = \partial\dot{\boldsymbol{x}}/\partial\dot{\boldsymbol{q}}$ is the $m \times n$ manipulator Jacobian. For the case where we do *not* want manipulator movement to be constrained by the RCM, we resolve redundancy by finding the Weighted Least Norm (WLN) solution [32] to the velocity kinematics (1), provided by the weighted pseudo inverse

$$\boldsymbol{J}^{\#} = \boldsymbol{W}^{-1}\boldsymbol{J}^{\mathsf{T}}\left(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^{\mathsf{T}}\right)^{-1}, \tag{2}$$

where $\boldsymbol{W}$ is a symmetric and positive definite $n \times n$ weighting matrix with which we can prioritize motion in the instrument joints over robot arm joints. The solution, which minimizes the Euclidean $\boldsymbol{W}$-weighted norm $\sqrt{\dot{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{W}\dot{\boldsymbol{q}}}$ is, for a desired task space velocity $\dot{\boldsymbol{\xi}}$, then given by

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}\dot{\boldsymbol{\xi}}. \tag{3}$$

For control we use the resolved-rate motion scheme and compute the desired end-effector velocity from the task space error

$$\dot{\boldsymbol{\xi}} = \boldsymbol{K}_p(\boldsymbol{x}_{\text{des}} - \boldsymbol{x}), \tag{4}$$

where $\boldsymbol{x}_{\text{des}}$ is the desired value and $\boldsymbol{K}_p$ is a symmetric and positive definite matrix.

The above WLN method will clearly not fulfill the RCM constraint. For that purpose we instead adopt the method of [33] to write the extended task velocity kinematics as

$$\begin{bmatrix}\dot{\boldsymbol{x}}\\ \dot{\boldsymbol{p}}_{\text{RCM}}\end{bmatrix} = \boldsymbol{J}_{\text{ext}}\begin{bmatrix}\dot{\boldsymbol{q}}\\ \dot{\kappa}\end{bmatrix}, \tag{5}$$

with the extended task Jacobian defined as

$$\boldsymbol{J}_{\text{ext}} = \begin{bmatrix}\boldsymbol{J} & \boldsymbol{0}_{m\times 1}\\ \boldsymbol{J}_i + \kappa(\boldsymbol{J}_{i+1} - \boldsymbol{J}_i) & \boldsymbol{p}_{i+1} - \boldsymbol{p}_i\end{bmatrix}, \tag{6}$$

where $\boldsymbol{p}_{\text{RCM}} = \boldsymbol{p}_i + \kappa(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i)$ is a point along the line with endpoints $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$ at the origin of joints i and $i+1$. The parameter $\kappa \in [0, 1]$ models the insertion depth.

To constrain the motion of $\boldsymbol{p}_{\text{RCM}}$ we set its velocity to zero, i.e. $\dot{\boldsymbol{p}}_{\text{RCM}} \overset{\text{set}}{=} \boldsymbol{0}_{3\times 1}$. Given a desired task and trocar location $\boldsymbol{p}_{\text{trocar}}$ we can then, similar to (4), compute the desired end-effector velocity from the task space error

$$\dot{\boldsymbol{\xi}}_{\text{ext}} = \begin{bmatrix}\boldsymbol{K}_p & \boldsymbol{0}_{m\times 3}\\ \boldsymbol{0}_{3\times m} & \boldsymbol{K}_{\text{RCM}}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_{\text{des}} - \boldsymbol{x}\\ \boldsymbol{p}_{\text{trocar}} - \boldsymbol{p}_{\text{RCM}}\end{bmatrix}, \tag{7}$$

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_{\text{RCM}}$ are symmetric and positive definite matrices, and finally derive the proportional control law

$$\begin{bmatrix}\dot{\boldsymbol{q}}\\ \dot{\kappa}\end{bmatrix} = \boldsymbol{J}_{\text{ext}}^{+}\,\dot{\boldsymbol{\xi}}_{\text{ext}}, \tag{8}$$

where $\boldsymbol{J}_{\text{ext}}^{+} = \boldsymbol{J}_{\text{ext}}^{\mathsf{T}}(\boldsymbol{J}_{\text{ext}}\boldsymbol{J}_{\text{ext}}^{\mathsf{T}})^{-1}$ is the Moore-Penrose inverse of the extended Jacobian.

During teleoperation, the value $\boldsymbol{x}_{\text{des}}$ is input from the haptic devices, but it may also be computer-generated, e.g. during autonomous task execution. The desired insertion point $\boldsymbol{p}_{\text{trocar}}$ is set manually.

The kinematic models used for computing the Jacobians and locations of the robot joints were generated in URDF and loaded into the software at runtime. The Jacobians and inverse kinematics are computed numerically and as such the implementation does not have to be changed if the robot model changes.

3.2 Stereo Pipeline and Viewer

Our machine vision pipeline begins with the camera driver components grabbing pairs of synchronized frames from the left and right cameras. In order to obtain frame rates of 25 Hz we grab only a 1280×800 pixel region, which still captures the desired workspace volume of approximately 20×20×20 cm, in each frame.

Internal camera clocks are synchronized via Precision Time Protocol (PTP) and the vision PC acts as the grandmaster clock. This allows for very accurately synchronized stereo frame pairs without the need for a hardware trigger.

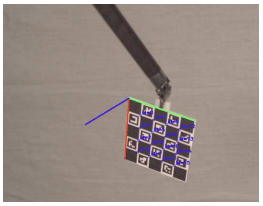
The camera driver plugs into the standard ROS image processing pipeline for calibration, image debayering and rectification. Rectified image pairs are used to construct an anaglyph for viewing with red/cyan glasses. This constitutes a very simple and easy to build viewer for our surgeon console. Anaglyph RGB values $\begin{bmatrix}r_a & g_a & b_a\end{bmatrix}^{\mathsf{T}}$ are composed as

$$\begin{bmatrix}r_a\\ g_a\\ b_a\end{bmatrix} = \begin{bmatrix}0.299 & 0.587 & 0.114\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}r_l\\ g_l\\ b_l\end{bmatrix} + \begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_r\\ g_r\\ b_r\end{bmatrix} \tag{9}$$

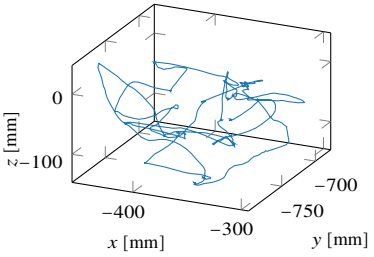
where $\begin{bmatrix}r_l & g_l & b_l\end{bmatrix}^{\mathsf{T}}$ are the RBG values of the left frame and $\begin{bmatrix}r_r & g_r & b_r\end{bmatrix}^{\mathsf{T}}$ are those of the right frame.

4 System Evaluation and Example Applications

In this section we demonstrate the feasibility of the proposed platform for surgical task automation with three brief experiments.

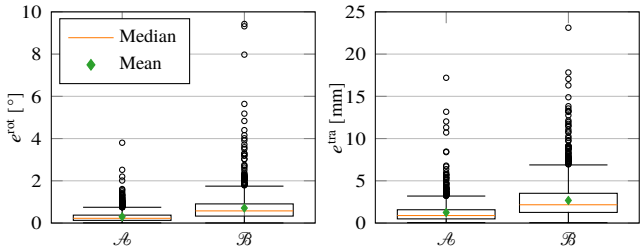


(a) Marker plate held rigidly in the instrument grasper jaws.



(b) End-effector trajectory.

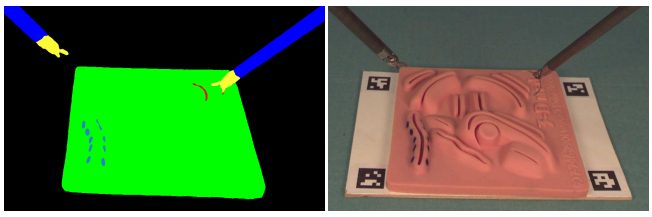
Figure 5: System evaluation by tracking a calibration plate during a trajectory following task.



(a) Rotational error.

(b) Translational error.

Figure 6: System error during a trajectory following task, evaluated based on camera sensor feedback (1346 samples).



(a) Predicted labels for scene objects.

(b) Autonomous needle pick-up.

Figure 7: Autonomous localization and pick-up of a suture needle.

4.1 System End-to-End Evaluation

In this first experiment we examined the end-to-end error of manipulators $\mathcal{A}$ and $\mathcal{B}$ and the stereo vision system during a trajectory following task. Lacking external ground-truth measurements, we evaluated whether the end-effector motion perceived by the camera agreed with the motion estimated from the joint encoders, i.e. forward kinematics. To that end we used a 4×4 cm planar ChArUco marker which was held rigidly with a Large Needle Driver (LND) instrument, as shown in Fig. 5a.

We performed hand-eye calibration using the formulation by [34] with 65 and 71 measurement pairs $\left\{ {}^{\text{tcp}}_{\text{base}}\boldsymbol{T}_i, {}^{\text{mark}}_{\text{cam}}\boldsymbol{T}_i \right\}$, where the homogeneous transformation ${}^{\text{tcp}}_{\text{base}}\boldsymbol{T}_i$ is the end-effector frame seen with respect to the robot base frame at time instant i, finding ${}^{\text{cam}}_{\text{base}}\boldsymbol{T}$ for each manipulator respectively.

Next, we in turn moved each end-effector along the trajectory displayed in Fig. 5b by teleoperation, recording 1346 pose pairs for each. End-effector rotation was varied as much as possible while keeping the marker in view of the camera. Between each time instant i and the next we computed the motion transforms

$$\boldsymbol{A}_i = {}^{\text{tcp}}_{\text{base}}\boldsymbol{T}_i \left[{}^{\text{tcp}}_{\text{base}}\boldsymbol{T}_{i+1} \right]^{-1} \tag{10}$$

and

$$\boldsymbol{B}_i = {}^{\text{mark}}_{\text{cam}}\boldsymbol{T}_i \left[{}^{\text{mark}}_{\text{cam}}\boldsymbol{T}_{i+1} \right]^{-1} \tag{11}$$

and finally residual error transforms between motion estimated from robot kinematics and motion estimated from the camera

$$\boldsymbol{D}_i = \left(\boldsymbol{A}_i \, {}^{\text{cam}}_{\text{base}}\boldsymbol{T} \right)^{-1} \left({}^{\text{cam}}_{\text{base}}\boldsymbol{T} \, \boldsymbol{B}_i \right) \tag{12}$$

for $i = 1, \ldots, n-1$. Let now $\theta_i \boldsymbol{v}_i$ be the equivalent axis-angle representation of the 3×3 rotational part of $\boldsymbol{D}_i$, and $\boldsymbol{t}_i$ the 3×1 translation vector of $\boldsymbol{D}_i$. We then define the rotational and translational error measures as

$$e_i^{\text{rot}}\left(\boldsymbol{D}_i\right) = |\theta_i| \tag{13}$$

$$e_i^{\text{tra}}\left(\boldsymbol{D}_i\right) = \|\boldsymbol{t}_i\|_2 \tag{14}$$

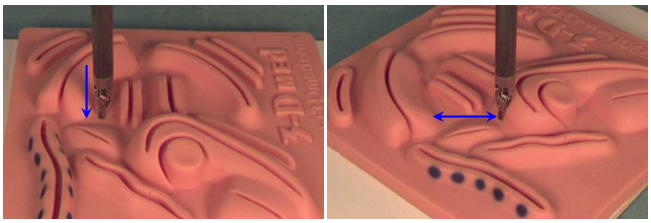
We obtained mean error values of 1.2 mm and 0.3° for manipulator $\mathcal{A}$, and 2.7 mm and 0.7° for $\mathcal{B}$. Fig. 6 plots the error values and visualizes the uncertainty.

4.2 Vision-Based Suture Needle Localization and Pick-up

In this experiment, we demonstrate an autonomous suture needle grasping and pick-up, utilizing the stereo vision system. To find image pixels belonging to the suture needle we employed a modified U-net [35]. The network was trained on a set of 298 laboratory images with semantic labels for a phantom tissue pad, surgical instruments and suturing needle. The network was trained for multi-class segmentation on patches sampled from training data to optimize a combo loss function similar to [36], consisting of the categorical crossentropy loss and Focal Tversky loss [37] to improve segmentation of smaller objects such as a needle. For needle segmentation, the network has an intersection over union score of 0.76 at a framerate of 36.6, using an NVIDIA GeForce RTX 2080 with 11 GB RAM. An example segmentation result is presented in Fig. 7a.

We used the needle segmentation masks of the left and right rectified images to find stereo correspondences and then triangulated to find 3D points. Then, we fit the 3D points to a model of the half-circle needle, using the Iterative Closest Point (ICP) algorithm, to finally determine the 6 DOF pose of the needle.

Using this method for needle pose estimation we conducted an experiment where a single manipulator was used to pick up the needle from an unknown location in 30 trials. An example pick-up is shown in Fig. 7b. The initial position of the needle was varied within a volume of approximately 5×5×1 cm throughout 30 trials, while the initial orientation was kept constant. The grasp pose was determined manually and was kept constant. The needle pose estimation, grasping and pick-up was successful in 28 of 30 trials.



(a) Downward contact detection. (b) Sideways contact detection.

Figure 8: Snapshots at the time of contact detection between the instrument tip and phantom tissue, based on force measurements of the sensor in the UR5e robot flange.

	Penetration [mm]	Force [N]
Contact with tool tip	3.08 ± 0.46	1.34 ± 0.11
Contact with side (right)	3.65 ± 0.25	1.41 ± 0.21
Contact with side (left)	2.42 ± 0.56	1.23 ± 0.20
Mean	3.05 ± 0.67	1.33 ± 0.19

Table 1: Penetration depth and contact force (mean $\pm$ std. dev.) when contacting against soft phantom tissue for 20 trials of each type of contact.

4.3 Contact Detection Using Force Feedback

One of the inherent advantages of using a UR5e as part of our surgical robot platform is the availability of force-torque sensor measurements at the robot tool flange. Its potential use is demonstrated with the following experiment, where we detected contact between the tip of a LND instrument and soft phantom tissue. We propose using a modified non-restarting version of the CUSUM algorithm with an upper bound on the threshold [38] to detect changes that indicate contact. The algorithm was applied on the magnitude of the end-effector forces, expressed in the tool flange frame.

Two scenarios, depicted in Fig. 8, were examined. In the first, the instrument shaft and jaws of the end-effector were placed perpendicular to the phantom surface. The end-effector was then commanded to move towards the phantom until contact was detected, after which it immediately stopped. In the second scenario, the instrument was moved sideways, parallel to the phantom surface, until contact with raised parts of the phantom was detected. After contact detection, the direction of movement was reversed until the next contact was made. 20 trials were completed for each scenario. The force-torque sensor was zeroed initially for the sideways scenario, and after moving to the uppermost point in the up-down contact scenario.

To evaluate the responsiveness of the contact detection scheme, we measured the penetration depth into the tissue, at the time contact was detected, and the magnitude of the contact force. The ground-truth values for penetration depth were estimated by manually positioning the instrument exactly at the phantom surface five times, reading the end-effector positions, determined from forward kinematics, and calculating the mean value.

The results of the experiment are listed in Table 1. Across all experiments, the mean penetration depth at time of contact detection was 3.05 mm and the mean force measured was 1.33 N. Contact detection was successful in all trials.

5 Discussion

As we have noted, a number of different robot systems have been used for automation in surgical robotics research. Only the dVRK and Raven-II have been made available to the research community under open source licenses. However, the availability of the dVRK is limited and the Raven-II is not very well suited to automation.

In this paper we propose a modular and open platform which can be built from different robot arms. Our reference system is built from UR5(e) robots and custom adapters for surgical instruments. Similar systems can be built from other robot hardware, as long as the positioning robot has 6 DOF, suitable reach and can carry a payload of at least 2 kg.

The dVRK and Raven-II feature passive spherical and double-parallelogram RCM mechanisms, which simplify the control algorithms (they can control in the joint space without violating the RCM constraint) and can act as an extra safety feature. Achieving the RCM actively with a software controller allows the use of other types of robot arms, which do not use cable drives for joint actuation and are potentially more precise. Another benefit to this approach is that the location of the RCM may be changed programmatically.

The experiment in Section 4.1 shows mean errors of 1.2 mm / 0.3° and 2.7 mm / 0.7° for manipulators $\mathscr{A}$ and $\mathscr{B}$ respectively. From Fig. 6 it is clear that, although the large majority of error values are low, there are also a number of very large error values, up to almost 10 mm for $\mathscr{A}$ and 25 mm for $\mathscr{B}$. These were caused by errors of the vision-based detection of the marker. Manipulator $\mathscr{A}$ shows better performance than $\mathscr{B}$. This is likely because $\mathscr{B}$ is made from the UR5 robot and AX-12A servos, both of which have worse accuracy than the UR5e and MX-28T servos used for $\mathscr{A}$. Although this experiment does not represent a rigorous evaluation of the true precision of our system, it at least provides a bound on errors associated with autonomous tasks that use visual feedback for locating scene objects.

Autonomous suture needle pick-up, using the vision system for needle pose estimation, was successful in 28 of 30 attempts. The two times it failed were due to erroneous pose estimation. This result demonstrates the feasibility of our system for needle manipulation tasks and encourages further study of automated surgical suturing.

Using commercially available robot arms allows placing additional sensors between the robot arm flange and surgical instrument or using sensors built into the robot arm. This can be used as supplementary sensor input for closed-loop control, which we demonstrate in Section 4.3. Here, contact was detected at a mean penetration depth of around 3 mm at a measured force of 1.3 N. Contact detection may, for example, be used to determine the location or mechanical properties of parts of the anatomy or as a safety feature.

The rigid mounting of robot and cameras has the advantage of making calibrating them to the robot easier and a one-time endeavor. Our setup also allows using types of cameras which are not usually available in a clinical setting. We argue that the same principles and methods can be applied to a stereo endoscope, even though the small baseline imposes some practical limitations.

Our operator console, and in particular the red/cyan anaglyph viewer, works well for performing small tasks, but is foreign to surgeons who are used to the da Vinci console. This is definitely a disadvantage in case the system needs to be used by a surgeon. However, there is nothing preventing the use of the da Vinci master console (via the dVRK), or another human-robot interface, with our system.

With the contribution of this paper, and by making the hardware designs and software for our platform available under an open source license, we hope to further facilitate research in the field of surgical robotics. Currently, researchers at the Jožef Stefan Institute (JSI), Slovenia, are setting up a MOPS-based system using KUKA robot arms.

At University of Southern Denmark (SDU), the platform we propose in this paper is currently being used for research on surgical task automation and computer vision algorithms for navigating the surgical scene. Future work on the platform itself includes collision detection and prevention, calibration improvement and integration with additional sensors. We additionally plan to adopt ROS2 in the near future.

Acknowledgment

We thank Michael K. Schmidt (SDU) for his assistance with setting up our robot system, Zhuoqi Cheng (SDU) for manufacturing the calibration plate and Miha Deniša (JSI) for his work on robot-instrument calibration.

References

[1] S. L. Jørgensen, O. Mogensen, C. Wu, K. Lund, M. Iachina, M. Korsholm, and P. T. Jensen, "Nationwide introduction of minimally invasive robotic surgery for early-stage endometrial cancer and its association with severe complications," *JAMA Surgery*, vol. 154, no. 6, p. 530, Jun. 2019. DOI: 10.1001/jamasurg.2018.5840.

[2] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci® surgical system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2014, pp. 6434–6439. DOI: 10.1109/ICRA.2014.6907809.

[3] B. Hannaford, J. Rosen, D. W. Friedman, H. King, P. Roan, L. Cheng, D. Glozman, J. Ma, S. N. Kosari, and L. White, "Raven-ii: An open platform for surgical robotics research," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 954–959, Apr. 2013. DOI: 10.1109/TBME.2012.2228858.

[4] K. L. Schwaner, I. Iturrate, J. K. H. Andersen, P. T. Jensen, and T. R. Savarimuthu, "Autonomous bi-manual surgical suturing based on skills learned from demonstration," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, to be published, Prague, Czech Republic: IEEE, Sep. 2021.

[5] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2010. DOI: 10.1109/ROBOT.2010.5509621.

[6] T. Osa, N. Sugita, and M. Mitsuishi, "Online trajectory planning and force control for automation of surgical tasks," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 675–691, Apr. 2018. DOI: 10.1109/TASE.2017.2676018.

[7] H. Mayer, I. Nagy, A. Knoll, E. Braun, R. Lange, and R. Bauernschmitt, "Adaptive control for human-robot skilltransfer: Trajectory planning based on fluid dynamics," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Roma: IEEE, Apr. 2007, pp. 1800–1807. DOI: 10.1109/ROBOT.2007.363583.

[8] C. Staub, T. Osa, A. Knoll, and R. Bauernschmitt, "Automation of tissue piercing using circular needles and vision guidance for computer aided laparoscopic surgery," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2010, pp. 4585–4590. DOI: 10.1109/ROBOT.2010.5509601.

[9] S. Leonard, K. L. Wu, Y. Kim, A. Krieger, and P. C. W. Kim, "Smart tissue anastomosis robot (star): A vision-guided robotics system for laparoscopic suturing," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1305–1317, Apr. 2014. DOI: 10.1109/TBME.2014.2302385.

[10] H. Saeidi, H. N. D. Le, J. D. Opfermann, S. Leonard, A. Kim, M. H. Hsieh, J. U. Kang, and A. Krieger, "Autonomous laparoscopic robotic suturing with a novel actuated suturing tool and 3d endoscope," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, May 2019. DOI: 10.1109/icra.2019.8794306.

[11] B. Kehoe, G. Kahn, J. Mahler, J. Kim, A. Lee, A. Lee, K. Nakagawa, S. Patil, W. D. Boyd, P. Abbeel, and K. Goldberg, "Autonomous multilateral debridement with the raven surgical robot," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1432–1439. DOI: 10.1109/ICRA.2014.6907040.

[12] D. Hu, Y. Gong, B. Hannaford, and E. J. Seibel, "Semi-autonomous simulated brain tumor ablation with ravenii surgical robot using behavior tree," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 3868–3875, 2015. DOI: 10.1109/ICRA.2015.7139738.

[13] P. Berthet-Rayne, M. Power, H. King, and G.-Z. Yang, "Hubot: A three state human-robot collaborative framework for bimanual surgical tasks based on learned models," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2016, pp. 715–722. DOI: 10.1109/ICRA.2016.7487198.

[14] K. Lindgren, K. Huang, and B. Hannaford, "Towards real-time surface tracking and motion compensation integration for robotic surgery," in *2017 IEEE/SICE International Symposium on System Integration (SII)*, Taipei, Taiwan: IEEE, Dec. 2017, pp. 450–456. DOI: 10.1109/SII.2017.8279262.

[15] C. Shin, P. W. Ferguson, S. A. Pedram, J. Ma, E. P. Dutson, and J. Rosen, "Autonomous tissue manipulation via surgical robot using learning based model predictive control," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 3875–3881, 2019. DOI: 10.1109/ICRA.2019.8794159.

[16] S. A. Pedram, C. Shin, P. W. Ferguson, J. Ma, E. P. Dutson, and J. Rosen, "Autonomous suturing framework and quantification using a cable-driven surgical robot," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 404–417, Apr. 2021. DOI: 10.1109/TRO.2020.3031236.

[17] J. Mahler, S. Krishnan, M. Laskey, S. Sen, A. Murali, B. Kehoe, S. Patil, J. Wang, M. Franklin, P. Abbeel, and K. Goldberg, "Learning accurate kinematic control of cable-driven surgical robots using data cleaning and gaussian process regression," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, New Taipei, Taiwan: IEEE, Aug. 2014, pp. 532–539. DOI: 10.1109/CoASE.2014.6899377.

[18] K. L. Schwaner, P. T. Jensen, and T. R. Savarimuthu, "Increasing precision of the raven-ii surgical robot by applying cascade control," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE, Dec. 2018, pp. 1138–1144. DOI: 10.1109/ROBIO.2018.8664821.

[19] H. Peng, X. Yang, Y.-H. Su, and B. Hannaford, "Real-time data driven precision estimator for raven-ii surgical robot end effector position," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2020, pp. 350–356. DOI: 10.1109/ICRA40945.2020.9196915.

[20] 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 *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2016. DOI: 10.1109/ICRA.2016.7487606.

[21] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. D. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA: IEEE, May 2015, pp. 1202–1209. DOI: 10.1109/ICRA.2015.7139344.

[22] B. Thananjeyan, A. Garg, S. Krishnan, C. Chen, L. Miller, and K. Goldberg, "Multilateral surgical pattern cutting in 2d orthotropic gauze with deep reinforcement learning policies for tensioning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2017, pp. 2371–2378. DOI: 10.1109/ICRA.2017.7989275.

[23] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2016, pp. 4178–4185. DOI: 10.1109/ICRA.2016.7487611.

[24] P. Sundaresan, B. Thananjeyan, J. Chiu, D. Fer, and K. Goldberg, "Automated extraction of surgical needles from tissue phantoms," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, IEEE, Aug. 2019. DOI: 10.1109/coase.2019.8843089.

[25] 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 *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2018, pp. 6651–6658. DOI: 10.1109/ICRA.2018.8460583.

[26] M. Hwang, D. Seita, B. Thananjeyan, J. Ichnowski, S. Paradis, D. Fer, T. Low, and K. Goldberg, "Applying depth-sensing to automated surgical manipulation with a da vinci robot," in *2020 International Symposium on Medical Robotics (ISMR)*, IEEE, Nov. 2020. DOI: 10.1109/ISMR48331.2020.9312948.

[27] M. Ginesi, D. Meli, A. Roberti, N. Sansonetto, and P. Fiorini, "Autonomous task planning and situation awareness in robotic surgery," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 3144–3150. DOI: 10.1109/IROS45743.2020.9341382.

[28] A. Garg, S. Sen, R. Kapadia, Y. Jen, S. McKinley, L. Miller, and K. Goldberg, "Tumor localization using automated palpation with gaussian process adaptive sampling," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, Fort Worth, TX: IEEE, Aug. 2016, pp. 194–200. DOI: 10.1109/COASE.2016.7743380.

[29] S. McKinley, A. Garg, S. Sen, D. V. Gealy, J. P. McKinley, Y. Jen, M. Guo, D. Boyd, and K. Goldberg, "An interchangeable surgical instrument system with application to supervised automation of multilateral tumor resection," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, Fort Worth, TX: IEEE, Aug. 2016, pp. 821–826. DOI: 10.1109/COASE.2016.7743487.

[30] M. Hwang, B. Thananjeyan, S. Paradis, D. Seita, J. Ichnowski, D. Fer, T. Low, and K. Goldberg, "Efficiently calibrating cable-driven surgical robots with rgbd fiducial sensing and recurrent neural networks," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5937–5944, Oct. 2020. DOI: 10.1109/LRA.2020.3010746.

[31] O. Özgüner, T. Shkurti, S. Huang, R. Hao, R. C. Jackson, W. S. Newman, and M. C. Çavuşoğlu, "Camera-robot calibration for the da vinci robotic surgery system," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 2154–2161, Oct. 2020. DOI: 10.1109/TASE.2020.2986503.

[32] D. E. Whitney, "The mathematics of coordinated control of prosthetic arms and manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 94, no. 4, pp. 303–309, Dec. 1972. DOI: 10.1115/1.3426611.

[33] N. Aghakhani, M. Geravand, N. Shahriari, M. Vendittelli, and G. Oriolo, "Task control with remote center of motion constraint for minimally invasive robotic surgery," in *2013 IEEE International Conference on Robotics and Automation*, IEEE, May 2013, pp. 5807–5812. DOI: 10.1109/ICRA.2013.6631412.

[34] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving ax=xb on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994. DOI: 10.1109/70.326576.

[35] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, ser. Lecture Notes in Computer Science, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., vol. 9351, Cham: Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[36] S. A. Taghanaki, Y. Zheng, S. K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu, and G. Hamarneh, "Combo loss: Handling input and output imbalance in multi-organ segmentation," *Computerized Medical Imaging and Graphics*, vol. 75, pp. 24–33, Jul. 2019. DOI: 10.1016/j.compmedimag.2019.04.005.

[37] N. Abraham and N. M. Khan, "A novel focal tversky loss function with improved attention u-net for lesion segmentation," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Venice, Italy: IEEE, Apr. 2019, pp. 683–687. DOI: 10.1109/ISBI.2019.8759329.

[38] A. Gandy and F. D.-H. Lau, "Non-restarting cumulative sum charts and control of the false discovery rate," *Biometrika*, vol. 100, no. 1, pp. 261–268, Dec. 2012. DOI: 10.1093/biomet/ass066.